CLASP: CLUSTER-CONDITIONED LATENT ALIGNMENT WITH SHARED PROTOTYPES FOR ENTITY-LEVEL TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasting across heterogeneous entities requires sharing statistical strength while preserving entity-specific variation. We introduce *CLASP*, Cluster-conditioned Latent Alignment with Shared Prototypes, a probabilistic latent partial-pooling mechanism. From an entity's observed history, CLASP constructs context-dependent latent distributions and aligns them with shared Gaussian prototype priors capturing population structure. Prototype-assignment weights, obtained from provided labels or learned from history and metadata, combine independent component draws, inducing an aggregate Gaussian posterior during training and a prototype-induced aggregate Gaussian prior at deployment. This partially pooled latent augments the observed history and conditions diverse forecasting backbones, with an optional validation-fitted readout.

We characterize the latent train–deployment discrepancy by deriving its exact Gaussian KL divergence, proving a data-processing upper bound by component KL divergences, and showing that under one-hot assignments it equals the uncapped CLASP pooling penalty. Under additional bounded-loss, sampling, and sample-independence assumptions, we obtain PAC-Bayes guarantees for posterior-sampled risk under a fixed prediction map and a single entity-level latent law.

We evaluate CLASP on held-out WEATHER-5K stations and eICU patients across forecasting horizons and provided and inferred population structure. Controlled ablations show substantial gains from prototype alignment and prior learning beyond the optional readout, with additional improvements from probabilistic and multi-prototype representations. Within-backbone comparisons show consistent gains from the complete CLASP pipeline across the evaluated probabilistic and deterministic forecasters.

1 INTRODUCTION

Forecasting related time series requires sharing information while preserving entity-specific dynamics. Weather stations share regional conditions, and patients exhibit recurring physiological patterns, yet their trajectories differ. Global models share representations but provide only indirect control over entity-level adaptation (Montero-Manso & Hyndman, 2021); local models specialize but can be unreliable with short or noisy histories. Hierarchical *partial pooling* balances these objectives by sharing statistical strength while retaining variation supported by observations (Gelman, 2006; Gelman et al., 2013).

Neural forecasting methods commonly share parameters, embeddings, or clustered representations (Wang et al., 2019; Cini et al., 2023; Bandara et al., 2020). We introduce *CLASP*, Cluster-conditioned Latent Alignment with Shared Prototypes,[1] to make this sharing explicit at the distributional level. CLASP aligns context-dependent Gaussian latent distributions with shared Gaussian prototype priors. We call these history-dependent distributions *component posteriors*, denoting amortized latent distributions rather than Bayesian posteriors from an explicit observation likelihood. Provided labels or a learned router determine responsibilities that weight both alignment and

[1] Code is available at https://anonymous.4open.science/r/CLASP-ICLR2027.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

aggregation of component draws. The aggregate latent conditions a forecasting backbone, with an optional validation-fitted readout.

During training, CLASP aggregates independent component-posterior draws; at deployment, it aggregates independent prototype-prior draws. These responsibility-weighted sums induce Gaussian distributions rather than categorical mixtures. Observed history remains available to the backbone and, when used, the router; unseen entities require no entity-specific retraining.

We derive the exact Gaussian KL between the aggregate training and deployment laws and a data-processing upper bound given by the unweighted sum of component KLs. Under one-hot assignments, aggregate KL equals the uncapped pooling penalty; under soft assignments, the weighted pooling penalty is generally distinct from both quantities. Appendix A.2 gives the additional conditions required for PAC-Bayes risk certificates.

Experiments on held-out WEATHER-5K stations and eICU patients evaluate unseen-entity forecasting across horizons and provided and inferred population structure. Component and readout ablations examine the latent mechanism, while within-backbone comparisons assess the complete conditioning and readout pipeline without uniquely attributing its gains. Our contributions are threefold: **(1) Latent partial pooling:** context-dependent Gaussian component posteriors aligned with shared prototype priors; **(2) Aggregate latent analysis:** exact aggregate Gaussian KL, a data-processing upper bound, and exact one-hot correspondence with the uncapped pooling penalty; and **(3) Unseen-entity evaluation:** controlled ablations and within-backbone comparisons across horizons, population structures, and forecasting models.

2 Related Work

Transformer, diffusion, and flow-matching approaches advance temporal representation learning and probabilistic forecasting (Wu et al., 2021; Nie et al., 2023; Liu et al., 2024; 2025; Rasul et al., 2021; Li et al., 2022; Kollovieh et al., 2023; Lipman et al., 2023; Liu et al., 2023; Kollovieh et al., 2025). CLASP complements these backbones with latent partial pooling through architecture-specific conditioning interfaces, rather than introducing a new diffusion or flow-matching formulation.

Global–local models, entity embeddings, and clustering structure information sharing across related time series (Wang et al., 2019; Sen et al., 2019; Salinas et al., 2020; Cini et al., 2023; Bandara et al., 2020). The Channel Clustering Module uses learned prototypes to condition forecasting backbones (Chen et al., 2024). Unlike deterministic cluster-conditioned forecasting, CLASP associates groups with Gaussian prototype priors and regularizes context-dependent component posteriors toward them. CLASP adopts classical hierarchical-Bayes partial pooling (Gelman, 2006; Gelman et al., 2013), using amortized neural functions for entity representation and forecasting.

Neural Processes and Neural Diffusion Processes support context-dependent adaptation (Garnelo et al., 2018; Kim et al., 2019; Singh et al., 2019; Kim et al., 2022; Dutordoir et al., 2023). CLASP instead organizes adaptation around multiple shared prototype priors and characterizes replacement of the aggregate component posterior by the aggregate prototype prior at deployment.

PAC-Bayes and Bayesian meta-learning characterize posterior–prior complexity and transferable priors across tasks (McAllester, 1999; Maurer, 2004; Pentina & Lampert, 2014; Amit & Meir, 2018; Rothfuss et al., 2021). Our aggregate-Gaussian analysis characterizes KL between the latent training and deployment laws, distinct from the complementary hierarchical reference-prior certificate under additional sampling and independence assumptions. Appendix F.1 provides detailed comparisons.

3 CLASP: Prototype-Based Latent Partial Pooling

CLASP implements latent partial pooling: context-dependent component posteriors can depart from shared prototype priors to support entity-specific forecasting, while the responsibility-weighted KL penalty discourages these departures. Figure 1 illustrates this mechanism. A forecasting backbone consumes the aggregate latent representation, while an optional validation-fitted readout adjusts the resulting forecasts.

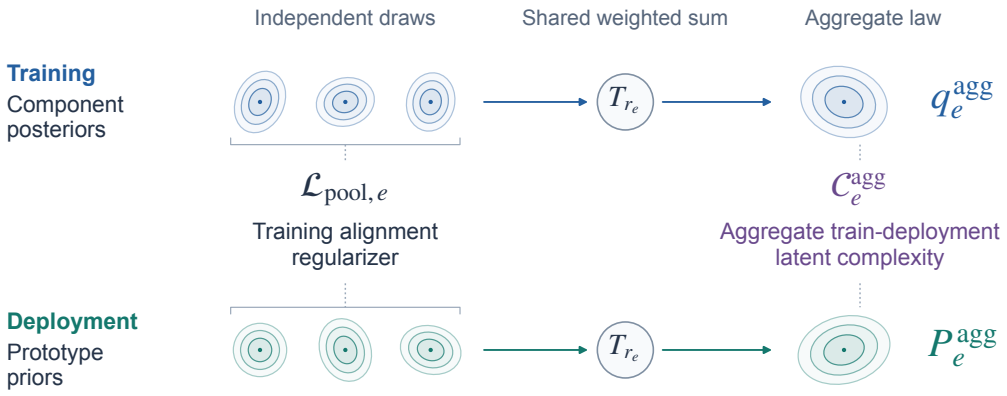

Figure 1: CLASP's latent partial-pooling mechanism. Independent draws from component posteriors during training and prototype priors at deployment pass through the same weighted-sum map T_{r_e}. The alignment regularizer $\mathcal{L}_{\text{pool},e}$ compares component laws, while C_e^{agg} compares the resulting aggregate laws. Observed history remains available at deployment.

3.1 Entity Representations and Shared Prototypes

Each entity $e \in \mathcal{E} = \{1, \dots, N\}$ provides history $x_e \in \mathbb{R}^{T \times d}$ and optional metadata $m_e \in \mathbb{R}^{d_m}$, from which we forecast $y_e \in \mathbb{R}^{H \times d}$. Context-dependent quantities are recomputed per observed window, with window indices suppressed:

$$h_e = f_\theta(x_e, m_e) \in \mathbb{R}^{d_h}. \tag{1}$$

For each $k \in \{1, \dots, K\}$, CLASP learns a Gaussian *prototype prior* P_k and a deterministic *prototype embedding* $u_k \in \mathbb{R}^{d_u}$. The context-dependent *component posterior* $q_{e,k}$ is

$$P_k = \mathcal{N}(\mu_k, \Sigma_k), \qquad q_{e,k} = \mathcal{N}(\mu_{e,k}, \Sigma_{e,k}), \quad (\mu_{e,k}, \Sigma_{e,k}) = g_\psi(h_e, u_k, m_e). \tag{2}$$

The latent dimension is d_z, with $\mu_k, \mu_{e,k} \in \mathbb{R}^{d_z}$; we write $\Theta_k = (\mu_k, \Sigma_k)$ for the prototype parameters. Here "posterior" denotes an amortized history-dependent latent distribution, not a Bayesian posterior from an explicit likelihood. P_k provides the partial-pooling reference, whereas u_k conditions $q_{e,k}$; covariances are diagonal in our implementation.

A *prototype-assignment weight* r_{ek} quantifies how strongly entity e, in its current observed context, is associated with the shared group represented by P_k. We refer to these weights as *responsibilities*. They are either *provided assignments*, encoded as one-hot weights from a provided cluster label c_e, or *inferred responsibilities*, produced by a jointly trained softmax router:

$$r_e = \begin{cases} \text{onehot}(c_e), & \text{provided assignments,} \\ \pi_\omega(h_e, m_e), & \text{inferred responsibilities,} \end{cases} \qquad r_e \in \Delta^{K-1}. \tag{3}$$

Responsibilities weight both posterior–prototype alignment and independent component draws. The responsibilities also define the deterministic aggregate prototype embedding $u_e = \sum_{k=1}^{K} r_{ek} u_k$. Training draws from $q_{e,k}$, whereas deployment draws from P_k; in both cases, the weighted sum is Gaussian, not a categorical mixture, so P_k are latent reference distributions rather than forecast-mixture components. The component posteriors use no future targets and remain computable at deployment; replacing their draws with prototype-prior draws is a deliberate modeling choice. Observed history remains available to the backbone and router, so unseen entities require no entity-specific retraining. Appendix A.1 summarizes the notation.

3.2 Partial Pooling Objective

CLASP aligns each component posterior with its prototype prior through the responsibility-weighted KL penalty

$$\mathcal{L}_{\text{pool},e} = \sum_{k=1}^{K} r_{ek} D_{\text{KL}}(q_{e,k} \| P_k), \qquad \mathcal{L}_{\text{pool}} = \frac{1}{N_{\text{tr}}} \sum_{e=1}^{N_{\text{tr}}} \mathcal{L}_{\text{pool},e}. \tag{4}$$

The forecasting loss $\mathcal{L}_{\text{forecast}}$ evaluates the backbone prediction, while $\mathcal{L}_{\text{cluster}}$ and $\mathsf{H}(\bar{r})$ regularize inferred responsibilities. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{pool}} + \gamma \mathcal{L}_{\text{cluster}} - \eta_{\text{ent}} \mathsf{H}(\bar{r}). \tag{5}$$

All reported experiments use the *uncapped pooling objective*. The pooling term learns both posterior–prototype alignment and prototype-prior parameters, so $\lambda = 0$ removes both signals and is not an alignment-only ablation. Further objective details, the prototype-moment interpretation, router terms, optional KL cap, warmup, and training algorithm appear in Appendix B.2.

3.3 Aggregate Training and Deployment Laws

Throughout this subsection, we condition on a fixed context, its responsibility vector r_e, and the model parameters. Define the weighted-sum map $T_{r_e}(z_1, \ldots, z_K) = \sum_{k=1}^{K} r_{ek} z_k$. During training, CLASP draws independently from the component posteriors, $z^q_{e,k} \sim q_{e,k}$, and applies the map:

$$z^q_e = \sum_{k=1}^{K} r_{ek} z^q_{e,k} \sim q^{\text{agg}}_e = \mathcal{N}(\mu^q_e, \Sigma^q_e), \qquad \mu^q_e = \sum_{k=1}^{K} r_{ek} \mu_{e,k}, \qquad \Sigma^q_e = \sum_{k=1}^{K} r^2_{ek} \Sigma_{e,k}. \tag{6}$$

At deployment, CLASP instead draws independently from the shared prototype priors, $z^P_{e,k} \sim P_k$, and applies the same map with the same context-dependent responsibilities:

$$z^P_e = \sum_{k=1}^{K} r_{ek} z^P_{e,k} \sim P^{\text{agg}}_e = \mathcal{N}(\mu^P_e, \Sigma^P_e), \qquad \mu^P_e = \sum_{k=1}^{K} r_{ek} \mu_k, \qquad \Sigma^P_e = \sum_{k=1}^{K} r^2_{ek} \Sigma_k. \tag{7}$$

Because the component draws are independent, both aggregates are Gaussian with covariance contributions weighted by r^2_{ek}. Importantly, P^{agg}_e is induced by a weighted sum of independent prototype draws, not by categorical mixture sampling: $P^{\text{mix}}_{r_e} = \sum_k r_{ek} P_k$ is a distinct auxiliary reference used only in Appendix A.4.2. Under one-hot assignments the two constructions coincide.

Component-posterior–to–prototype-prior replacement removes context-specific component-posterior parameters from the sampled latent, but does not remove observed history from the forecasting backbone. With inferred responsibilities, history also influences the aggregate deployment prior through r_e. The component posteriors remain computable from observed inputs; replacing their draws with prototype-prior draws is a deliberate modeling choice. Section 4.5 (Table 6) compares aggregate posterior sampling, the aggregate prior mean, and aggregate prior sampling while holding the fitted forecasting model, per-context responsibilities, and validation-fitted readout fixed within each horizon.

3.4 Aggregate Latent Complexity

For a fixed observed context, learned component parameters and common responsibilities, define the train–deployment latent discrepancy

$$\mathcal{C}^{\text{agg}}_e := D_{\text{KL}}(q^{\text{agg}}_e \| P^{\text{agg}}_e). \tag{8}$$

Its exact Gaussian form is given in Appendix A.2.1. Let $\mathcal{C}^{\text{comp}}_e = \sum_{k=1}^{K} D_{\text{KL}}(q_{e,k} \| P_k)$. Because the component laws pass through the same deterministic weighted-sum map, data processing gives:

Theorem 1 (Aggregate latent data-processing bound)**.** *For fixed component laws, independent component draws, and common $r_e \in \Delta^{K-1}$,*

$$\mathcal{C}^{\text{agg}}_e \le \mathcal{C}^{\text{comp}}_e = \sum_{k=1}^{K} D_{\text{KL}}(q_{e,k} \| P_k). \tag{9}$$

The uncapped pooling penalty $\mathcal{L}_{\text{pool},e} = \sum_k r_{ek} D_{\text{KL}}(q_{e,k} \| P_k)$ is also bounded by $\mathcal{C}^{\text{comp}}_e$, but under soft responsibilities it has no general ordering with $\mathcal{C}^{\text{agg}}_e$. For one-hot assignments, however, the aggregate laws select the same component and the correspondence is exact:

Corollary 1 (Exact one-hot correspondence)**.** *If $r_e = \text{onehot}(c_e)$, then*

$$\mathcal{C}^{\text{agg}}_e = D_{\text{KL}}(q_{e,c_e} \| P_{c_e}) = \mathcal{L}_{\text{pool},e}. \tag{10}$$

Thus, for the uncapped configurations used here, provided one-hot assignments give an exact theory–algorithm correspondence between the optimized per-context pooling penalty and aggregate train–deployment KL. Proofs and soft-assignment counterexamples are in Appendix A.2.

PAC-Bayes scope. Under additional bounded-loss, sampling, and sample-independence conditions, $\mathcal{C}_e^{\mathrm{agg}}$ also yields a PAC-Bayes certificate for posterior-sampled risk under a single entity-level latent law. This does not directly certify the implemented window-dependent, prior-sampled predictor; assumptions and extensions are given in Appendix A.2.6.

3.5 Forecasting Interface and Optional Readout

CLASP augments rather than replaces observed history with a partially pooled latent representation, injected through an architecture-specific conditioning interface. A raw forecast or trajectory is

$$\widehat{y}_e^{(s)} = F_\phi(x_e, g_e^{(s)}), \qquad g_e^{(s)} = [h_e, u_e, z_e^{(s)}], \tag{11}$$

where $z_e^{(s)} \sim q_e^{\mathrm{agg}}$ during training and $z_e^{(s)} \sim P_e^{\mathrm{agg}}$ at deployment by default. The Timer-XL and D3VAE retrofits instead use the aggregate prior mean (Appendix C.2); Table 8 summarizes these conventions. Generative backbones additionally receive encoder states $M_e \in \mathbb{R}^{T \times d_h}$, suppressed along with generative randomness. Flow matching is the primary backbone (Lipman et al., 2023; Liu et al., 2023); Appendix B.1 describes diffusion and deterministic instantiations (Figure 2).

Separate from latent partial pooling, an optional *validation-fitted readout* blends raw trajectories with history-based persistence and climatology/context-mean anchors:

$$\widetilde{y}_e^{(s)} = (1 - \lambda_1 - \lambda_2)\widehat{y}_e^{(s)} + \lambda_1 y_e^{\mathrm{pers}} + \lambda_2 y_e^{\mathrm{clim}}. \tag{12}$$

The prototype- and lead-specific profiles (indices suppressed) satisfy $\lambda_1, \lambda_2 \geq 0$ and $\lambda_1 + \lambda_2 \leq 1$. They are fitted on validation entities and fixed on test entities.

We call latent-conditioned forecasts before this readout *raw CLASP* and those after the fixed readout *decoded CLASP*. The *full CLASP pipeline* combines latent conditioning with the reported readout configuration. Unless an ablation specifies otherwise, full CLASP uses both persistence and climatology/context-mean anchors. These terms refer to readout use; the point estimator and draw count are stated separately.

Headline MAE/MSE for the readout-enabled pipeline use the arithmetic mean $\widetilde{y}_e^{\mathrm{point}} = \frac{1}{S}\sum_{s=1}^{S} \widetilde{y}_e^{(s)}$. The no-readout WEATHER-5K comparisons in Tables 3 and 23, and the eICU raw-forecast row in Table 24, average raw trajectories, keeping point aggregation fixed when the blend is enabled or disabled. The basic probabilistic backbones use $S = 100$ trajectories with fresh latent and initial-noise samples; the four direct point forecasters use $S = 1$. Retrofit-specific exceptions appear in Table 8. CRPS and coverage use raw, unblended predictive draws. P10–P90 intervals use pointwise empirical quantiles of those draws. Appendix D.1.1 details the anchors, fitting, and inference; Table 3 reports readout ablations.

4 Experiments

We evaluate unseen-entity forecasting with provided and inferred population structure through component ablations, within-backbone comparisons, predictive-distribution analysis, and inference-time latent replacement.

Protocol and configurations. On **WEATHER-5K**, the headline and six-backbone comparisons use provided zones ($K = 5$), while component ablations and inference-distribution analysis use inferred responsibilities ($K = 4$), unless varied. **eICU** uses inferred responsibilities ($K = 4$). Headline CLASP uses flow matching on WEATHER-5K and diffusion on eICU. Train/validation/test entities are disjoint; validation supports checkpoint selection and readout fitting, with no refitting on test entities. Full protocols, hyperparameters, and computational costs are in Appendices D.2, D.3, D.1, B.3, and B.4.

Datasets. WEATHER-5K (Han et al., 2026) contains 5,672 stations; we use 125 balanced across five latitude zones, split 99/11/15 for train/validation/test. From 48-hour histories, we forecast six channels at $H \in \{24, 72, 168\}$ using coordinates as metadata. eICU (Pollard et al., 2018; 2019;

Goldberger et al., 2000) contains 93,286 patients, split 76,834/8,529/7,923. From 12-hour histories, we forecast six vital signs at $H \in \{6, 12, 24\}$; four overlapping diagnosis indicators are metadata, not cluster labels. Preprocessing details are in Appendices D.2 and D.3.1.

Metrics and uncertainty. We report MAE and MSE averaged over forecast steps and six channels. Headline CLASP forecasts average $S = 100$ draws after the validation-fitted readout; CRPS and coverage use raw draws. Five training seeds share the same 15 WEATHER-5K test stations, so seed-level variation measures run-to-run variability, not uncertainty across station populations.

Baselines. We compare against D3VAE (Li et al., 2022), TSDiff (Kollovieh et al., 2023), NsDiff (Ye et al., 2025), TSFlow (Kollovieh et al., 2025), and Timer-XL (Liu et al., 2025) in both zero-shot and fine-tuned settings. WEATHER-5K D3VAE and TSDiff use shared-protocol re-implementations; their eICU rows retain configurations. Appendix C.1 documents adaptations and sampling. TSFlow is evaluated only on WEATHER-5K because its implementation requires horizons divisible by 24.

4.1 Main Results

Tables 1 and 2 compare performance against external baselines. All CLASP results use the full validation-fitted readout, distinct from explicitly marked raw-forecast ablations. Relative error reductions use the lowest listed baseline error at each horizon and metric as the denominator.

Table 1: Forecasting the 15 held-out WEATHER-5K stations with provided zones ($K = 5$). CLASP uses flow matching and the full validation-fitted forecast readout. Lower is better; best values are **bold** and second-best values are underlined. ZS/FT denote zero-shot/fine-tuned Timer-XL.

	$H = 24$		$H = 72$		$H = 168$	
Model	MAE	MSE	MAE	MSE	MAE	MSE
D3VAE	2.306	13.887	3.285	29.050	3.131	26.462
TSDiff	1.602	8.308	2.321	17.408	2.646	21.316
Timer-XL ZS	1.668	8.822	2.324	18.615	2.667	24.547
Timer-XL FT	1.642	7.736	2.194	15.618	2.578	22.501
NsDiff	1.478	6.355	2.036	13.400	2.435	19.591
TSFlow	1.427	6.398	2.173	13.912	2.426	18.948
CLASP	**1.291**	**5.111**	**1.902**	**11.848**	**2.290**	**17.193**

Table 2: Forecasting held-out entities with inferred responsibilities ($K = 4$). CLASP uses diffusion on eICU and flow matching on WEATHER-5K with the full validation-fitted readout. Horizons are 6/12/24 and 24/72/168 hours, respectively. Lower is better; bold/underlining mark best/second-best. The WEATHER-5K row is compared with the baselines in Table 1.

		H_1		H_2		H_3	
Dataset	Model	MAE	MSE	MAE	MSE	MAE	MSE
eICU $H = 6, 12, 24$	D3VAE	8.396	161.909	12.196	198.283	10.886	225.790
	TSDiff	8.119	154.895	8.253	157.774	9.367	201.054
	Timer-XL ZS	6.441	105.946	7.294	127.099	8.374	157.647
	Timer-XL FT	7.027	120.602	7.552	135.798	8.135	154.234
	NsDiff	6.673	105.058	7.191	118.805	7.576	131.727
	CLASP	**5.801**	**91.291**	**6.350**	**104.696**	**6.936**	**120.552**
WEATHER-5K $H = 24, 72, 168$	CLASP	**1.280**	**5.049**	**1.894**	**11.691**	**2.298**	**17.040**

Provided clusters. On the 15 held-out WEATHER-5K stations, flow-matching CLASP with provided zones ($K = 5$) achieves relative error reductions versus the strongest listed baseline of $9.5\%/6.6\%/5.6\%$ in MAE and $19.6\%/11.6\%/9.3\%$ in MSE at $H = 24/72/168$ hours. These gains concern held-out stations under the reported split with geographic assignments available.

Inferred clusters. With inferred responsibilities ($K = 4$), diffusion CLASP on eICU achieves relative error reductions versus the strongest listed baseline of $9.9\%/11.7\%/8.4\%$ in MAE and

$13.1\%/11.9\%/8.5\%$ in MSE at $H = 6/12/24$ hours. On the same 15 held-out WEATHER-5K stations, flow-matching CLASP with inferred responsibilities ($K = 4$) achieves relative MAE error reductions versus the strongest listed baseline in Table 1 of $10.3\%/7.0\%/5.3\%$ at $H = 24/72/168$ hours. Appendix E.3.3 examines sensitivity to the number of clusters K.

Variation across runs and entities. Results average five training seeds unless stated otherwise. Seed-level variation (Table 21) measures optimization stability under a fixed entity split, not uncertainty across new populations. WEATHER-5K has only 15 independent test entities, three per provided zone; repeating training does not enlarge this sample. Station-level variability (Table 22) is therefore reported separately, without establishing paired significance between methods. Appendix E.1.1 provides channel-level errors.

4.2 Contributions of Latent Partial Pooling and Forecast Readout

Table 3 provides the main mechanism analysis on the 15 held-out WEATHER-5K stations. Within each configuration, raw and readout evaluations use the same fitted forecaster.

Table 3: Flow-matching controls on WEATHER-5K; CLASP uses inferred responsibilities ($K = 4$ unless varied). The plain-backbone and CLASP raw/readout pairs form a matched 2×2 design, using arithmetic means for both raw and decoded point forecasts. All other controls use the validation-fitted readout. Lower is better; bold marks column minima. Complete controls appear in Table 23.

	$H = 24$		$H = 72$		$H = 168$	
Variant	MAE	MSE	MAE	MSE	MAE	MSE
Plain backbone (raw)	1.555	6.801	2.350	15.275	2.710	21.035
Plain backbone + readout	1.538	6.678	2.301	14.835	2.626	20.383
No pooling ($\lambda = 0$; prior untrained)	1.629	7.269	2.302	15.266	2.659	20.636
Raw CLASP (no readout)	1.294	5.181	1.935	11.899	2.371	17.536
Full CLASP	**1.280**	**5.049**	**1.894**	**11.691**	**2.298**	**17.040**
Deterministic prototypes	1.318	5.342	1.931	12.084	2.324	17.488
Single prototype ($K = 1$)	1.347	5.606	1.908	11.732	2.329	17.341
Without encoder states M_e	1.358	5.742	1.974	12.461	2.361	17.812

Raw and readout-adjusted forecasts use sample means. The plain backbone uses a global readout profile; CLASP uses prototype-specific profiles. The 2×2 design compares each model with readout disabled or enabled, so the readout-enabled contrast includes profile specialization. It does not separately identify the contributions of alignment and prototype-prior learning within CLASP.

Pooling and readout. No pooling $\rightarrow$ raw CLASP $\rightarrow$ full CLASP gives MAE $1.629 \rightarrow 1.294 \rightarrow 1.280$ at $H = 24$, with the same ordering thereafter. Raw CLASP reduces MAE versus the matched plain backbone by 16.8%, 17.7%, and 12.5% across horizons, so most of the gain precedes readout; for the same fitted CLASP model, readout further reduces MAE by 1.1%, 2.1%, and 3.1%, respectively.

Setting $\lambda = 0$ retains component-posterior training and prototype-prior sampling at deployment but removes both posterior–prototype alignment and the pooling-based prototype-prior learning signal, leaving the priors untrained. Full CLASP's 13.6–21.4% MAE reduction relative to this control supports the *joint contribution of alignment and prototype-prior learning*, not alignment alone.

Prototype structure and temporal context. The Gaussian-prototype/KL construction reduces MAE by 1.1–2.9% relative to the tested deterministic-vector/squared-distance alternative, not deterministic representations generally. The $K = 1$ control retains a shared latent reference but removes prototype specialization; its higher errors support multiple shared regimes over a single global prior in this setting. Removing M_e disables time-resolved encoder-state cross-attention while retaining the summary representation, latent, prototype embedding, and raw history.

Connection to aggregate-latent theory. Theorem 1 establishes $\mathcal{C}_e^{\mathrm{agg}} \leq \mathcal{C}_e^{\mathrm{comp}}$ using the *unweighted* component KL sum. Corollary 1 equates aggregate KL with the *uncapped* pooling penalty under one-hot assignments. Under soft responsibilities, weighted pooling and aggregate KL have no general ordering. The degradation without pooling complements these results but does not establish that

reducing aggregate KL causes lower forecast error. Appendix E.2 reports further controls, including eICU. Table 25 examines sensitivity to λ, which controls alignment and prototype-prior learning.

4.3 CLASP Across Backbones

Table 4 compares four deterministic and two probabilistic backbones with CLASP-augmented counterparts on held-out WEATHER-5K stations. CLASP uses geographic-zone assignments ($K = 5$) and adds both latent conditioning and the validation-fitted forecast readout. These within-family comparisons evaluate the complete pipeline, not latent conditioning alone. Here, *plain* means the backbone without the CLASP module or readout and is distinct from CLASP with $K = 1$. Optimization protocols and training and tuning budgets are matched within each pair (Appendix B.1).

Table 4: Within-backbone error reductions (%) on held-out WEATHER-5K stations with provided assignments ($K = 5$). Each CLASP variant adds both latent conditioning and the validation-fitted readout to its plain counterpart. Entries are $100(E_{\text{plain}} - E_{\text{CLASP}})/E_{\text{plain}}$ for MAE or MSE; positive values indicate improvement. Absolute errors appear in Table 26.

	$H = 24$		$H = 72$		$H = 168$	
Backbone	MAE	MSE	MAE	MSE	MAE	MSE
Transformer	3.3	7.1	3.2	7.6	4.3	11.2
GRU	13.8	18.4	25.6	20.8	18.3	21.2
SSM	4.6	8.9	4.9	10.6	5.8	13.8
TCN	9.0	19.4	7.2	15.0	7.4	16.6
DiT	8.3	17.2	10.3	19.1	8.5	16.9
FM	17.0	24.8	19.1	22.4	15.5	18.3

The complete CLASP pipeline reduces both errors at every reported horizon in all six evaluated families, with relative reductions of 3.2–25.6% in MAE and 7.1–24.8% in MSE. These results support backbone flexibility under the reported protocol, rather than an architectural ranking or a claim about unevaluated backbones. Section 4.2 separately examines pooling and readout contributions through controlled flow-matching comparisons.

For CLASP, deterministic backbones use one forward pass and one latent draw; DiT and flow matching (FM) use $S = 100$ trajectories. Point forecasts use arithmetic means after the validation-fitted readout. Appendix E.3.1 reports absolute errors; Appendices E.3.2 and C.2 extend the evaluation to five external forecasters and describe their architecture-specific conditioning interfaces.

Population structure. Inferred responsibilities produce overlapping groups on WEATHER-5K and eICU; cluster errors, heterogeneity, and sensitivity are reported in Appendix E.3.

4.4 Probabilistic Forecasting Quality

Table 5 evaluates CRPS and empirical P10–P90 coverage from *raw predictive draws*, without the optional forecast readout. These metrics assess predictive distributions, whereas headline MAE/MSE evaluate the validation-fitted point-forecast pipeline. CRPS is a proper scoring rule for which lower is better; coverage is assessed against the nominal level 0.80, not minimized.

CLASP has the lowest CRPS at all reported horizons, with reductions of 6.4–6.8% versus TSFlow on WEATHER-5K and 7.3–8.5% versus NsDiff on eICU. Coverage is not uniformly closer to the nominal 0.80 level, showing that improved CRPS does not imply improved coverage. Full distributional results appear in Appendix E.1.2.

4.5 Inference-Distribution Analysis

Table 6 directly tests CLASP's deliberate posterior-to-prior deployment choice (Section 3.3): does sampling from the shared prototype-induced law P_e^{agg} improve forecasting when the history-dependent aggregate component posterior q_e^{agg} remains computable? Within each horizon, the fitted forecasting model, per-context responsibilities, and validation-fitted readout are held fixed; only the latent source changes.

Table 5: Predictive-distribution quality from raw draws, distinct from readout-adjusted MAE/MSE. CLASP uses flow matching with provided assignments ($K = 5$) on WEATHER-5K and diffusion with inferred responsibilities ($K = 4$) on eICU. Entries are CRPS | P10–P90 coverage; bold marks CRPS minima, with coverage compared against nominal 0.80. Reproduced from Table 20.

WEATHER-5K	$H = 24$	$H = 72$	$H = 168$
NsDiff	1.085 | 0.787	1.583 | 0.745	1.912 | 0.714
TSFlow	1.052 | 0.749	1.547 | 0.793	1.881 | 0.741
CLASP	**0.981** | 0.796	**1.442** | 0.789	**1.761** | 0.776
eICU	$H = 6$	$H = 12$	$H = 24$
NsDiff	4.823 | 0.833	5.412 | 0.796	5.891 | 0.780
CLASP	**4.412** | 0.781	**4.987** | 0.772	**5.463** | 0.765

Table 6: Latent-source controls on 15 held-out WEATHER-5K stations with flow matching and inferred responsibilities ($K = 4$). Within each horizon, controls share the fitted model, responsibilities, and validation-fitted readout; all retain generative noise, including the prior-mean control. Lower is better; bold marks minima.

	$H = 24$		$H = 72$		$H = 168$	
Latent source	MAE	MSE	MAE	MSE	MAE	MSE
Aggregate posterior samples	1.318	5.301	1.933	12.067	2.321	17.398
Aggregate prior mean	1.299	5.184	1.912	11.911	2.302	17.272
Aggregate prior samples	**1.280**	**5.049**	**1.894**	**11.691**	**2.298**	**17.040**

Prior sampling reduces MAE by 2.9%/2.0%/1.0% and MSE by 4.8%/3.1%/2.1% relative to aggregate-posterior sampling. The result supports the reported deployment construction, although posterior-to-prior replacement changes both latent means and covariances and therefore does not isolate shrinkage causally. Relative to the aggregate prior mean, prior sampling reduces MAE by 1.5%/0.9%/0.2%; posterior-mean performance is unavailable, so this is not a complete mean-versus-sampling factorial. Appendix E.2.2 specifies the sampling and readout protocol.

Appendix E.4.1 separately examines latent train–deployment discrepancy with frozen $K = 5$ parameters. Aggregate KL and uncapped pooling both equal 0.97 nats under one-hot assignments but differ under soft routing (1.21 versus 1.38 nats). The mean unweighted component sum is 6.84 nats for both rules, consistent with the data-processing bound (Theorem 1); the one-hot equality agrees with Corollary 1. These diagnostics are not forecast-risk guarantees. Fixed-predictor bounds and their sampling and independence assumptions are detailed in Appendices E.4.2 and E.4.3.

5 Limitations and Future Work

CLASP relies on transferable population structure: homogeneous or continuously varying populations may favor simpler alternatives, while inappropriate assignments can suppress entity-specific variation. Larger K increases specialization but reduces support per prototype and increases training cost, which scales as $O(B_{\text{batch}} K d_z)$. Although $K = 4$ attains the lowest horizon-averaged errors among the evaluated prototype counts, eICU optima vary by horizon and metric; this does not establish a universal choice of K.

Empirical evidence is limited by the 15-station WEATHER-5K holdout. The matched 2×2 experiment separates readout effects for the main flow-matching configuration, but not across all backbones, and the $\lambda = 0$ control jointly removes alignment and prototype-prior learning. The aggregate-KL theory characterizes latent discrepancy; its PAC-Bayes guarantees require additional assumptions and do not directly certify the implemented window-dependent, prior-sampled predictor. Finally, eICU evaluates forecasting methodology rather than clinical utility or subgroup discovery. Clinical use would require external and subgroup validation, calibration assessment, and human oversight. Further limitations and computational details appear in Appendix F.2.

AI USE STATEMENT

In this work, we used generative AI tools for retrieval and discovery, including identifying potentially relevant related work. We have not used generative AI tools for research ideation, methodology development, theoretical derivations, experimental design, data analysis, result generation, or evaluation, and the remaining required disclosure categories are not applicable to this work. Additionally, we used generative AI tools for writing, editing, formatting, grammar refinement, and language polishing. We reviewed all AI-assisted work and manually verified retrieved references and related-work information against the original sources. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We evaluate CLASP using WEATHER-5K observations and de-identified records from the eICU Collaborative Research Database. We use eICU under the PhysioNet Credentialed Health Data License and Use Agreement, do not redistribute restricted patient-level records, and report only aggregate forecasting results and cohort statistics. These experiments assess forecasting methodology and do not establish clinical decision-support utility or safety. Shared latent structure may reflect population imbalance and missingness patterns, leading to uneven performance across subgroups. Inaccurate forecasts or miscalibrated uncertainty may also encourage overconfidence. High-stakes use would require external validation, subgroup evaluation, calibration checks, and human oversight. Dataset access and handling are described in Appendices D.2.3 and D.3.3, and Appendix D.4 discusses potential benefits and broader risks.

REPRODUCIBILITY STATEMENT

The CLASP implementation is available in the anonymous code repository at `https://anonymous.4open.science/r/CLASP-ICLR2027`. Section 3 defines the latent module and training objective. Appendices B.2, B.1, and B.3 specify the training algorithm, backbone dimensions and conditioning interfaces, optimizer settings, training budgets, and hyperparameters. Baseline implementations and protocol adaptations are documented in Appendix C.1.

The training/validation/test splits contain $99/11/15$ WEATHER-5K stations and $76{,}834/8{,}529/7{,}923$ eICU patients. Appendix D.1 specifies the split rules, window construction, horizons, sampling budgets, and seed usage; Tables 16 and 21 give the exact station assignments and per-seed results, respectively. Dataset construction, preprocessing, and access requirements are detailed in Appendices D.2 and D.3. Validation entities support checkpoint selection and readout fitting; held-out test entities are never refit. Appendix D.1.1 specifies latent sampling, readout fitting, point and interval readouts, and the inference algorithm, with backbone-specific numerical samplers in Appendix B.1.

Appendix B.4 reports the training hardware, including the single 24 GiB RTX 4090 GPU, CPU and system memory, and total training and inference times across the three horizons for the CLASP flow-matching backbone. Appendix A provides the assumptions and proofs, including aggregate-KL theory in Appendix A.2 and complementary mixture-reference certificates in Appendix A.4. Their applicability conditions are detailed in Appendices A.2.5–A.2.8 and A.4.1–A.4.5; the separate fixed-predictor evaluation protocol is in Appendix E.4.2. Appendix E.4.1 separately documents the measured aggregate-KL diagnostic.

REFERENCES

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. In *International Conference on Machine Learning*, pp. 205–214. PMLR, 2018.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert systems with applications*, 140:112896, 2020.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130635–130663, 2024.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023.

Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. Taming local effects in graph-based spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55375–55393, 2023.

Vincent Dutordoir, Alan Saul, Zoubin Ghahramani, and Fergus Simpson. Neural diffusion processes. In *International Conference on Machine Learning*, pp. 8990–9012. PMLR, 2023.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pp. 1704–1713. PMLR, 2018.

Andrew Gelman. Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics*, 48 (3):432–435, 2006.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC Press, 2013.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35971–35983, 2022.

Tao Han, Zhibin Wen, Zhenghao Chen, Dazhao Du, Song Guo, and LEI BAI. Benchmarking physics-informed time-series models for operational global station weather forecasting. In *International Conference on Machine Learning*, 2026.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7407–7417. IEEE, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Donggyun Kim, Seongwoong Cho, Wonkwang Lee, and Seunghoon Hong. Multi-task processes. In *International Conference on Learning Representations*, 2022.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28341–28364, 2023.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 95435–95457, 2025.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23009–23022, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 83982–84006, 2025.

Andreas Maurer. A note on the pac bayesian theorem. *arXiv preprint cs/0411099*, 2004.

David A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999. doi: 10.1023/A:1007618624809.

Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary ϕ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 11(26):789–814, 2010.

Pablo Montero-Manso and Rob J Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, 37(4):1632–1653, 2021.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with Transformers. In *International Conference on Learning Representations*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pp. 991–999. PMLR, 2014.

Tom Pollard, Alistair Johnson, Jesse Raffa, Leo Anthony Celi, Omar Badawi, and Roger Mark. eICU Collaborative Research Database (version 2.0). PhysioNet, 2019. RRID:SCR_007345.

Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):180178, 2018.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pp. 8857–8868. PMLR, 2021.

Jonas Rothfuss, Vincent Fortuin, Martin Josifoski, and Andreas Krause. Pacoh: Bayes-optimal meta-learning with pac-guarantees. In *International Conference on Machine Learning*, pp. 9116–9126. PMLR, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. Sequential neural processes. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pp. 6607–6617. PMLR, 2019.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Appendix roadmap. Appendix A contains the theory and its assumptions. Appendix B gives architecture and training details, and Appendix C documents baseline implementations and CLASP retrofit interfaces. Appendix D specifies datasets, preprocessing, splits, and forecast readout. Appendix E reports additional forecasting, ablation, robustness, and latent-divergence analyses; Appendix F provides extended discussion. Appendix E.4 separates empirical latent-KL diagnostics from fixed-predictor concentration calculations under their stated sampling assumptions. These analyses do not automatically certify the deployed, window-dependent CLASP predictor.

TABLE OF CONTENTS

A Theory . . . 17

A.1 Overview and Notation . . . 17

A.2 Aggregate Latent Complexity . . . 17

A.2.1 Aggregate complexity and exact Gaussian KL . . . 17

A.2.2 Data-processing bound and proof . . . 19

A.2.3 Soft pooling has no general ordering . . . 19

A.2.4 Exact one-hot theory–algorithm correspondence . . . 20

A.2.5 PAC-Bayes setup: risks and additional assumptions . . . 20

A.2.6 PAC-Bayes guarantee for fixed responsibilities . . . 21

A.2.7 Optional Extension: Soft Responsibilities via a Simplex Cover . . . 21

A.2.8 Optional Extension: Hierarchical Aggregate Certificate . . . 24

A.3 Responsibility-Weighted Prototype Moment Matching . . . 25

A.4 Complementary Mixture-Reference Certificates . . . 27

A.4.1 Risk definitions and assumptions . . . 27

A.4.2 Reference mixtures and entity-level certificate . . . 27

A.4.3 Hierarchical reference-prior learning . . . 28

A.4.4 Partial pooling and aggregate latent laws . . . 29

A.4.5 Window-level prediction and decoding . . . 29

A.5 Proof of Lemma 3 . . . 30

A.6 Proof of Theorem 4 . . . 31

A.7 Proof of Theorem 5 . . . 32

A.8 Proof of Corollary 3 . . . 33

B Architecture and Training . . . 35

B.1 Backbone Pipeline Details . . . 35

B.1.1 Headline Flow-Matching Backbone . . . 36

B.1.2 DiT Diffusion Backbone . . . 37

B.1.3 Transformer Forecaster . . . 38

B.1.4 GRU Forecaster . . . 38

B.1.5 SSM Forecaster . . . 38

B.1.6 TCN Forecaster . . . 39

B.2 Shared Training Procedure . . . 40

B.3 CLASP Hyperparameter Details . . . 43

B.4 Training Environment and Cost . . . 44

B.5 Scalability of Inferred Responsibilities . . . 44

C Baseline Implementations . . . 46

C.1 Implementation Sources and Protocol Adaptations . . . 46

C.2 CLASP Retrofit Interfaces . . . 49

D Datasets and Experimental Protocol . . . 54

D.1 Unseen-Entity Evaluation Protocol . . . 54

D.1.1 Forecast readout and inference . . . 55

D.2 WEATHER-5K Details. . . . 57

D.2.1 Circular encoding of wind direction. . . . 58

D.2.2 Sampling and grouping . . . 58

D.2.3 License and access . . . 58

D.3 eICU Details . . . 59

D.3.1 Preprocessing pipeline . . . 59

D.3.2 Diagnosis overlap and split statistics . . . 60

D.3.3 License and use agreement . . . 60

D.4 Data Use and Broader Impacts . . . 61

E Additional Experimental Results . . . 62

E.1 Unseen-Entity Forecasting Quality and Variability . . . 62

E.1.1 Per-channel and per-vital results . . . 62

E.1.2 Predictive Distribution Quality . . . 62

E.1.3 Variability Across Training Runs and Held-Out Entities . . . 63

E.2 Contributions of Latent Partial Pooling and Forecast Readout . . . 67

E.2.1 Component ablations across datasets . . . 67

E.2.2 Which latent source is used at inference? . . . 69

E.2.3 Sensitivity to the alignment weight λ . . . 70

E.2.4 Diagnostic: Prototype Moment Alignment . . . 71

E.3 Transfer Across Backbones and Population Structures . . . 73

E.3.1 Within-Backbone Forecasting Results . . . 73

E.3.2 CLASP Retrofits Across Forecasting Backbones . . . 73

E.3.3 Sensitivity to the Number of Prototypes. . . . 74

E.3.4 Partial Pooling Under Limited Training Support . . . 76

E.3.5 Cluster Analysis . . . 77

E.3.6 Cluster-wise forecasting on WEATHER-5K . . . 77

E.3.7 Cluster-wise forecasting on eICU . . . 79

E.4 Aggregate Latent, Fixed-Predictor, and Reference Diagnostics . . . 80

E.4.1 Aggregate latent divergence diagnostics. . . . 80

E.4.2 Held-Out Fixed-Predictor Concentration Evaluation . . . 81

E.4.3 Additional Fixed-Predictor and Reference Diagnostics . . . 83

F Extended Discussion . . . 87

F.1 Additional Related Work . . . 87

F.2 Population Structure, Computational Cost, and Transfer Limitations 87

A THEORY

Theory roadmap. Appendix A.2 gives the primary theory–algorithm connection: exact aggregate Gaussian KL, deterministic data processing, and one-hot equality with uncapped pooling. Its PAC-Bayes interpretations and optional extensions require additional assumptions. Appendix A.3 gives the responsibility-weighted moment interpretation of the pooling objective. Appendix A.4 and the subsequent proofs provide complementary reference-prior and hierarchical certificates. Their categorical mixture is an auxiliary PAC-Bayes reference; the weighted-sum aggregate Gaussian is CLASP's sampled deployment law. Hierarchical certificates retain a posterior-producing rule fixed independently of the certified entities and the stated within- and across-entity sampling assumptions.

A.1 OVERVIEW AND NOTATION

Throughout the paper, $q_{e,k}$ denotes a component posterior, while q_e^{agg} and P_e^{agg} denote the aggregate training and deployment laws. Their moments are (μ_e^q, Σ_e^q) and (μ_e^P, Σ_e^P), respectively. Prototype priors P_k remain distinct from prototype embeddings u_k. We use P_e^{agg} as the canonical notation for the deployed law. When a proof varies the responsibilities, $P_e^{\mathrm{agg}}(r)$ means the construction in equation 7 with r replacing r_e; thus $P_e^{\mathrm{agg}} = P_e^{\mathrm{agg}}(r_e)$. The same explicit-argument convention applies to q_e^{agg} and the aggregate moments. In the hierarchical analysis, $P_e^{\mathrm{agg}}(\cdot \mid \Theta)$ exposes the prototype-parameter tuple Θ in that same construction. These arguments are suppressed elsewhere; they do not introduce a different deployment law. The primary theoretical object is the aggregate train–deployment complexity from equation 8,

$$\mathcal{C}_e^{\mathrm{agg}} := D_{\mathrm{KL}}(q_e^{\mathrm{agg}} \| P_e^{\mathrm{agg}}).$$

It is distinct from categorical-mixture KL $\mathcal{C}_e^{\mathrm{mix}}$, the unweighted component sum $\mathcal{C}_e^{\mathrm{comp}}$, the weighted pooling penalty $\mathcal{L}_{\mathrm{pool},e}$, and the cross-prototype reference complexity $\mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e)$. Appendix A.2.1 defines the aggregate and component quantities; Appendix A.4.2 gives the complementary mixture-reference definitions and comparisons.

A.2 AGGREGATE LATENT COMPLEXITY

This subsection first establishes the aggregate-latent characterization in Section 3.4: the exact Gaussian KL, its data-processing upper bound, and the one-hot equality with uncapped pooling. These statements hold for every fixed observed context, component parameters, and common responsibility vector, with independent component draws. They require no bounded-loss, within-entity sampling, or sample-independence assumptions. The PAC-Bayes guarantee and its optional extensions follow separately, under additional assumptions, and concern posterior-sampled risk for a single entity-level latent law.

A.2.1 AGGREGATE COMPLEXITY AND EXACT GAUSSIAN KL

The aggregate complexity measures the discrepancy between the context-dependent aggregate latent law used during training and the shared prototype-induced aggregate law used at deployment. With the laws in Section 3.3, distinguish

$$\begin{aligned}
\mathcal{C}_e^{\mathrm{agg}} &= D_{\mathrm{KL}}(q_e^{\mathrm{agg}} \| P_e^{\mathrm{agg}}) && \text{(aggregate train–deployment KL)},\\
D_{e,k}^{\mathrm{comp}} &= D_{\mathrm{KL}}(q_{e,k} \| P_k) && \text{(component KL)},\\
\mathcal{C}_e^{\mathrm{comp}} &= \sum_k D_{e,k}^{\mathrm{comp}} && \text{(unweighted component sum)},\\
\mathcal{L}_{\mathrm{pool},e} &= \sum_k r_{ek} D_{e,k}^{\mathrm{comp}} && \text{(uncapped weighted pooling)}.
\end{aligned} \tag{13}$$

The following results compare these quantities for the same context and component laws. They apply pointwise even when the amortized posterior and router depend on that context. The training penalty with an optional cap $\mathcal{L}_{\mathrm{pool},e}^{\mathrm{train}}$ and the window-averaged diagnostic D_e^{proto} retain their separate definitions in equation 61 and equation 74. The aggregate prior is the law induced by CLASP's weighted sum of independent prototype draws. In general,

$$P_e^{\mathrm{agg}} \neq P_{r_e}^{\mathrm{mix}} := \sum_k r_{ek} P_k.$$

Table 7: Notation used throughout the paper.

Symbol	Meaning	Defined
x_e, y_e, m_e	history, future target, static metadata	§3.1
T, H	history length and forecast horizon	§3.1
d_z	latent dimension, with $z_e \in \mathbb{R}^{d_z}$	§3.1
$h_e, h_e(x)$	context summary; fixed context map evaluated at x	§3.1; §A.2.5
c_e, r_e, r_{ek}	provided cluster label, responsibility vector, component	§3.1
p_c	fixed uniform categorical reference for the router	§B.2
Θ_k, P_k	prototype parameters (μ_k, Σ_k) and Gaussian prototype prior	§3.1
$q_{e,k}$	component posterior $\mathcal{N}(\mu_{e,k}, \Sigma_{e,k})$	equation 2
$q_e^{\mathrm{agg}}, \mu_e^q, \Sigma_e^q$	aggregate posterior law and its mean and covariance	equation 6
P_e^{agg}	aggregate deployment prior from weighted prototype draws	equation 7
$P_{r_e}^{\mathrm{mix}}$	categorical reference mixture $\sum_k r_{ek} P_k$	equation 44
z_e	aggregate latent: q_e^{agg} in training, P_e^{agg} at deployment; mean-only retrofit exceptions	§3.5; §C.2
M_e	encoder sequence states for cross-attention	§3.5
u_k, u_e, g_e	prototype embedding, weighted mean, $g_e = [h_e, u_e, z_e]$	§3.1; §3.5
$\tau, v, \hat{v}_\phi$	flow time, target velocity, velocity network	§B.1.1
T_{gen}	generative training time-grid size	§B.1.1
$\alpha, \alpha_{\mathrm{diff}}, S$	flow/diffusion reconstruction weights; draws per window	§B.1.1–B.1.2
$\lambda_1, \lambda_2; \widetilde{y}_e^{\mathrm{point}}$	readout blend weights; decoded sample-mean forecast	equation 12; equation 68
$\mathcal{L}_{\mathrm{pool},e}, \mathcal{L}_{\mathrm{pool}}$	uncapped componentwise pooling penalty and training-context mean	equation 4
$\mathcal{C}_e^{\mathrm{agg}}, \mathcal{C}_e^{\mathrm{comp}}$	aggregate latent KL and its unweighted component-KL upper bound	§3.4
$\mathcal{L}_{\mathrm{pool}}^{\mathrm{train}}, \kappa_z$	training penalty and its component-KL cap	equation 61
$\mathcal{L}_{\mathrm{cluster}}, \eta_{\mathrm{ent}}$	router KL and batch-entropy reward weight	§B.2
F_ϕ	forecasting backbone with fixed conditioning maps	equation 11
$Q_e, \mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e)$	certified law and cross-prototype reference complexity	§A.2.5; equation 44
$R_e(Q_e), \widehat{R}_e(Q_e)$	population and posterior-averaged empirical risks	equation 19
$R_e^{\mathrm{fix}}, \widehat{R}_e^{\mathrm{fix}}$	population and sampled empirical risks of the complete fixed predictor	equation 53
n_e	number of certified windows for entity e	§A.2.5
D_e^{proto}	raw component-KL diagnostic, window-averaged in the held-out evaluation	equation 74
$U_e(a)$	fixed-predictor binary-KL upper bound with nonnegative slack a	equation 73
B, ε	reference-divergence budget and simplex-net radius	§A.4.1–A.4.2
$C_{\mathrm{net}}, C_\varepsilon$	logarithmic cover term and total cover-transfer complexity	equation 47
N_k^{eff}	effective support $\sum_e r_{ek}$ of prototype k	Prop. 2
$\pi, \rho, C_{\mathrm{hyp}}(\rho)$	prototype hyper-prior, hyper-posterior, their KL	§A.4.3
R_{post}	environment risk of the fixed posterior-producing rule	equation 49

The categorical mixture on the right selects one prototype rather than aggregating independent draws. It serves only as an auxiliary reference in Appendix A.4.2 and is not needed for the aggregate characterization below.

Exact aggregate Gaussian KL.

Proposition 1 (Exact aggregate Gaussian KL). *For the laws in equation 6 and equation 7, assume* $\Sigma_e^q, \Sigma_e^P \succ 0$ *and let* $\Delta_e = \mu_e^q - \mu_e^P$. *Then*

$$\mathcal{C}_e^{\mathrm{agg}} = \frac{1}{2}\Big[\operatorname{tr}\big((\Sigma_e^P)^{-1}\Sigma_e^q\big) + \Delta_e^\top (\Sigma_e^P)^{-1}\Delta_e - d_z + \ln \frac{\det \Sigma_e^P}{\det \Sigma_e^q}\Big]. \tag{14}$$

Proof. For $z \sim q_e^{\text{agg}}$ and $\Delta_e = \mu_e^q - \mu_e^P$, $\mathbb{E}[(z-\mu_e^P)(z-\mu_e^P)^\top] = \Sigma_e^q + \Delta_e\Delta_e^\top$. Taking the expectation of the Gaussian log-density ratio gives equation 14; evaluating its trace, quadratic form, and log-determinant coordinatewise gives equation 15. □

For diagonal covariance matrices, let $\Sigma_e^q = \text{diag}(v_{e,1}^q, \ldots, v_{e,d_z}^q)$ and $\Sigma_e^P = \text{diag}(v_{e,1}^P, \ldots, v_{e,d_z}^P)$, where $v_{e,j}^q, v_{e,j}^P > 0$ are variances. Then

$$D_{\text{KL}}(q_e^{\text{agg}} \,\|\, P_e^{\text{agg}}) = \frac{1}{2}\sum_{j=1}^{d_z}\left[\frac{v_{e,j}^q}{v_{e,j}^P} + \frac{(\mu_{e,j}^q - \mu_{e,j}^P)^2}{v_{e,j}^P} - 1 + \log\frac{v_{e,j}^P}{v_{e,j}^q}\right]. \tag{15}$$

Appendix E.4.1 (Table 34) evaluates these aggregate divergences and illustrates the one-hot identity of Corollary 1. These measurements describe latent discrepancy rather than forecasting risk.

A.2.2 Data-processing bound and proof

Both joint component laws pass through the same deterministic aggregation map, under which KL divergence cannot increase. Theorem 1 therefore gives

$$\boxed{\mathcal{C}_e^{\text{agg}} \le \mathcal{C}_e^{\text{comp}} = \sum_{k=1}^{K} D_{\text{KL}}(q_{e,k} \| P_k).}$$

The upper bound uses the *unweighted* component sum. The soft responsibility-weighted penalty $\mathcal{L}_{\text{pool},e}$ has no general ordering with $\mathcal{C}_e^{\text{agg}}$; the examples immediately below the proof show both directions.

Define the product component laws

$$Q_e^{\text{joint}} = \bigotimes_{k=1}^{K} q_{e,k}, \qquad P^{\text{joint}} = \bigotimes_{k=1}^{K} P_k. \tag{16}$$

Let

$$T_{r_e}(z_1, \ldots, z_K) = \sum_{k=1}^{K} r_{ek} z_k.$$

Then

$$q_e^{\text{agg}} = (T_{r_e})_{\#} Q_e^{\text{joint}}, \qquad P_e^{\text{agg}} = (T_{r_e})_{\#} P^{\text{joint}},$$

where $(T_{r_e})_{\#}$ denotes pushforward by T_{r_e}.

Proof. The mapping T_{r_e} is deterministic and measurable. The data-processing inequality for KL divergence therefore gives

$$D_{\text{KL}}\big((T_{r_e})_{\#} Q_e^{\text{joint}} \,\big\|\, (T_{r_e})_{\#} P^{\text{joint}}\big) \le D_{\text{KL}}\big(Q_e^{\text{joint}} \,\big\|\, P^{\text{joint}}\big). \tag{17}$$

Because both joint laws factorize across components,

$$\begin{aligned} D_{\text{KL}}\big(Q_e^{\text{joint}} \,\big\|\, P^{\text{joint}}\big) &= D_{\text{KL}}\left(\bigotimes_{k=1}^{K} q_{e,k} \,\middle\|\, \bigotimes_{k=1}^{K} P_k\right) \\ &= \sum_{k=1}^{K} D_{\text{KL}}(q_{e,k} \,\|\, P_k). \end{aligned} \tag{18}$$

Combining the two displays yields equation 9. □

A.2.3 Soft pooling has no general ordering

Remark 1 (Relation to the training pooling objective). The unweighted component sum $\mathcal{C}_e^{\text{comp}}$ and uncapped weighted penalty $\mathcal{L}_{\text{pool},e}$ are defined in equation 13. Since $0 \le r_{ek} \le 1$ and component KLs are nonnegative, $\mathcal{L}_{\text{pool},e} \le \mathcal{C}_e^{\text{comp}}$. Thus both aggregate KL and weighted pooling are bounded by the component sum, but they have no general ordering with each other. For example, with $r_e = (1/2, 1/2)$, $q_{e,1} = q_{e,2} = \mathcal{N}(a, 1)$, and $P_1 = P_2 = \mathcal{N}(0, 1)$, the weighted penalty is $a^2/2$ while the aggregate KL is a^2. Replacing $q_{e,2}$ by $\mathcal{N}(-a, 1)$ leaves the penalty at $a^2/2$ but makes the aggregate KL zero. Here $a \neq 0$.

A.2.4 EXACT ONE-HOT THEORY–ALGORITHM CORRESPONDENCE

Corollary 1 identifies the aggregate discrepancy with the selected-component pooling penalty:

$$\boxed{r_e = \mathrm{onehot}(c_e) \quad \Longrightarrow \quad \mathcal{C}_e^{\mathrm{agg}} = D_{\mathrm{KL}}(q_{e,c_e} \| P_{c_e}) = \mathcal{L}_{\mathrm{pool},e}.}$$

Proof. One-hot responsibilities have $r_{e,c_e} = 1$ and $r_{e,k} = 0$ for $k \neq c_e$. Substitution into equation 6 and equation 7 gives $q_e^{\mathrm{agg}} = q_{e,c_e}$ and $P_e^{\mathrm{agg}} = P_{c_e}$, so aggregate KL equals $D_{\mathrm{KL}}(q_{e,c_e} \| P_{c_e})$. Substitution into the uncapped pooling definition equation 4 leaves exactly the same selected component KL, proving equation 10. □

For provided one-hot assignments and an inactive cap, the per-context quantity optimized by the pooling regularizer is therefore exactly the aggregate train–deployment latent KL. If the cap is active,

$$\mathcal{L}_{\mathrm{pool},e}^{\mathrm{train}} = \min\{\mathcal{C}_e^{\mathrm{agg}}, \kappa_z\}.$$

This is an exact correspondence between the theory and the training objective, independent of the PAC-Bayes assumptions below; it is not by itself a forecasting-risk guarantee. The unweighted $\mathcal{C}_e^{\mathrm{comp}}$ still includes unselected components and need not equal the selected-component KL.

A.2.5 PAC-BAYES SETUP: RISKS AND ADDITIONAL ASSUMPTIONS

For the conditional generalization guarantee, consider responsibilities r_e fixed independently of the certified sample. This includes an externally provided entity label and a fixed soft responsibility vector, provided the prototype priors and prediction map also satisfy the independence conditions below.

For these risk certificates, $Q_e = q_e^{\mathrm{agg}}$ denotes *one entity-level latent law*, which may be selected using S_e but is used unchanged across the windows in both risks. This is an additional statistical setup: CLASP's amortized posterior and inferred responsibilities depend on each observed window. A frozen encoder or router alone does not identify those window-dependent laws with this single Q_e.

Let $S_e = \{(x_e^{(i)}, y_e^{(i)})\}_{i=1}^{n_e}$ denote the certified samples for entity e. Write $h_e(x) = f_\theta(x, m_e)$ and retain the aggregate prototype embedding u_e in the conditioning vector of equation 11. For a latent law Q_e, define

$$R_e(Q_e) = \mathbb{E}_{z\sim Q_e}\mathbb{E}_{(x,y)\sim D_e}\left[\ell(F_\phi(x, [h_e(x), u_e, z]), y)\right], \tag{19}$$

$$\widehat{R}_e(Q_e) = \mathbb{E}_{z\sim Q_e}\frac{1}{n_e}\sum_{i=1}^{n_e}\ell\Big(F_\phi(x_e^{(i)}, [h_e(x_e^{(i)}), u_e, z]), y_e^{(i)}\Big). \tag{20}$$

The expectation may include all generative randomness other than the latent variable explicitly displayed. These risk functionals are shared by the aggregate and mixture-reference certificates.

Assumption 1 (Fixed prediction map)**.** *Conditionally on the entity and its exogenous metadata, the measurable prediction map F_ϕ, context map $h_e(x)$, aggregate prototype embedding u_e, and all other conditioning and readout rules are fixed independently of S_e. For each fixed z, they define a fixed loss function of (x, y). Selecting Q_e or an entity-level responsibility vector holds u_e and the other conditioning rules fixed.*

Assumption 2 (Bounded loss)**.** *The loss satisfies*

$$0 \leq \ell(\widehat{y}, y) \leq 1.$$

Any normalization or clipping threshold is fixed independently of the certified sample.

Assumption 3 (Within-entity sampling)**.** *Conditionally on entity e, the certified windows $\{(x_e^{(i)}, y_e^{(i)})\}_{i=1}^{n_e}$ are i.i.d. from D_e. Temporal spacing or non-overlap alone does not establish this condition.*

Assumption 4 (Fixed aggregate deployment prior)**.** *Conditionally on the entity and its exogenous metadata, the prototype parameters $\{(\mu_k, \Sigma_k)\}_{k=1}^K$ and responsibility vector r_e are fixed independently of S_e. Thus P_e^{agg} is a sample-independent PAC-Bayes prior. Prototypes learned on separate training entities may satisfy this condition; freezing parameters after fitting them on S_e does not.*

A.2.6 PAC-BAYES GUARANTEE FOR FIXED RESPONSIBILITIES

Use the single entity-level law and risks defined in Appendix A.2.5. The law may depend on S_e, but is used unchanged across its empirical and population risk inputs. The prediction map, prototype parameters, and responsibilities are sample-independent under the stated assumptions.

Theorem 2 (Aggregate latent PAC-Bayes certificate for a fixed entity-level latent law)**.** *Under Assumptions 1–4 and the single-law setup above, let $n_e \geq 8$ and $0 < \delta < 1$. With probability at least $1 - \delta$ over S_e, simultaneously for all such posterior laws,*

$$\mathrm{kl}\Big(\widehat{R}_e(q_e^{\mathrm{agg}}) \,\Big\|\, R_e(q_e^{\mathrm{agg}})\Big) \leq \frac{\mathcal{C}_e^{\mathrm{agg}} + \ln(2\sqrt{n_e}/\delta)}{n_e}. \tag{21}$$

Consequently,

$$R_e(q_e^{\mathrm{agg}}) \leq \widehat{R}_e(q_e^{\mathrm{agg}}) + \sqrt{\frac{\mathcal{C}_e^{\mathrm{agg}} + \ln(2\sqrt{n_e}/\delta)}{2n_e}}. \tag{22}$$

Proof. Under Assumptions 1–4, the aggregate deployment prior P_e^{agg} is fixed independently of S_e, whereas q_e^{agg} may depend on S_e. Apply the standard PAC-Bayes inequality equation 43 with prior P_e^{agg} and posterior q_e^{agg}, uniformly over posterior laws for the fixed prediction map. Substituting equation 8 gives equation 21. Pinsker's inequality, $\mathrm{kl}(a\|b) \geq 2(a-b)^2$, then gives equation 22 on the same event. □

The certificate controls posterior-sampled risk $R_e(q_e^{\mathrm{agg}})$, not deployed prior-sampled risk $R_e(P_e^{\mathrm{agg}})$. Its aggregate KL measures latent mismatch but does not establish that replacing posterior draws improves forecasts. Under the certificate's assumptions and one-hot assignments, Corollary 1 identifies its reference complexity with the uncapped pooling penalty.

Scope and extensions. The implementation's amortized posterior and inferred responsibilities depend on each observed window. Freezing their neural networks does not turn these window-dependent laws into the single entity-level law required above; Remark 3 describes a distinct fixed-predictor argument. Likewise, freezing a prior after fitting it on the certified sample does not establish sample independence. Appendix A.2.7 gives the optional certificate for a data-dependent *entity-level* responsibility vector, requiring sample-independent prototypes and uniform bounds on Gaussian means and covariance eigenvalues. Appendix A.2.8 gives the hierarchical extension with an explicit joint change of measure and hyper-posterior expectation. Both retain the fixed-map conditions; the hierarchical extension additionally requires a fixed posterior-producing rule. Neither automatically certifies adaptation of the deployed sampler on the certified entities. Ensemble readouts additionally require the conditions in Remark 4.

Component-KL relaxation. The unweighted data-processing bound also yields a conditional posterior-risk guarantee under the same PAC-Bayes assumptions.

Corollary 2 (Component-KL aggregate certificate)**.** *Under the assumptions of Theorem 2, with probability at least $1 - \delta$,*

$$R_e(q_e^{\mathrm{agg}}) \leq \widehat{R}_e(q_e^{\mathrm{agg}}) + \sqrt{\frac{\displaystyle\sum_{k=1}^{K} D_{\mathrm{KL}}(q_{e,k}\|P_k) + \ln\left(\frac{2\sqrt{n_e}}{\delta}\right)}{2n_e}}. \tag{23}$$

Proof. On the common PAC-Bayes event in Theorem 2, substitute $\mathcal{C}_e^{\mathrm{agg}} \leq \mathcal{C}_e^{\mathrm{comp}}$ from Theorem 1 into equation 22. Monotonicity of the square root gives equation 23 without an additional failure-probability allocation. □

A.2.7 OPTIONAL EXTENSION: SOFT RESPONSIBILITIES VIA A SIMPLEX COVER

An inferred responsibility vector can depend on the certified sample. For a *single entity-level vector* $r_e = r_e(S_e)$ and a sample-independent prototype tuple, a simplex net and continuity bound allow

data-dependent selection of the aggregate deployment prior. This extension retains the fixed prediction map and single-law risk definitions in Appendix A.2.5. It does not treat the implemented per-window router $r_e(x)$ as one fixed vector across all risk inputs.

Norm conventions. Vector norms $\|\cdot\|_2$ are Euclidean; matrix norms $\|\cdot\|_{\mathrm{op}}$ and $\|\cdot\|_F$ are the spectral operator and Frobenius norms. Covariance derivatives use the Frobenius inner product $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$ on symmetric matrices. The next two lemmas are deterministic consequences of the parameter bounds; sample independence is required only when they are used in the subsequent risk certificate.

Assumption 5 (Uniform latent parameter bounds)**.** *Fix constants $0 \le B_\mu < \infty$ and $0 < m \le M < \infty$. Every admissible entity-level component posterior and prototype prior satisfies*

$$\|\mu_{e,k}\|_2 \le B_\mu, \qquad \|\mu_k\|_2 \le B_\mu, \tag{24}$$

and

$$mI \preceq \Sigma_{e,k}, \Sigma_k \preceq MI. \tag{25}$$

Lemma 1 (Aggregate covariance bounds)**.** *Under Assumption 5, for every $r \in \Delta^{K-1}$,*

$$\frac{m}{K} I \preceq \sum_{k=1}^{K} r_k^2 \Sigma_{e,k} \preceq MI, \tag{26}$$

and

$$\frac{m}{K} I \preceq \sum_{k=1}^{K} r_k^2 \Sigma_k \preceq MI. \tag{27}$$

Moreover,

$$\|\mu_e^q(r) - \mu_e^q(\tilde{r})\|_2 \le B_\mu \|r - \tilde{r}\|_1, \tag{28}$$

and

$$\left\|\mu_e^P(r) - \mu_e^P(\tilde{r})\right\|_2 \le B_\mu \|r - \tilde{r}\|_1, \tag{29}$$

while

$$\left\|\Sigma_e^P(r) - \Sigma_e^P(\tilde{r})\right\|_F \le 2\sqrt{d_z}\, M \|r - \tilde{r}\|_1. \tag{30}$$

Proof. Since

$$\sum_{k=1}^{K} r_k^2 \ge \frac{1}{K}$$

for every $r \in \Delta^{K-1}$, the lower covariance bounds follow from

$$\sum_k r_k^2 \Sigma_k \succeq m \left(\sum_k r_k^2 \right) I \succeq \frac{m}{K} I,$$

and the same argument applies to the posterior aggregate. The upper bounds use $\sum_k r_k^2 \le 1$.

For the means,

$$\begin{aligned} \|\mu_e^P(r) - \mu_e^P(\tilde{r})\|_2 &= \left\| \sum_k (r_k - \tilde{r}_k) \mu_k \right\|_2 \\ &\le \sum_k |r_k - \tilde{r}_k| \|\mu_k\|_2 \\ &\le B_\mu \|r - \tilde{r}\|_1, \end{aligned}$$

and the posterior bound is identical.

Finally,

$$|r_k^2 - \tilde{r}_k^2| = |r_k - \tilde{r}_k|(r_k + \tilde{r}_k) \le 2|r_k - \tilde{r}_k|,$$

while $\|\Sigma_k\|_F \le \sqrt{d_z} M$. Therefore

$$\begin{aligned} \left\|\Sigma_e^P(r) - \Sigma_e^P(\tilde{r})\right\|_F &\le \sum_k |r_k^2 - \tilde{r}_k^2| \|\Sigma_k\|_F \\ &\le 2\sqrt{d_z} M \|r - \tilde{r}\|_1. \end{aligned}$$

□

Lemma 2 (Prior-argument Lipschitz continuity of the aggregate Gaussian KL)**.** *Under Assumption 5, there exists a finite constant $L_{\mathrm{agg}} = L_{\mathrm{agg}}(B_\mu, m, M, K, d_z)$ such that, for all $r, \tilde{r} \in \Delta^{K-1}$, holding the posterior argument $q_e^{\mathrm{agg}}(r)$ fixed,*

$$|D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r) \,\|\, P_e^{\mathrm{agg}}(r)) - D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r) \,\|\, P_e^{\mathrm{agg}}(\tilde{r}))| \le L_{\mathrm{agg}} \|r - \tilde{r}\|_1. \tag{31}$$

One valid explicit choice is

$$\begin{aligned} L_{\mathrm{agg}} = & \frac{2KB_\mu^2}{m} \\ & + \sqrt{d_z} M \left[\frac{\sqrt{d_z} K}{m} + \frac{\sqrt{d_z} MK^2}{m^2} + \frac{4B_\mu^2 K^2}{m^2} \right]. \end{aligned} \tag{32}$$

Proof. Fix the posterior aggregate $q_e^{\mathrm{agg}}(r)$ and view the Gaussian KL as a function of the prior parameters (μ_P, Σ_P). The Gaussian KL is differentiable on the set of positive-definite covariance matrices satisfying

$$\frac{m}{K} I \preceq \Sigma_P \preceq MI.$$

Its derivatives with respect to the prior mean and covariance are

$$\nabla_{\mu_P} D_{\mathrm{KL}}(q\|P) = \Sigma_P^{-1}(\mu_P - \mu_q) \tag{33}$$

and

$$\nabla_{\Sigma_P} D_{\mathrm{KL}}(q\|P) = \frac{1}{2} \left[\Sigma_P^{-1} - \Sigma_P^{-1}\Sigma_q\Sigma_P^{-1} - \Sigma_P^{-1}(\mu_P - \mu_q)(\mu_P - \mu_q)^\top \Sigma_P^{-1} \right]. \tag{34}$$

Use $\|ABC\|_F \le \|A\|_{\mathrm{op}} \|B\|_F \|C\|_{\mathrm{op}}$ and $\|vv^\top\|_F = \|v\|_2^2$. The covariance bounds imply

$$\|\Sigma_P^{-1}\|_{\mathrm{op}} \le \frac{K}{m}, \qquad \|\Sigma_q\|_F \le \sqrt{d_z} M,$$

while Assumption 5 gives

$$\|\mu_P - \mu_q\|_2 \le 2B_\mu.$$

Hence

$$\|\nabla_{\mu_P} D_{\mathrm{KL}}\|_2 \le \frac{2KB_\mu}{m}, \tag{35}$$

and

$$\|\nabla_{\Sigma_P} D_{\mathrm{KL}}\|_F \le \frac{1}{2} \left[\frac{\sqrt{d_z} K}{m} + \frac{\sqrt{d_z} MK^2}{m^2} + \frac{4B_\mu^2 K^2}{m^2} \right]. \tag{36}$$

Keep $q_e^{\mathrm{agg}}(r)$ fixed and consider the straight segment

$$(\mu_t, \Sigma_t) = (1-t)(\mu_e^P(r), \Sigma_e^P(r)) + t(\mu_e^P(\tilde{r}), \Sigma_e^P(\tilde{r})), \qquad 0 \le t \le 1.$$

Its endpoints lie in the convex enclosing Gaussian parameter set

$$\mathcal{D} = \{(\mu, \Sigma) : \|\mu\|_2 \le B_\mu,\ (m/K)I \preceq \Sigma \preceq MI\}.$$

The entire segment therefore stays in $\mathcal{D}$, where the displayed gradient bounds hold. The segment need not itself be realizable by responsibility vectors under the fixed prototype bank; only membership in this enclosing set is used. Integrating the gradients along this segment and combining them with equation 29 and equation 30 gives equation 31 with the constant in equation 32. □

Theorem 3 (Soft-responsibility certificate for a fixed entity-level latent law)**.** *Assume Assumptions 1–3 and 5, with the prototype tuple fixed independently of S_e. The constants B_μ, m, M are also fixed independently of S_e. Let $n_e \ge 8$, $0 < \delta < 1$, and fix $0 < \epsilon \le 1$ independently of S_e. Let $\mathcal{N}_\epsilon \subset \Delta^{K-1}$ be a sample-independent ℓ_1 ϵ-net satisfying*

$$|\mathcal{N}_\epsilon| \le \left(\frac{5}{\epsilon} \right)^{K-1}.$$

Such a net exists by the construction in Appendix A.6. With probability at least $1-\delta$ over S_e, simultaneously for all entity-level responsibility vectors r_e and component posterior laws satisfying the bounds, including data-dependent choices,

$$R_e(q_e^{\mathrm{agg}}(r_e)) \le \widehat{R}_e(q_e^{\mathrm{agg}}(r_e)) + \sqrt{\frac{D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r_e) \,\|\, P_e^{\mathrm{agg}}(r_e)) + C_\epsilon^{\mathrm{agg}} + \ln\left(\frac{2\sqrt{n_e}}{\delta}\right)}{2n_e}}, \tag{37}$$

where

$$C_\epsilon^{\mathrm{agg}} = L_{\mathrm{agg}}\epsilon + (K-1)\ln\left(\frac{5}{\epsilon}\right). \tag{38}$$

Proof. For every fixed $\tilde{r} \in \mathcal{N}_\epsilon$, the Gaussian $P_e^{\mathrm{agg}}(\tilde{r})$ is sample-independent because both the net and prototype tuple are fixed independently of S_e. Apply the standard PAC-Bayes inequality to each such prior at confidence $\delta/|\mathcal{N}_\epsilon|$, uniformly over posterior laws, and take a union bound over the net. On the resulting common event, choose any realized entity-level r_e and admissible component posteriors, then select a nearest net point $\tilde{r}_e$ with $\|r_e - \tilde{r}_e\|_1 \le \epsilon$. The common event gives

$$\mathrm{kl}\Big(\widehat{R}_e(q_e^{\mathrm{agg}}(r_e)) \,\Big\|\, R_e(q_e^{\mathrm{agg}}(r_e))\Big) \le \frac{D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r_e) \,\|\, P_e^{\mathrm{agg}}(\tilde{r}_e)) + \ln|\mathcal{N}_\epsilon| + \ln(2\sqrt{n_e}/\delta)}{n_e}. \tag{39}$$

By Lemma 2,

$$D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r_e) \,\|\, P_e^{\mathrm{agg}}(\tilde{r}_e)) \le D_{\mathrm{KL}}(q_e^{\mathrm{agg}}(r_e) \,\|\, P_e^{\mathrm{agg}}(r_e)) + L_{\mathrm{agg}}\epsilon.$$

Using

$$\ln|\mathcal{N}_\epsilon| \le (K-1)\ln(5/\epsilon)$$

and applying Pinsker's inequality yields equation 37. □

Scope of the soft-router result. The soft-responsibility theorem is considerably more assumption-heavy than the fixed-responsibility result. In particular, it requires uniform bounds on the component posterior and prototype prior means and covariance eigenvalues in order to obtain a finite Lipschitz constant for the Gaussian KL. We therefore view Theorem 3 as an optional extension rather than as the primary theoretical interpretation of CLASP. These bounds are assumptions on the actual Gaussian parameters; neither component-KL capping nor clipping sampled coordinates inside the forecaster establishes them. The theorem controls posterior risk for a single entity-level law. A separate certificate is needed for window-dependent latent kernels or for the deployed prior-sampling risk. The fixed-map assumption also excludes changing the prediction map through u_e or readout-profile selection when the certified entity-level responsibility vector is selected; the present cover controls only the latent reference prior.

A.2.8 Optional Extension: Hierarchical Aggregate Certificate

The aggregate reference can also be used in a hierarchical construction, but this requires a joint change of measure rather than an automatic substitution into Theorem 5. Retain the i.i.d. environment, fixed prediction map, and fixed posterior-producing rule of Appendix A.4.3, together with bounded loss and within-entity i.i.d. sampling (Assumptions 2 and 3). The rule produces one entity-level $Q_e = q_e^{\mathrm{agg}}$ and r_e and does not read the reference prototype tuple Θ. Let the hyper-prior π be independent of all certified data and let $C_{\mathrm{hyp}}(\rho) = D_{\mathrm{KL}}(\rho\|\pi) < \infty$.

For responsibilities fixed independently of each certified entity sample, use the joint prior and posterior

$$\mathsf{P}_{r_e}(\mathrm{d}\Theta, \mathrm{d}z) = \pi(\mathrm{d}\Theta)P_e^{\mathrm{agg}}(\mathrm{d}z \mid \Theta), \qquad \mathsf{Q}_e^\rho(\mathrm{d}\Theta, \mathrm{d}z) = \rho(\mathrm{d}\Theta)Q_e(\mathrm{d}z).$$

Here P_{r_e} is a joint reference law on (Θ, z); its conditional latent law is $P_e^{\mathrm{agg}}(\cdot \mid \Theta)$. The chain rule yields

$$D_{\mathrm{KL}}(\mathsf{Q}_e^\rho \| \mathsf{P}_{r_e}) = C_{\mathrm{hyp}}(\rho) + \mathbb{E}_{\Theta\sim\rho} D_{\mathrm{KL}}(Q_e \| P_e^{\mathrm{agg}}(\cdot \mid \Theta)).$$

Because the loss map does not read Θ, its risks under Q_e^ρ equal those under Q_e. Applying the joint-prior PAC-Bayes inequality at confidence $\delta/(2N)$ for each entity, followed by the same environment-level step at confidence $\delta/2$ as in Appendix A.7, gives the within-entity term

$$\mathcal{B}_e^{\mathrm{agg}}(\rho) := \sqrt{\frac{C_{\mathrm{hyp}}(\rho) + \mathbb{E}_{\Theta\sim\rho} D_{\mathrm{KL}}(Q_e \| P_e^{\mathrm{agg}}(\cdot \mid \Theta)) + \ln(4N\sqrt{n_e}/\delta)}{2n_e}}. \tag{40}$$

Under these fixed-map, fixed-rule, and i.i.d. sampling conditions, for $N \geq 8$, $n_e \geq 8$, and $0 < \delta < 1$, the resulting bound is, with probability at least $1 - \delta$, simultaneously for admissible hyper-posteriors with finite displayed complexity,

$$R_{\mathrm{post}} \leq \frac{1}{N}\sum_{e=1}^{N} \widehat{R}_e(Q_e) + \sqrt{\frac{C_{\mathrm{hyp}}(\rho) + \ln(4\sqrt{N}/\delta)}{2N}} + \frac{1}{N}\sum_{e=1}^{N} \mathcal{B}_e^{\mathrm{agg}}(\rho). \tag{41}$$

Here R_{post} is the posterior-sampled environment risk of the independently fixed rule in equation 49, not the risk of the deployed prior-sampled predictor. The expectation over ρ is essential when the reference tuple is random. For one-hot assignments, its integrand equals the uncapped selected-component pooling penalty for the corresponding fixed posterior and reference tuple.

For data-dependent entity-level responsibilities, fix $0 < \epsilon \leq 1$ and its net independently of all certified entities, and use the fixed joint priors indexed by $\tilde{r} \in \mathcal{N}_\epsilon$. Assume the same prespecified parameter bounds hold for the certified component posteriors and ρ-almost every reference tuple, so a single L_{agg} applies. A union bound over entities and net points and the Lipschitz transfer then add $C_\epsilon^{\mathrm{agg}}$ from equation 38 to the numerator of equation 40. This construction uses the Gaussian parameter bounds in place of the mixture-reference divergence budget in Assumption 6.

These hierarchical statements retain the fixed-rule independence requirements. They do not certify joint adaptation of a prototype-dependent posterior rule or deployed sampler on the certified entities. The numerical calculations in Appendix E.4.2 instead use fixed-predictor concentration with optional diagnostic slack; they do not evaluate this aggregate construction.

A.3 Responsibility-Weighted Prototype Moment Matching

Proposition 2 (Responsibility-weighted moment matching). *Fix $\{q_{e,k}\}$ and $\{r_e\}$ for the N_{tr} training entities, with $N_k^{\mathrm{eff}} = \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} > 0$. The minimizers of the uncapped $\mathcal{L}_{\mathrm{pool}}$ over unconstrained prototype means and positive-definite covariances satisfy*

$$\begin{aligned} \mu_k^\star &= \frac{1}{N_k^{\mathrm{eff}}}\sum_{e=1}^{N_{\mathrm{tr}}} r_{ek}\mu_{e,k}, \\ \Sigma_k^\star &= \frac{1}{N_k^{\mathrm{eff}}}\sum_{e=1}^{N_{\mathrm{tr}}} r_{ek}\left[\Sigma_{e,k} + (\mu_{e,k} - \mu_k^\star)(\mu_{e,k} - \mu_k^\star)^\top\right]. \end{aligned} \tag{42}$$

For diagonal prototypes, retain only the diagonal of the covariance expression.

Thus, a prototype mean is a responsibility-weighted average of entity posterior means. Its covariance captures both within-entity posterior uncertainty and variation between those means. This characterizes the uncapped pooling objective with posteriors and responsibilities fixed; it does not imply that joint optimization of posteriors, prototypes, and inferred responsibilities exactly attains these moments. The proof below establishes stationarity and global optimality; Appendix E.2.4 examines moment residuals during training.

Proof. Fix the component posteriors $\{q_{e,k}\}$ and responsibilities $\{r_e\}$; they are treated as constants throughout. Since our implementation uses diagonal covariances, we give the coordinatewise computation; the full-covariance case follows from the same stationarity conditions applied to the matrix parameters. Write, for coordinate $j \in \{1, \ldots, d_z\}$,

$$v_{k,j} = (\Sigma_k)_{jj}, \qquad v_{e,k,j} = (\Sigma_{e,k})_{jj}, \qquad \mu_{k,j} = (\mu_k)_j, \qquad \mu_{e,k,j} = (\mu_{e,k})_j.$$

The uncapped objective $\mathcal{L}_{\mathrm{pool}}$ separates over prototypes. Its common factor $1/N_{\mathrm{tr}}$ does not affect the minimizers, so it suffices to analyze

$$\mathcal{J}_k = \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek}\, D_{\mathrm{KL}}(q_{e,k} \,\|\, P_k), \qquad N_k^{\mathrm{eff}} = \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} > 0.$$

For diagonal Gaussians, the divergence decomposes coordinatewise:

$$D_{\mathrm{KL}}(q_{e,k} \,\|\, P_k) = \frac{1}{2}\sum_{j=1}^{d_z}\left[\log\frac{v_{k,j}}{v_{e,k,j}} + \frac{v_{e,k,j} + (\mu_{e,k,j} - \mu_{k,j})^2}{v_{k,j}} - 1\right].$$

Prototype mean. Differentiating $\mathcal{J}_k$ with respect to $\mu_{k,j}$ gives

$$\frac{\partial \mathcal{J}_k}{\partial \mu_{k,j}} = \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} \, \frac{\mu_{k,j} - \mu_{e,k,j}}{v_{k,j}}.$$

Setting this to zero and using $v_{k,j} > 0$ yields

$$\mu^\star_{k,j} = \frac{1}{N_k^{\mathrm{eff}}} \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} \, \mu_{e,k,j},$$

which is the first claim after stacking coordinates.

Prototype variance. Differentiating with respect to $v_{k,j}$ gives

$$\frac{\partial \mathcal{J}_k}{\partial v_{k,j}} = \frac{1}{2} \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} \left[\frac{1}{v_{k,j}} - \frac{v_{e,k,j} + (\mu_{e,k,j} - \mu_{k,j})^2}{v_{k,j}^2} \right].$$

The positive factor $1/v_{k,j}$ cancels from the mean condition, leaving it independent of $v_{k,j}$. Thus any joint stationary point has $\mu_k = \mu^\star_k$; setting the derivative above to zero, multiplying by $2v_{k,j}^2$, and evaluating at $\mu^\star_k$ therefore yields

$$v^\star_{k,j} = \frac{1}{N_k^{\mathrm{eff}}} \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} \left[v_{e,k,j} + (\mu_{e,k,j} - \mu^\star_{k,j})^2 \right],$$

which is the diagonal projection of the full-covariance moment matrix. The unconstrained full-covariance optimum includes off-diagonal between-entity moments; a diagonal prototype retains only the displayed coordinate variances.

Global minimality. The objective $\mathcal{J}_k$ is not convex in $v_{k,j}$, so stationarity alone does not certify a minimum; we argue directly.

First, $\mathcal{J}_k \to \infty$ whenever $v_{k,j} \to 0^+$, $v_{k,j} \to \infty$, or $|\mu_{k,j}| \to \infty$. Hence a minimizer exists and is an interior stationary point.

Second, for each fixed $v_{k,j} > 0$, the map $\mu_{k,j} \mapsto \mathcal{J}_k$ is a strictly convex quadratic with curvature

$$\frac{N_k^{\mathrm{eff}}}{v_{k,j}} > 0,$$

minimized at $\mu^\star_{k,j}$, independently of $v_{k,j}$.

Third, substituting $\mu^\star_k$, the remaining function of $v_{k,j}$ equals

$$\frac{N_k^{\mathrm{eff}}}{2} \left(\log v_{k,j} + \frac{v^\star_{k,j}}{v_{k,j}} \right) + \mathrm{const},$$

and

$$t \mapsto \log t + \frac{a}{t}, \qquad a > 0,$$

is strictly decreasing on $(0, a)$ and strictly increasing on (a, ∞), with its minimum at $t = a$.

Hence these mean and variance coordinates define the unique stationary point and global minimizer in the diagonal prototype class. Feasibility holds since

$$v^\star_{k,j} \geq \frac{1}{N_k^{\mathrm{eff}}} \sum_{e=1}^{N_{\mathrm{tr}}} r_{ek} \, v_{e,k,j} > 0.$$

If the prototype scale is parameterized through log-standard deviations, the map to $v_{k,j}$ is a smooth bijection onto $(0, \infty)$ with strictly positive derivative, so stationarity and global minimality transfer unchanged. This characterization concerns the uncapped Gaussian-KL objective with $q_{e,k}$ and r_e held fixed; capping component divergences in $\mathcal{L}^{\mathrm{train}}_{\mathrm{pool}}$ need not preserve it. □

A.4 Complementary Mixture-Reference Certificates

The categorical-mixture results below are complementary reference-prior analyses; they are not required for the aggregate-latent characterization in Section 3.4. A categorical mixture selects one prototype law, whereas the aggregate Gaussian is the law of a weighted sum of independent component draws. These constructions generally give different latent laws and different KL complexities under soft responsibilities.

This subsection gives entity-level and hierarchical guarantees for posterior-sampled risk under additional PAC-Bayes assumptions. Selecting a reference prior changes the complexity in these bounds; it does not certify the risk of the deployed prior-sampled predictor. Separate fixed-predictor bounds and reference-complexity diagnostics are reported in Appendix E.4.

A.4.1 Risk definitions and assumptions

Use the single entity-level risks in equation 19 and equation 20, with the fixed-map, bounded-loss, and sampling conditions in Appendix A.2.5. For this mixture-reference analysis, any loss clipping or normalization threshold is fixed before training. Non-overlapping windows do not establish independence in a temporally dependent series (Mohri & Rostamizadeh, 2010). The additional assumption below restricts reference divergences; it is not required by the aggregate Gaussian identities.

Assumption 6 (Bounded reference divergences)**.** *For a fixed finite $B \geq 0$, certified latent laws satisfy $\max_k D_{\mathrm{KL}}(Q_e \| P_k) \leq B$. This restricts the actual reference divergences and is not implied by capping the training penalty or by using a Gaussian aggregate law.*

Under Assumptions 1, 2, and 3, for $n_e \geq 8$ and a prior P fixed independently of S_e, the PAC-Bayes inequality (McAllester, 1999; Maurer, 2004) gives, with probability at least $1-\delta$, simultaneously for all laws Q,

$$\mathrm{kl}\big(\widehat{R}_e(Q) \,\|\, R_e(Q)\big) \leq \frac{D_{\mathrm{KL}}(Q\|P) + \ln \frac{2\sqrt{n_e}}{\delta}}{n_e}, \tag{43}$$

where $\delta \in (0,1)$ and kl denotes Bernoulli relative entropy. The displayed theorems and the fixed-predictor environment-risk calculation in Appendix E.4.2 use its Pinsker relaxation; the sample-size analysis in Appendix E.4.3 uses the binary-KL inverse. The fixed-predictor specialization is described in Remark 3.

A.4.2 Reference mixtures and entity-level certificate

For prototype priors $P_1, \ldots, P_K$ and reference weights $r_e \in \Delta^{K-1}$, define

$$\begin{aligned} P_{r_e}^{\mathrm{mix}} &= \sum_{k=1}^{K} r_{ek} P_k, \\ \mathcal{C}_e^{\mathrm{mix}}(Q_e, r_e) &= D_{\mathrm{KL}}(Q_e \| P_{r_e}^{\mathrm{mix}}), \\ \mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) &= \sum_{k=1}^{K} r_{ek} D_{\mathrm{KL}}(Q_e \| P_k). \end{aligned} \tag{44}$$

The same law Q_e appears in every summand of $\mathcal{C}_e^{\mathrm{ref}}$. In contrast, the pooling penalty $\mathcal{L}_{\mathrm{pool},e} = \sum_k r_{ek} D_{\mathrm{KL}}(q_{e,k} \| P_k)$ uses a potentially different component posterior $q_{e,k}$ in each summand. Reference weights select a prior without changing the fixed prediction map.

Lemma 3 (Categorical-mixture domination)**.** *For any law Q, Gaussian laws $P_1, \ldots, P_K$, and $r \in \Delta^{K-1}$,*

$$D_{\mathrm{KL}}\left(Q \,\middle\|\, \sum_{k=1}^{K} r_k P_k\right) \leq \sum_{k=1}^{K} r_k D_{\mathrm{KL}}(Q\|P_k), \tag{45}$$

with equality for one-hot r.

Thus $\mathcal{C}_e^{\mathrm{mix}}(Q_e, r_e) \leq \mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e)$. Appendix A.5 proves this reference-mixture comparison by convexity. It does not identify either quantity with $\mathcal{C}_e^{\mathrm{agg}}$, whose reference is the weighted-sum law P_e^{agg}.

For example, let $q_{e,1} = q_{e,2} = P_1 = P_2 = \mathcal{N}(0, I_{d_z})$ and $r_e = (1/2, 1/2)$. Then $q_e^{\mathrm{agg}} = P_e^{\mathrm{agg}} = \mathcal{N}(0, I_{d_z}/2)$, while $P_{r_e}^{\mathrm{mix}} = \mathcal{N}(0, I_{d_z})$, so

$$\mathcal{C}_e^{\mathrm{agg}} = \mathcal{L}_{\mathrm{pool},e} = 0,$$
$$\mathcal{C}_e^{\mathrm{mix}}(q_e^{\mathrm{agg}}, r_e) = \mathcal{C}_e^{\mathrm{ref}}(q_e^{\mathrm{agg}}, r_e) = \frac{d_z}{2}\left(\ln 2 - \frac{1}{2}\right) > 0. \tag{46}$$

Even exact component alignment therefore does not make the mixture reference coincide with the aggregate deployment law, and neither the raw nor capped pooling penalty generally upper-bounds these mixture-reference complexities.

For the entity-level certificate, the prototypes are fixed independently of S_e. Fix $\varepsilon \in (0, 1]$ and an ℓ_1 ε-net $\mathcal{N}_\varepsilon \subset \Delta^{K-1}$ independently of the sample, with $|\mathcal{N}_\varepsilon| \le (5/\varepsilon)^{K-1}$. Define

$$C_{\mathrm{net}} = (K-1)\ln\frac{5}{\varepsilon}, \qquad C_\varepsilon = \varepsilon B + C_{\mathrm{net}}. \tag{47}$$

Theorem 4 (Entity-level certificate)**.** *Under the preceding fixed-map and fixed-prototype conditions and Assumptions 2, 3, and 6, for $n_e \ge 8$, fixed $\varepsilon \in (0, 1]$, and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over S_e, simultaneously for all $r_e \in \Delta^{K-1}$ and all Q_e satisfying $\max_k D_{\mathrm{KL}}(Q_e \| P_k) \le B$,*

$$R_e(Q_e) \le \widehat{R}_e(Q_e) + \sqrt{\frac{\mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) + C_\varepsilon + \ln\frac{2\sqrt{n_e}}{\delta}}{2n_e}}. \tag{48}$$

Appendix A.6 gives the simplex-net construction and proof. The cover makes the bound uniform over data-dependent reference weights. The controlled risk is $R_e(Q_e)$, with draws from the same entity-level posterior used in $\widehat{R}_e(Q_e)$; the bound does not control risk after replacing those draws by a mixture or aggregate prior. At $\varepsilon = 0.1$, its logarithmic contribution is $3\ln 50 \approx 11.736$ nats for $K = 4$ and $4\ln 50 \approx 15.648$ nats for $K = 5$; the additional transfer term εB is retained separately.

Appendix E.4.3 compares reference-prior complexities (Table 37) and direct uniform-mixture versus joint-law surrogate complexities (Table 38). These are complexity diagnostics, not numerical evaluations of Theorem 4.

A.4.3 Hierarchical reference-prior learning

Let $\Theta = (\Theta_1, \ldots, \Theta_K)$ be a prototype tuple with associated laws $P_1, \ldots, P_K$. A hyper-prior π over Θ is fixed independently of the certified data, while the hyper-posterior ρ may depend on those data. This construction follows shared-prior PAC-Bayes analyses (Pentina & Lampert, 2014; Amit & Meir, 2018; Rothfuss et al., 2021). The entity–sample pairs (D_e, S_e) are drawn i.i.d. from an environment $\mathcal{T}$, including their sample sizes and static metadata. Static metadata does not depend on sampled targets. Fix the prediction map and a rule $\mathcal{A} : S \mapsto (Q_S, r_S)$ independently of the certified entities; neither reads the reference tuple Θ. Define

$$R_{\mathrm{post}} = \mathbb{E}_{(D,S)\sim\mathcal{T}}\big[R_D(Q_S)\big], \qquad (Q_e, r_e) = \mathcal{A}(S_e). \tag{49}$$

An admissible hyper-posterior satisfies $C_{\mathrm{hyp}}(\rho) = D_{\mathrm{KL}}(\rho\|\pi) < \infty$ and Assumption 6 for every certified entity, ρ-almost surely. Expectations under ρ average over the reference prototype tuple, whose occurrence in $\mathcal{C}_e^{\mathrm{ref}}$ is implicit.

Theorem 5 combines an across-entity term with within-entity estimation terms and includes $C_{\mathrm{hyp}}(\rho)$ at both levels. Appendix A.7 proves the result using a joint change of measure over prototype parameters and entity latents; averaging a fixed-prototype certificate over a data-dependent ρ is insufficient. Because the prediction map and posterior-producing rule do not depend on Θ, R_{post} is independent of ρ. Reference-prior selection changes the certificate, not the predictor being certified, and the theorem does not certify adaptation of a prototype-dependent deployed sampler on the certified entities.

Theorem 5 (Hierarchical reference certificate)**.** *Under Assumptions 2, 3, and 6 and the fixed-map, fixed-rule and i.i.d. environment conditions above, let $N \ge 8$, $n_e \ge 8$, $0 < \delta < 1$, and fix $0 < \varepsilon \le 1$ independently of the certified samples. The hyper-prior is sample-independent. With probability at least $1 - \delta$, simultaneously for admissible hyper-posteriors ρ with finite $C_{\mathrm{hyp}}(\rho)$ and the stated*

divergence bound ρ-almost surely,

$$R_{\mathrm{post}} \le \frac{1}{N}\sum_{e=1}^{N}\widehat{R}_e(Q_e) + \sqrt{\frac{C_{\mathrm{hyp}}(\rho)+\ln(4\sqrt{N}/\delta)}{2N}} + \frac{1}{N}\sum_{e=1}^{N}\sqrt{\frac{C_{\mathrm{hyp}}(\rho)+\mathbb{E}_\rho\mathcal{C}_e^{\mathrm{ref}}(Q_e,r_e)+C_\varepsilon+\ln(4N\sqrt{n_e}/\delta)}{2n_e}}. \tag{50}$$

The environment-risk bounds in Appendix E.4.2 (Table 35) instead use the separate fixed-predictor construction of Remark 3; they do not instantiate this hierarchical latent-law certificate.

A.4.4 Partial pooling and aggregate latent laws

For one-hot assignments and $Q_e = q_{e,c_e}$, both mixture-reference complexities reduce to $D_{\mathrm{KL}}(q_{e,c_e}\|P_{c_e})$ and hence to $\mathcal{C}_e^{\mathrm{agg}} = \mathcal{L}_{\mathrm{pool},e}$ by Corollary 1. With an inactive cap, this is the quantity optimized by the pooling regularizer. The equality requires no PAC-Bayes assumptions; using it in a risk certificate requires the additional conditions stated above. With component posteriors and assignments fixed, Proposition 2 characterizes the prototypes minimizing this local objective, not the complete hierarchical certificate. Appendix E.2.4 reports the associated moment diagnostics.

Corollary 3 (Responsibility-weighted complexity decomposition)**.** *Let $n_{\min} = \min_e n_e$ and $n_{\max} = \max_e n_e$. For each prototype with $N_k^{\mathrm{eff}} = \sum_e r_{ek} > 0$, define*

$$\overline{D}_k(\rho) = \frac{1}{N_k^{\mathrm{eff}}}\sum_{e=1}^{N} r_{ek}\,\mathbb{E}_\rho\big[D_{\mathrm{KL}}(Q_e\|P_k)\big]. \tag{51}$$

On the event of Theorem 5,

$$R_{\mathrm{post}} \le \frac{1}{N}\sum_{e=1}^{N}\widehat{R}_e(Q_e) + \sqrt{\frac{C_{\mathrm{hyp}}(\rho)+\ln\frac{4\sqrt{N}}{\delta}}{2N}} + \sqrt{\frac{C_{\mathrm{hyp}}(\rho)+C_\varepsilon+\ln\frac{4N\sqrt{n_{\max}}}{\delta}+\sum_{k:N_k^{\mathrm{eff}}>0}\frac{N_k^{\mathrm{eff}}}{N}\overline{D}_k(\rho)}{2n_{\min}}}. \tag{52}$$

For one-hot assignments, $\overline{D}_k(\rho)$ is the mean reference divergence among entities assigned to prototype k. With all other quantities and responsibility weights fixed, reducing these divergences cannot loosen the relaxed certificate.

Appendix A.8 proves this algebraic relaxation using Jensen's inequality. The weights N_k^{eff}/N regroup the average reference divergence; larger prototype support alone does not imply a tighter certificate.

Soft aggregates. For soft assignments, the reference-mixture comparison in equation 45 and the aggregate data-processing bound in Theorem 1 concern different pairs of laws. Applying Theorem 4 with $Q_e = q_e^{\mathrm{agg}}$ requires separate verification of its fixed-map conditions and reference divergences; it bounds posterior-sampled risk using $\mathcal{C}_e^{\mathrm{ref}}(q_e^{\mathrm{agg}}, r_e)$ rather than $\mathcal{C}_e^{\mathrm{agg}}$ or the soft pooling penalty. For one-hot r_e and $Q_e = P_{c_e}$, the local complexity is zero, but admissibility still requires all cross-prototype divergences to satisfy Assumption 6.

A.4.5 Window-level prediction and decoding

Remark 2 (Backbone independence). The certificates require a fixed measurable prediction map, bounded loss, and the stated conditions on latent laws and reference priors. They impose no additional restriction on the backbone architecture. All context-dependent conditioning pathways remain part of the fixed prediction map.

Remark 3 (Window-level amortization). The implementation recomputes $q_e^{\mathrm{agg}}(\cdot \mid x)$ and, when inferred, $r_e(x)$ for each context window, whereas Theorems 4, 2, and 3 assign one latent law Q_e to the entity. When the encoder, router, prototypes, backbone, and any decoding parameters are fixed before the certified entity is observed, the complete procedure is a fixed randomized predictor. Write it as $\mathsf{F}(x, m_e; \xi)$, where ξ contains all latent and generative sampling randomness. Its population risk and sampled empirical risk are

$$R_e^{\mathrm{fix}} = \mathbb{E}_{(x,y)\sim D_e,\,\xi}\left[\ell\big(\mathsf{F}(x, m_e; \xi), y\big)\right],$$
$$\widehat{R}_e^{\mathrm{fix}} = \frac{1}{n_e}\sum_{i=1}^{n_e} \ell\big(\mathsf{F}(x_e^{(i)}, m_e; \xi_i), y_e^{(i)}\big). \tag{53}$$

Under Assumptions 2 and 3, with fresh independent ξ_i for each window, equation 43 applies to the augmented observations $(x_e^{(i)}, y_e^{(i)}, \xi_i)$ on a singleton hypothesis space. The prior and posterior both put unit mass on the complete predictor, so their hypothesis KL is zero. This argument does not set the latent reference divergences to zero or identify the component training penalty with C_e^{ref}. It can cover a deployed prior-sampling rule when that entire rule satisfies the stated independence conditions; this is a separate fixed-predictor guarantee, not a consequence of a posterior-sampled latent-law certificate.

Appendix E.4.2 extends this fixed-predictor construction to environment risk under its additional i.i.d. entity-sampling assumption (Table 35); Appendix E.4.3 reports entity-level sample-size sensitivity with Monte Carlo correction (Table 36). Both retain the stated sampling assumptions and use reference scores only as optional nonnegative additions to the bounds.

Remark 4 (Certificates for decoded and ensemble predictions). Blend profiles are fitted on validation entities and fixed before the certified entity is observed. The affine blend and ordinary sample-mean readout follow a fixed rule applied to the generated ensemble. Anchors and prototype-indexed profile selection are fixed functions of each window's context. This allows an unseen entity to inherit a correction without refitting and makes the forecast readout a fixed composition with the forecasting map. Appendix D.1.1 specifies the fitting and readout procedures. For any latent-law certificate, the decoded map must preserve the corresponding theorem's independence and single-law requirements. The mixture-reference certificate in Theorem 4 additionally requires Assumption 6; the aggregate certificates use the conditions stated in Theorems 2 and 3. These conditions do not follow automatically from using the aggregate deployment prior. For an ensemble of S independently redrawn certified latents, a posterior-based analysis uses $Q_e^{\otimes S}$ and the corresponding product-prior complexity rather than reusing a single-draw bound. Alternatively, when the complete ensemble or interval procedure is fixed independently of the certified entity, the singleton argument of Remark 3 applies to its specified bounded loss. Single-trajectory loss, ensemble-point loss, and interval-miscoverage loss are distinct quantities and require separate identification.

A.5 Proof of Lemma 3

Proof of Lemma 3. If the right-hand side is infinite, the claim is immediate. Otherwise, let q and p_k denote the densities of Q and P_k with respect to a common dominating measure. Pointwise in z, convexity of $-\log$ gives

$$-\log\left(\sum_k r_k p_k(z)\right) \le -\sum_k r_k \log p_k(z).$$

Multiplying by $q(z)$, integrating, and adding $\int q(z)\log q(z)\,\mathrm{d}z$ to both sides yields

$$D_{\mathrm{KL}}\big(Q \,\big\|\, P_r^{\mathrm{mix}}\big) \le \sum_k r_k D_{\mathrm{KL}}(Q\|P_k).$$

When r is one-hot, the mixture equals its selected component and the inequality is an equality. □

A.6 Proof of Theorem 4

Proof of Theorem 4. Construction of the simplex net. For $K = 1$, take $\mathcal{N}_\varepsilon = \{(1)\}$. For $K \geq 2$, write

$$\Delta' = \left\{ u \in \mathbb{R}^{K-1} : u_j \geq 0,\ \sum_{j=1}^{K-1} u_j \leq 1 \right\},$$

and define the lift

$$\iota(u) = \left(u_1, \ldots, u_{K-1}, 1 - \sum_{j=1}^{K-1} u_j \right).$$

This is a bijection from Δ' to Δ^{K-1}. Moreover,

$$\begin{aligned} \|\iota(u) - \iota(v)\|_1 &= \|u - v\|_1 + \left| \sum_{j=1}^{K-1} (u_j - v_j) \right| \\ &\leq 2\|u - v\|_1. \end{aligned}$$

Choose and fix, independently of S_e, a maximal $(\varepsilon/2)$-separated set $\mathcal{U} \subset \Delta'$ in ℓ_1. Maximality implies that $\mathcal{U}$ is an $(\varepsilon/2)$-net of Δ'. Hence $\mathcal{N}_\varepsilon = \iota(\mathcal{U})$ is an ε-net of Δ^{K-1}.

The balls

$$\left\{ B_1\left(t, \frac{\varepsilon}{4}\right) : t \in \mathcal{U} \right\}$$

are pairwise disjoint. Writing B_1^{K-1} for the unit ℓ_1 ball in $\mathbb{R}^{K-1}$, we have $\Delta' \subset B_1^{K-1}$, so the balls are all contained in

$$\left(1 + \frac{\varepsilon}{4}\right) B_1^{K-1}.$$

A volume comparison therefore gives

$$\begin{aligned} |\mathcal{U}| &\leq \left(\frac{1 + \varepsilon/4}{\varepsilon/4} \right)^{K-1} \\ &= \left(1 + \frac{4}{\varepsilon} \right)^{K-1} \leq \left(\frac{5}{\varepsilon} \right)^{K-1}, \end{aligned}$$

where the last inequality uses $\varepsilon \leq 1$. Since ι is injective, $|\mathcal{N}_\varepsilon| = |\mathcal{U}|$, proving the required cardinality bound.

Uniform PAC-Bayes event. For each $\tilde{r} \in \mathcal{N}_\varepsilon$, define

$$P_{\tilde{r}}^{\mathrm{mix}} = \sum_{k=1}^{K} \tilde{r}_k P_k.$$

The net and the prototypes are fixed independently of S_e, so $P_{\tilde{r}}^{\mathrm{mix}}$ is a valid PAC-Bayes prior.

Apply equation 43 at confidence $\delta/|\mathcal{N}_\varepsilon|$ and take a union bound over $\tilde{r} \in \mathcal{N}_\varepsilon$. With probability at least $1 - \delta$, simultaneously for every posterior Q and every $\tilde{r} \in \mathcal{N}_\varepsilon$,

$$\mathrm{kl}\Big(\widehat{R}_e(Q) \,\Big\|\, R_e(Q) \Big) \leq \frac{D_{\mathrm{KL}}\big(Q \,\|\, P_{\tilde{r}}^{\mathrm{mix}}\big) + \ln|\mathcal{N}_\varepsilon| + \ln\frac{2\sqrt{n_e}}{\delta}}{n_e}. \tag{54}$$

Transfer to the realized responsibility. Fix any admissible Q_e and $r_e \in \Delta^{K-1}$. Choose $\tilde{r} \in \mathcal{N}_\varepsilon$ such that

$$\|r_e - \tilde{r}\|_1 \leq \varepsilon.$$

By Lemma 3,

$$D_{\mathrm{KL}}\big(Q_e \,\|\, P_{\tilde{r}}^{\mathrm{mix}}\big) \leq \sum_{k=1}^{K} \tilde{r}_k D_{\mathrm{KL}}(Q_e \,\|\, P_k)\,.$$

Consequently,

$$D_{\mathrm{KL}}\big(Q_e \,\big\|\, P_{\tilde{r}}^{\mathrm{mix}}\big) \le \sum_{k=1}^{K} r_{ek} D_{\mathrm{KL}}(Q_e \,\|\, P_k) + \sum_{k=1}^{K} |\tilde{r}_k - r_{ek}| D_{\mathrm{KL}}(Q_e \,\|\, P_k)$$
$$\le \sum_{k=1}^{K} r_{ek} D_{\mathrm{KL}}(Q_e \,\|\, P_k) + \|r_e - \tilde{r}\|_1 \max_k D_{\mathrm{KL}}(Q_e \,\|\, P_k)$$
$$\le \sum_{k=1}^{K} r_{ek} D_{\mathrm{KL}}(Q_e \,\|\, P_k) + \varepsilon B. \tag{55}$$

The last inequality follows from Assumption 6. An intended choice such as $Q_e = q_e^{\mathrm{agg}}$ or $Q_e = P_e^{\mathrm{agg}}$ must separately satisfy that assumption; a cap on the training penalty does not establish it.

Cardinality and Pinsker relaxation. Substitute equation 55 and $\ln|\mathcal{N}_\varepsilon| \le (K-1)\ln(5/\varepsilon)$ into equation 54. Its numerator is bounded by $\mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) + C_\varepsilon + \ln(2\sqrt{n_e}/\delta)$, with C_ε defined in equation 47. Pinsker's inequality $\mathrm{kl}(a\|b) \ge 2(a-b)^2$ and the nonnegative square root give equation 48. □

A.7 Proof of Theorem 5

Use the environment, fixed posterior-producing rule, and risk R_{post} of Appendix A.4.3. For a reference tuple Θ, define

$$g\big(\Theta, (D,S)\big) = R_D(Q_S) \in [0,1],$$

bounded by Assumption 2, its environment risk

$$G(\Theta) = \mathbb{E}_{(D,S)\sim\mathcal{T}}\big[g\big(\Theta, (D,S)\big)\big],$$

and its empirical counterpart

$$\widehat{G}(\Theta) = \frac{1}{N}\sum_{e=1}^{N} g\big(\Theta, (D_e, S_e)\big).$$

The fixed posterior-producing rule and loss do not read Θ. Thus $g(\Theta,\cdot)$ is constant in Θ, R_{post} is independent of ρ, and

$$\mathbb{E}_\rho[G] = R_{\mathrm{post}}, \qquad \mathbb{E}_\rho[\widehat{G}] = \frac{1}{N}\sum_{e=1}^{N} R_e(Q_e).$$

The argument below uses only measurability and boundedness of g. Assumption 6 is understood to hold ρ-almost surely.

Proof of Theorem 5. Environment level. The draws (D_e, S_e) are i.i.d. from $\mathcal{T}$, $g \in [0,1]$, and the hyper-prior π is fixed. Apply equation 43 with the N draws as the sample at confidence $\delta/2$. With probability at least $1-\delta/2$, simultaneously for all hyper-posteriors ρ,

$$\mathrm{kl}\Big(\mathbb{E}_\rho[\widehat{G}] \,\Big\|\, \mathbb{E}_\rho[G]\Big) \le \frac{D_{\mathrm{KL}}(\rho\|\pi) + \ln\frac{4\sqrt{N}}{\delta}}{N}.$$

By Pinsker's inequality,

$$R_{\mathrm{post}} \le \frac{1}{N}\sum_{e=1}^{N} R_e(Q_e) + \sqrt{\frac{D_{\mathrm{KL}}(\rho\|\pi) + \ln\frac{4\sqrt{N}}{\delta}}{2N}}. \tag{56}$$

Entity level: joint change of measure. Because ρ depends on the training entities, the fixed-Θ statement of Theorem 4 cannot be integrated over ρ. Instead, fix an entity e and a net point $\tilde{r} \in \mathcal{N}_\varepsilon$, and consider on pairs (Θ, z) the joint prior and posterior

$$\mathsf{P}_{\tilde{r}}(\mathrm{d}\Theta, \mathrm{d}z) = \pi(\mathrm{d}\Theta)\, P_{\tilde{r}}^{\mathrm{mix}}(\mathrm{d}z \mid \Theta),$$
$$\mathsf{Q}_e^{\rho}(\mathrm{d}\Theta, \mathrm{d}z) = \rho(\mathrm{d}\Theta)\, Q_e(\mathrm{d}z),$$

following Amit & Meir (2018). The prior is fixed independently of S_e; only the posterior depends on the data.

Apply equation 43 over S_e with this prior at confidence $\delta/(2N|\mathcal{N}_\varepsilon|)$. Since the loss does not read Θ, the Q_e^ρ-averaged empirical and true losses coincide with $\widehat{R}_e(Q_e)$ and $R_e(Q_e)$. The chain rule gives

$$D_{\mathrm{KL}}(\mathsf{Q}_e^\rho \,\|\, \mathsf{P}_{\tilde{r}}) = D_{\mathrm{KL}}(\rho\|\pi) + \mathbb{E}_\rho\big[D_{\mathrm{KL}}\big(Q_e \,\big\|\, P_{\tilde{r}}^{\mathrm{mix}}(\cdot \mid \Theta)\big)\big]\,.$$

Take a union bound over $\tilde{r} \in \mathcal{N}_\varepsilon$ and $e \in \{1,\dots,N\}$. On the resulting common event, choose for each entity e a net point $\tilde{r} \in \mathcal{N}_\varepsilon$ satisfying $\|r_e - \tilde{r}\|_1 \le \varepsilon$. Apply equation 55 under $\mathbb{E}_\rho$, valid ρ-almost surely by Assumption 6:

$$\mathbb{E}_\rho\big[D_{\mathrm{KL}}\big(Q_e \,\big\|\, P_{\tilde{r}}^{\mathrm{mix}}(\cdot \mid \Theta)\big)\big] \le \mathbb{E}_\rho\left[\sum_{k=1}^K r_{ek}\, D_{\mathrm{KL}}(Q_e \,\|\, P_k)\right] + \varepsilon B.$$

For the confidence term, note

$$\ln\left(\frac{2\sqrt{n_e}\cdot 2N|\mathcal{N}_\varepsilon|}{\delta}\right) = \ln\frac{4N\sqrt{n_e}}{\delta} + \ln|\mathcal{N}_\varepsilon|, \qquad \ln|\mathcal{N}_\varepsilon| \le (K-1)\ln\frac{5}{\varepsilon}.$$

With probability at least $1-\delta/2$, simultaneously for all ρ, all admissible Q_e, and all $r_e \in \Delta^{K-1}$, Pinsker's inequality then gives, for every entity e,

$$R_e(Q_e) \le \widehat{R}_e(Q_e) + \sqrt{\frac{D_{\mathrm{KL}}(\rho\|\pi) + \mathbb{E}_\rho[\sum_k r_{ek}\, D_{\mathrm{KL}}(Q_e \,\|\, P_k)] + C_\varepsilon + \ln\frac{4N\sqrt{n_e}}{\delta}}{2\,n_e}}, \tag{57}$$

with $C_\varepsilon = \varepsilon B + (K-1)\ln(5/\varepsilon)$.

Combination. The environment level consumes $\delta/2$; the entity level consumes $N\cdot\delta/(2N) = \delta/2$. Both events therefore hold jointly with probability at least $1-\delta$. Substituting equation 57 into equation 56 yields equation 50. □

The treatment of shared learned priors follows the meta-learning PAC-Bayes arguments of Pentina & Lampert (2014) and Amit & Meir (2018); the responsibility-weighted complexity handled through Lemma 3 and the simplex-net construction in the proof of Theorem 4 are the reference-prior ingredients used here. The argument certifies R_{post} for the independently fixed prediction map and posterior-producing rule under the stated within- and across-entity sampling assumptions; it does not identify these reference divergences with the soft component-weighted training penalty or the prototype-dependent deployed risk.

A.8 Proof of Corollary 3

Proof. Let

$$\mathcal{C}_e^{\mathrm{meta}}(\rho) := \mathbb{E}_\rho \mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) = \mathbb{E}_\rho\left[\sum_{k=1}^K r_{ek} D_{\mathrm{KL}}(Q_e\|P_k)\right],$$

$$A = C_{\mathrm{hyp}}(\rho) + C_\varepsilon + \ln\frac{4N\sqrt{n_{\max}}}{\delta}.$$

Because $n_e \ge n_{\min}$ and $\ln(4N\sqrt{n_e}/\delta) \le \ln(4N\sqrt{n_{\max}}/\delta)$, the entity-level contribution in equation 50 satisfies

$$\begin{aligned}\frac{1}{N}\sum_{e=1}^N \sqrt{\frac{C_{\mathrm{hyp}}(\rho) + \mathcal{C}_e^{\mathrm{meta}}(\rho) + C_\varepsilon + \ln\frac{4N\sqrt{n_e}}{\delta}}{2n_e}} &\le \frac{1}{N}\sum_{e=1}^N \sqrt{\frac{A + \mathcal{C}_e^{\mathrm{meta}}(\rho)}{2n_{\min}}} \\ &\le \sqrt{\frac{A + \frac{1}{N}\sum_{e=1}^N \mathcal{C}_e^{\mathrm{meta}}(\rho)}{2n_{\min}}},\end{aligned}$$

where the second inequality is Jensen's inequality for the concave square root. Linearity of expectation and interchange of the finite sums give

$$\frac{1}{N}\sum_{e=1}^{N}\mathcal{C}_e^{\mathrm{meta}}(\rho) = \frac{1}{N}\sum_{e=1}^{N}\sum_{k=1}^{K} r_{ek}\,\mathbb{E}_\rho[D_{\mathrm{KL}}(Q_e\|P_k)] = \sum_{k:N_k^{\mathrm{eff}}>0}\frac{N_k^{\mathrm{eff}}}{N}\,\overline{D}_k(\rho).$$

Substituting this identity into the Jensen bound and then into equation 50 proves equation 52.

For one-hot assignments, N_k^{eff} is the number of entities assigned to prototype k, so $\overline{D}_k(\rho)$ is their arithmetic mean local reference divergence. The final monotonicity statement follows directly from monotonicity of the square root, with all other displayed quantities held fixed. □

B ARCHITECTURE AND TRAINING

B.1 BACKBONE PIPELINE DETAILS

This appendix specifies every instantiation of the backbone F_ϕ: the headline flow-matching backbone, the diffusion backbone, and the four deterministic forecaster families of the backbone matrix in Section 4.3. The latent partial-pooling construction is shared across these families and is described in Section 3; each backbone uses an *architecture-specific conditioning interface*, specified below.

In the architecture diagrams, independent draws $z^q_{e,k} \sim q_{e,k}$ are aggregated as $z_e = \sum_k r_{ek} z^q_{e,k} \sim q^{\text{agg}}_e$ during training. At deployment, the same map aggregates independent $z^P_{e,k} \sim P_k$ into P^{agg}_e, except for the explicitly identified prior-mean retrofits. The alignment label $\mathcal{L}_{\text{pool},e}$ denotes uncapped componentwise KL; its context average is the optimized pooling penalty on both datasets, where $\kappa_z = \infty$ and $\mathcal{L}^{\text{train}}_{\text{pool}} = \mathcal{L}_{\text{pool}}$. In contrast, $\mathcal{C}^{\text{agg}}_e = D_{\text{KL}}(q^{\text{agg}}_e \| P^{\text{agg}}_e)$ compares the aggregate laws, and $\mathcal{C}^{\text{comp}}_e = \sum_k D_{\text{KL}}(q_{e,k} \| P_k)$ is the unweighted component sum. The latter upper-bounds aggregate KL by data processing; the responsibility-weighted pooling penalty need not.

Figure 2 gives an architectural overview of the flow-matching instantiation, complementing the statistical mechanism in Figure 1.

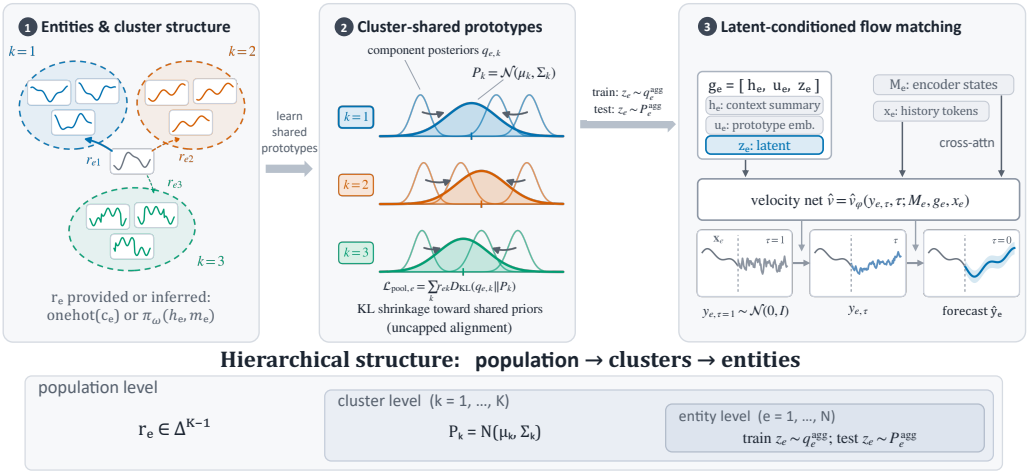

Figure 2: Architectural overview of CLASP with a flow-matching backbone. Entity histories and cluster assignments (**1**) yield prototype-aligned latent representations (**2**), which condition forecast generation (**3**). The forecaster receives $g_e = [h_e, u_e, z_e]$, encoder states M_e, and raw-history tokens. Independent component-posterior draws form q^{agg}_e during training; independent prototype-prior draws form P^{agg}_e at deployment. The displayed $\mathcal{L}_{\text{pool},e} = \sum_k r_{ek} D_{\text{KL}}(q_{e,k} \| P_k)$ is the uncapped componentwise regularizer, not aggregate latent KL.

Common to all families: the input is the enriched history sequence $[x^{\text{norm}}_e,\ \text{mask},\ \Delta t]$ with $2d{+}1 = 13$ channels, where x^{norm}_e denotes the normalized history under the selected configuration; predictions are restored to original units for evaluation. The conditioning vector is

$$g_e = [h_e,\ u_e,\ z_e] \in \mathbb{R}^{192} \qquad (d_h{=}128,\ d_u{=}32,\ d_z{=}32),$$

where d_u is the prototype-embedding dimension. In the pipeline diagrams, blocks labeled "Prototype bank" contain both the prototype priors P_k and the learned prototype embeddings u_k. The representation $g_e = [h_e, u_e, z_e]$ is common across backbones; the mechanism that injects it into the forecaster is architecture-specific. The eICU configuration clips the latent coordinates to $[-5, 5]$ inside the deterministic conditioning map of F_ϕ. This operation is applied after drawing z_e and does not redefine the Gaussian aggregate posterior q^{agg}_e or aggregate deployment prior P^{agg}_e.

Plain-backbone convention. A *plain* backbone retains its native history processing, forecasting objective, and generation procedure, with the CLASP context encoder, router, prototype bank, la-

tent branch, and conditioning interface absent. It also omits the CLASP forecast readout unless an explicit readout-only control enables it. Native encoders and latent variables, such as D3VAE's own latent variables, remain part of the host. The prototype count K is inapplicable to a plain backbone; the $K = 1$ CLASP control instead retains a pooled latent and its regularizer. The plain-vs-CLASP contrasts evaluate the combined conditioning and validation-fitted readout pipeline. Within each plain/CLASP pair across all six families, the optimizer, learning-rate schedule, training-step budget, checkpoint-selection rule, and hyperparameter-tuning budget are matched. In the implementation, the component-posterior head is evaluated once per prototype, conditioned on the corresponding prototype embedding, and the resulting per-prototype quantities are combined with the responsibilities r_e; the prototype embedding u_e also enters the conditioning vector.

Latent sampling across backbones. Table 8 collects the training and test latent sources, forecast draw counts, numerical sampling budgets, and readout conventions. All listed CLASP variants train with independent component-posterior draws aggregated into q_e^{agg}. The two prior-mean retrofits replace only the CLASP latent at inference; we use $z_e = \mu_e^P$ for Timer-XL and D3VAE to preserve their native single-forecast protocols without adding CLASP-specific Monte Carlo variability that cannot be averaged across draws at $S = 1$. D3VAE retains its host's decoder-mean prediction path and score correction; fixing the CLASP latent does not assert that internal VAE latents are deterministic. Timer-XL is deterministic at inference. The other external retrofits (NsDiff, TSFlow, and TSDiff) average $S = 100$ or 32 forecasts, reducing Monte Carlo variability across draws. The four direct point forecasters use a different reported protocol: they are deterministic conditional on z_e but retain one aggregate-prior draw per window at $S = 1$.

Table 8: Latent-sampling protocols for the full CLASP backbone and retrofit comparisons. The columns specify the CLASP latent z_e using the aggregate posterior q_e^{agg}, aggregate deployment law P_e^{agg}, and its mean μ_e^P. "Blend" is the fixed validation-fitted two-anchor readout of equation 12, followed by the arithmetic mean of S forecasts (the single forecast when $S = 1$). S counts requested draws; NsDiff excludes invalid draws (Appendix C.1). The first six rows specify the provided-zone WEATHER-5K backbone comparison ($K = 5$); the flow-matching and DiT rows also specify the inferred-$K = 4$ WEATHER-5K and eICU configurations, respectively. The last five rows describe the provided-zone WEATHER-5K retrofits ($K = 5$). Latent-source ablations and fixed-predictor concentration diagnostics have their own explicitly stated protocols.

Backbone	Training latent	Test latent	S	Sampling steps / calls	Readout
Flow matching	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	100	50 Euler steps	Blend
DiT diffusion	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	100	50 DDIM steps	Blend
Transformer	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	1	1 forward pass	Blend
GRU	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	1	1 forward pass	Blend
SSM	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	1	1 forward pass	Blend
TCN	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	1	1 forward pass	Blend
NsDiff	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	100	20 reverse steps	Blend
TSFlow	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	100	32 Euler steps	Blend
Timer-XL	$z_e \sim q_e^{\mathrm{agg}}$	$z_e = \mu_e^P$	1	1 call; 2 for $H = 168$	Blend
D3VAE	$z_e \sim q_e^{\mathrm{agg}}$	$z_e = \mu_e^P$	1	Decoder mean and 1 score step	Blend
TSDiff	$z_e \sim q_e^{\mathrm{agg}}$	$z_e \sim P_e^{\mathrm{agg}}$	32	25 DDIM steps	Blend

B.1.1 Headline Flow-Matching Backbone

The network is a DiT-style Transformer (Peebles & Xie, 2023) with 6 blocks, 8 heads, hidden width 128, and MLP ratio 4; each block applies self-attention over the token sequence, cross-attention to the encoder states M_e, and a feedforward update. CLASP enters through three conditioning pathways: adaLN-Zero modulation by g_e together with the time embedding, cross-attention to M_e, and raw-history tokens prepended to the token sequence. Time is kept on the integer grid $t \in \{0, \dots, T_{\mathrm{gen}} - 1\}$ with $T_{\mathrm{gen}} = 200$ and mapped to $\tau_t = t/(T_{\mathrm{gen}} - 1) \in [0, 1]$. Training samples $t \sim \mathrm{Unif}\{0, \dots, T_{\mathrm{gen}} - 1\}$ and corrupts the clean target y_e along the straight path

$$y_{e,\tau} = (1 - \tau)\, y_e + \tau\, \epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \tau = \tau_t. \tag{58}$$

The conditional velocity network $\hat{v}_\phi(y_{e,\tau}, \tau; h_e, u_e, z_e, M_e, x_e)$ regresses the constant path velocity $v = \epsilon - y_e$. The forecasting loss is

$$\mathcal{L}_{\text{forecast}} = \mathbb{E}_{e,\tau,\epsilon,z_e}\left[\ell_{\text{H}}(\hat{v}_\phi,\ \epsilon - y_e) + \frac{\alpha}{Hd}\left\|\widehat{y}_e^{(0)} - y_e\right\|_1\right], \qquad \widehat{y}_e^{(0)} = y_{e,\tau} - \tau\,\hat{v}_\phi, \tag{59}$$

where ℓ_{H} is the mean elementwise Huber loss, $z_e \sim q_e^{\text{agg}}$, and $\alpha = 0.05$. The norm sums absolute entries, so the auxiliary term is α times the mean absolute reconstruction error; no SNR-dependent weighting is used. The clean-target estimate $\widehat{y}_e^{(0)}$ also serves as the self-conditioning input (Chen et al., 2023). Sampling integrates the probability-flow ODE $\mathrm{d}y/\mathrm{d}\tau = \hat{v}_\phi$ with 50 forward-Euler steps on a uniform τ-grid from 1 to 0, $y \leftarrow y - \Delta\tau\,\hat{v}_\phi$, with $\Delta\tau = 1/50$; the network receives the rounded integer index $\mathrm{round}((T_{\text{gen}} - 1)\tau)$. The integration is deterministic given the initial draw $y_{e,\tau=1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At prediction time the latent is drawn from the aggregate deployment prior, $z_e \sim P_e^{\text{agg}}$, and $S = 100$ trajectories are sampled with the latent and the initial noise redrawn per trajectory; the CLASP readout stage of Section 3.5 converts them into point and interval forecasts.

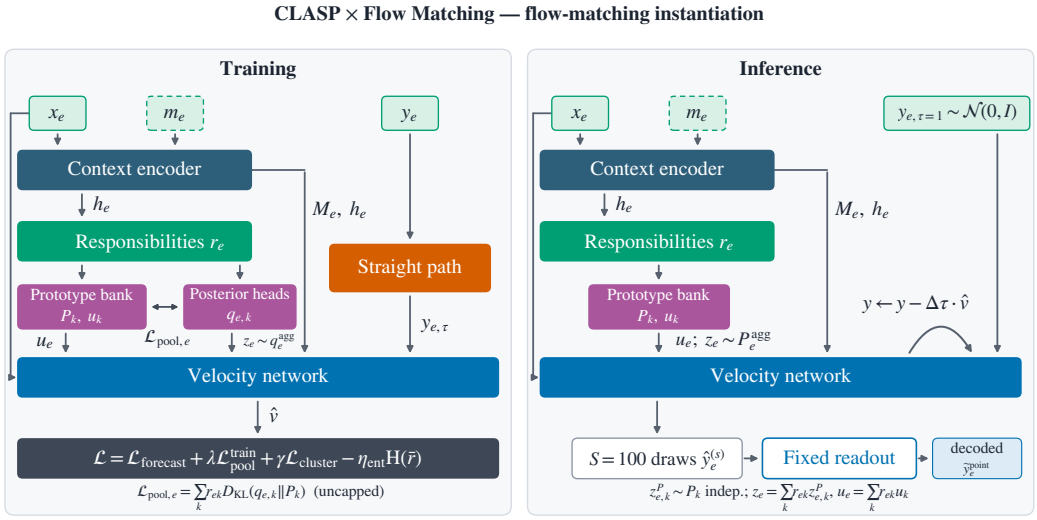

Figure 3: CLASP with flow matching: training on the left and inference on the right. The forecaster learns a conditional velocity field and generates trajectories by ODE integration.

B.1.2 DiT Diffusion Backbone

The diffusion instantiation uses the identical network and conditioning pathways as the flow-matching backbone and exchanges only the generative procedure. The clean target y_e is corrupted with the standard forward process (Ho et al., 2020). With zero-based noise indices $t \in \{0, \ldots, T_{\text{gen}} - 1\}$,

$$y_{e,t} \mid y_e \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\, y_e, (1 - \bar{\alpha}_t)\mathbf{I}\right), \qquad \bar{\alpha}_t = \prod_{j=0}^{t}(1 - \beta_j),$$

with a cosine noise schedule; equivalently, $y_{e,t} = \sqrt{\bar{\alpha}_t}\, y_e + \sqrt{1 - \bar{\alpha}_t}\,\epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus $y_{e,0}$ is the first corrupted state, whereas y_e denotes the clean target. The model uses the v-parameterization (Salimans & Ho, 2022) with target $v = \sqrt{\bar{\alpha}_t}\,\epsilon - \sqrt{1 - \bar{\alpha}_t}\, y_e$ and the diffusion analogue of the velocity term in equation 59,

$$\mathcal{L}_{\text{forecast}} = \mathbb{E}_{e,t,\epsilon,z_e}\left[w(t)\,\ell_{\text{H}}(\hat{v}_\phi, v) + \frac{\alpha_{\text{diff}}}{Hd}\left\|\widehat{y}_e^{(0)} - y_e\right\|_{\text{F}}^2\right], \tag{60}$$

where t is uniform over the noise indices, $z_e \sim q_e^{\text{agg}}$, and $\widehat{y}_e^{(0)} = \sqrt{\bar{\alpha}_t}\, y_{e,t} - \sqrt{1 - \bar{\alpha}_t}\,\hat{v}_\phi$ is the clean-target estimate. Here $w(t)$ is the Min-SNR weight (Hang et al., 2023) and $\|\cdot\|_{\text{F}}$ is the Frobenius norm. The reported eICU checkpoint uses $\alpha_{\text{diff}} = 0.1$, giving mean-squared reconstruction loss; this coefficient and loss differ from the flow backbone's $\alpha = 0.05$ mean-absolute term. Self-conditioning is used as in the flow backbone. Sampling starts from $y_{e,T_{\text{gen}}-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applies DDIM (Song et al., 2021) reverse steps; the latent source and the $S = 100$ protocol are as above.

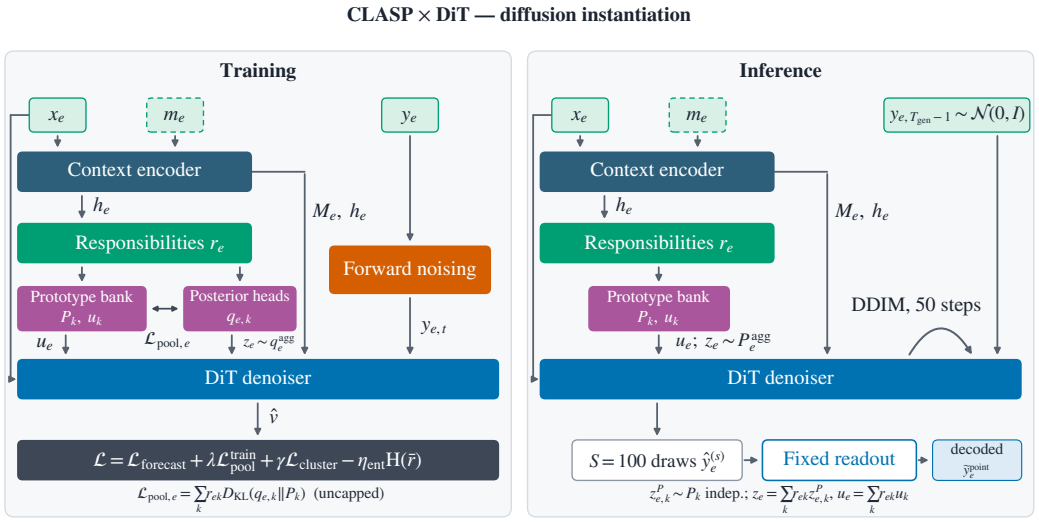

Figure 4: CLASP with DiT diffusion: training on the left and inference on the right. The conditioning pathways match Figure 3, with noising and DDIM sampling replacing the flow-matching procedure.

Deterministic forecaster families: The four families below are direct multi-horizon point forecasters: a single forward pass maps the enriched history to the full 168-step horizon, which is sliced to the evaluated H. Training uses pointwise MAE plus the CLASP regularizers of equation 5, with AdamW, learning rate 3×10^{-4}, cosine schedule with 1,000-step warmup, batch 128, at most 30,000 steps, the best-validation checkpoint, and no EMA. They are evaluated as single-pass point forecasts with $S{=}1$ and one latent draw $z_e \sim P_e^{\text{agg}}$ per window. The anchor blend of equation 12 applies to a single forecast as well. The full CLASP rows of Table 26 include this validation-fitted blend; the plain rows use unmodified forecasts. These deterministic configurations produce no intervals.

B.1.3 Transformer Forecaster

A patch-based encoder in the style of PatchTST (Nie et al., 2023) splits the history into patches of length 16 with stride 8, producing 5 tokens. These are embedded to width 128 and processed by 3 pre-norm Transformer encoder layers with 8 heads, feedforward width 512, GELU activations, and dropout 0.1. CLASP enters as one learned *prefix token*, a linear map of g_e to width 128 prepended to the patch tokens and attended by all of them; the prefix is dropped before the head. The forecast head flattens the patch tokens and maps them linearly to the $168 \times d$ horizon.

B.1.4 GRU Forecaster

A sequence-to-sequence GRU serves as the forecaster, separate from the GRU *context encoder* of the CLASP module. A 2-layer encoder of width 128 reads the history; a 2-layer decoder unrolls 168 learned per-step query embeddings in one pass, with a per-step linear head to the d output channels. CLASP enters *input-only*: g_e is concatenated to every decoder input, giving an input width of $128 + 192$ and leaving the encoder state untouched. We ablated three injection variants: fusing g_e into the initial decoder state through a $\tanh$ layer, adding a zero-initialized residual to the initial state, and input-only conditioning. Input-only conditioning performed best and is the reported variant.

B.1.5 SSM Forecaster

We use a stack of 4 diagonal state-space blocks of width 128 in the style of S4D (Gu et al., 2022). Each channel carries a real diagonal continuous-time system $x' = Ax + Bu$, $y = Cx + Du$ with $A = -\exp(\log a) < 0$, ensuring stability by construction. It uses S4D-real initialization $a_n = n + \frac{1}{2}$, state dimension 16, and a learnable per-channel step size discretized by zero-order

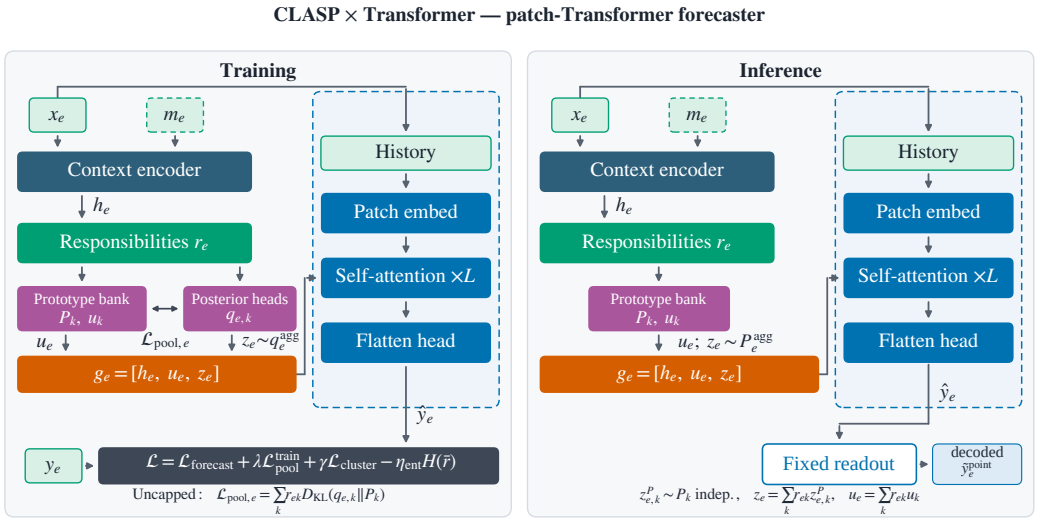

Figure 5: CLASP with the Transformer forecaster, conditioned by a learned prefix token derived from $g_e = [h_e, u_e, z_e]$.

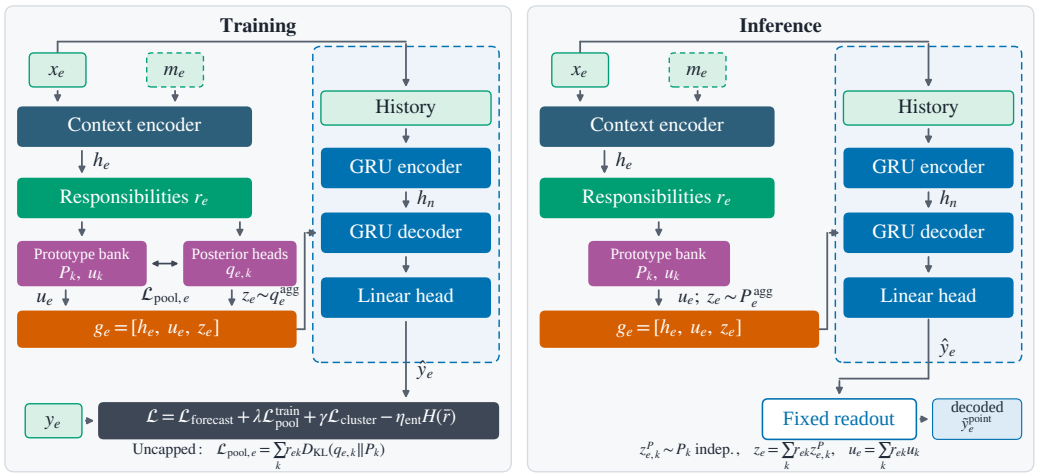

Figure 6: CLASP with the GRU forecaster, with $g_e = [h_e, u_e, z_e]$ concatenated to each decoder input.

hold; the convolution kernel is materialized explicitly and applied as a grouped causal convolution with a D-skip. CLASP enters as zero-initialized FiLM modulation of every block input, which is the identity at initialization. The head applies layer normalization to the final timestep and maps it linearly to the $168 \times d$ horizon.

B.1.6 TCN Forecaster

A causal temporal convolutional network (Bai et al., 2018) uses 5 residual blocks of width 128, kernel size 3, and dilations $1, 2, 4, 8, 16$, using causal left-padding. The receptive field spans $63 \geq 48$ history steps. CLASP enters as channel-wise zero-initialized FiLM applied after each convolution, before the residual addition. The head is as in the SSM forecaster: layer normalization on the final timestep followed by a linear map to the $168 \times d$ horizon.

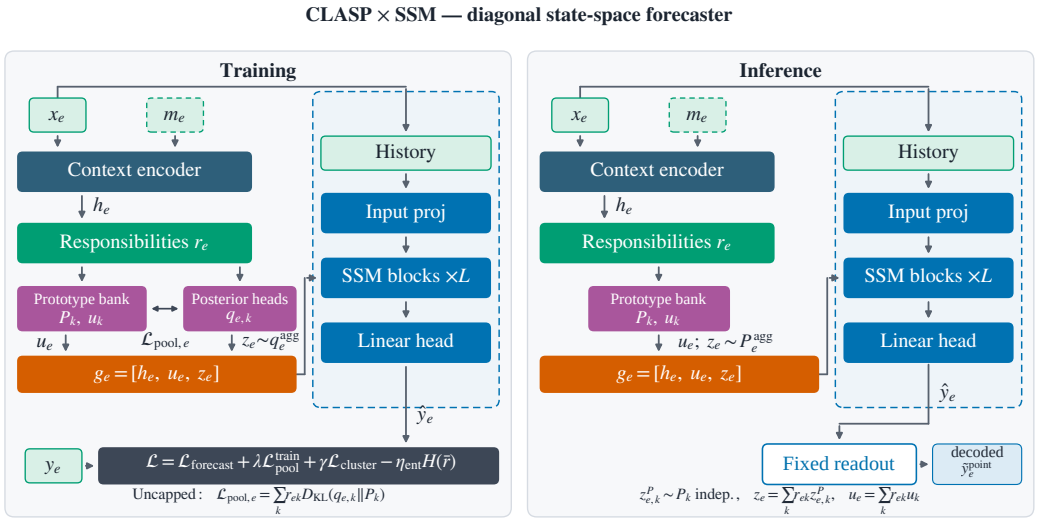

Figure 7: CLASP with the SSM forecaster, using zero-initialized FiLM driven by $g_e = [h_e, u_e, z_e]$ at each block input.

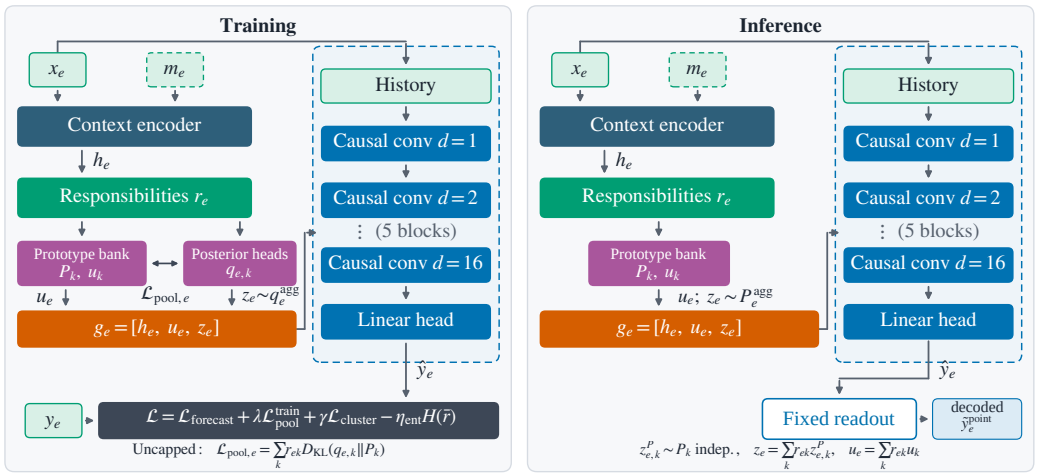

Figure 8: CLASP with the TCN forecaster, using zero-initialized FiLM driven by $g_e = [h_e, u_e, z_e]$ after each convolution.

B.2 Shared Training Procedure

CLASP constructs the conditioning vector $g_e = [h_e, u_e, z_e]$ from the history representation, responsibility-weighted prototype embedding, and aggregate latent. Training draws independent component latents from $q_{e,k}$, whereas the default inference procedure draws them from P_k; in both cases, $z_e = \sum_k r_{ek} z_{e,k}$. These weighted sums follow the aggregate Gaussian laws in equation 6 and equation 7, rather than categorical mixtures. Table 8 identifies the prior-mean inference exceptions.

Pooling objective and optional component-KL cap. The definition in equation 4 distinguishes the uncapped entity-context penalty $\mathcal{L}_{\text{pool},e}$ from its training-context average $\mathcal{L}_{\text{pool}}$. Here N_{tr} is the number of training entity contexts, with window dependence suppressed. The forecasting loss can favor context-dependent departures from the shared prototype priors, while the pooling penalty

regularizes these departures. This objective does not approximate a marginal likelihood and does not require an explicit generative observation likelihood.

The general training formula allows an optional component-KL cap:

$$\begin{aligned}\mathcal{L}_{\text{pool},e}^{\text{train}} &= \sum_{k=1}^{K} r_{ek} \min\{D_{\text{KL}}(q_{e,k} \| P_k), \kappa_z\}\,, \\ \mathcal{L}_{\text{pool}}^{\text{train}} &= \frac{1}{N_{\text{tr}}} \sum_{e=1}^{N_{\text{tr}}} \mathcal{L}_{\text{pool},e}^{\text{train}}.\end{aligned} \tag{61}$$

If finite, the cap applies to each component KL *before* responsibility weighting and context averaging. The corresponding training objective is

$$\mathcal{L} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{pool}}^{\text{train}} + \gamma \mathcal{L}_{\text{cluster}} - \eta_{\text{ent}} \mathsf{H}(\bar{r}). \tag{62}$$

Both WEATHER-5K and eICU use uncapped pooling, $\kappa_z = \infty$, so $\mathcal{L}_{\text{pool}}^{\text{train}} = \mathcal{L}_{\text{pool}}$ and equation 62 reduces to equation 5 in all reported experiments. With a finite cap, the training and uncapped pooling penalties coincide only when no positively weighted component KL is clipped.

For each entity-window in a minibatch $\mathcal{B}$, write

$$\ell_{\text{pool},e} = \mathcal{L}_{\text{pool},e}, \qquad \ell_{\text{pool},e}^{\text{train}} = \mathcal{L}_{\text{pool},e}^{\text{train}}. \tag{63}$$

Minibatch averages estimate the training-context objectives. The uncapped pooling penalty is retained for diagnostics; latent-coordinate clipping inside the forecaster does not change its definition or cap its value. The coefficient λ controls the pooling contribution relative to the forecasting and router terms. Because training forecasts use component-posterior draws, the pooling objective supplies both the posterior–prototype alignment signal and the learning signal for the prototype-prior parameters. Setting $\lambda = 0$ therefore removes both signals, rather than isolating alignment alone.

Router regularization. For inferred responsibilities, $\mathcal{L}_{\text{cluster}}$ is the minibatch mean of $D_{\text{KL}}(r_e \| p_c)$ with $p_c = (1/K, \ldots, 1/K)$, regularizing each window's weights toward uniformity. This categorical KL regularizes the weights; it does not sample a prototype index for the latent construction. Its coefficient is $\gamma = 1$ in the reported inferred configurations. The entropy term uses

$$\bar{r} = |\mathcal{B}|^{-1} \sum_{e \in \mathcal{B}} r_e, \qquad \mathsf{H}(\bar{r}) = -\sum_k \bar{r}_k \ln \bar{r}_k.$$

Thus $\gamma \mathcal{L}_{\text{cluster}}$ regularizes each window's responsibilities, whereas $-\eta_{\text{ent}} \mathsf{H}(\bar{r})$ encourages balanced usage across the minibatch. The batch-entropy weights are $\eta_{\text{ent}} = 0$ on WEATHER-5K and $\eta_{\text{ent}} = 0.01$ on eICU. Provided labels fix $r_e = \text{onehot}(c_e)$ and set $\mathcal{L}_{\text{cluster}} = 0$ by convention, with $\gamma = 0$; the one-hot-to-uniform KL is not used as a loss, and the entropy bonus has no dependence on trainable parameters.

Prototype moments and the interpretation of partial pooling. Proposition 2 in Appendix A.3 characterizes the optimizer of the *uncapped* prototype objective conditional on fixed component posteriors and responsibilities. For $N_k^{\text{eff}} = \sum_{e=1}^{N_{\text{tr}}} r_{ek} > 0$, the full-covariance optimum satisfies

$$\begin{aligned}\mu_k^\star &= \frac{1}{N_k^{\text{eff}}} \sum_{e=1}^{N_{\text{tr}}} r_{ek} \mu_{e,k}, \\ \Sigma_k^\star &= \frac{1}{N_k^{\text{eff}}} \sum_{e=1}^{N_{\text{tr}}} r_{ek} \left[\Sigma_{e,k} + (\mu_{e,k} - \mu_k^\star)(\mu_{e,k} - \mu_k^\star)^\top\right].\end{aligned} \tag{64}$$

The mean $\mu_k^\star$ expresses responsibility-weighted sharing of entity-context means. The covariance decomposes prototype dispersion into *within-posterior uncertainty*, represented by $\Sigma_{e,k}$, and *between-entity/context heterogeneity*, represented by the dispersion of $\mu_{e,k}$ around the shared mean. This gives the statistical interpretation of partial pooling: the reference distribution shares information across related contexts while accommodating both uncertainty within their component posteriors and variation between their means.

For the diagonal prototype-covariance family used in our implementation, the corresponding conditional optimum retains only the diagonal of the covariance expression in equation 64. A prototype with $N_k^{\text{eff}} = 0$ receives no constraint from this conditional objective. These statements describe a fixed-posterior, fixed-responsibility optimum, not the solution necessarily reached by neural training. Joint optimization changes the component posteriors and, when inferred, the responsibilities, so learned prototype priors need not attain these moments exactly. Appendix E.2.4 reports empirical moment diagnostics assessing this conditional moment-matching structure.

Relation to aggregate latent KL. The pooling–KL correspondence in Section 3.4 concerns the *uncapped entity-context penalty* $\mathcal{L}_{\text{pool},e}$. For the aggregate laws defined in Section 3.3, write $\mathcal{C}_e^{\text{agg}} = D_{\text{KL}}(q_e^{\text{agg}} \| P_e^{\text{agg}})$. One-hot assignments give

$$\mathcal{C}_e^{\text{agg}} = \mathcal{L}_{\text{pool},e}, \qquad \mathcal{L}_{\text{pool},e}^{\text{train}} = \min\{\mathcal{C}_e^{\text{agg}}, \kappa_z\}\,. \tag{65}$$

Thus, equality with the optimized entity-context penalty holds only when the cap is inactive, meaning that the selected component KL does not exceed κ_z; this condition holds for the uncapped configurations reported here. For soft responsibilities, neither the uncapped responsibility-weighted penalty nor its capped counterpart is a general upper bound on the aggregate KL; the data-processing bound in equation 9 instead uses the unweighted sum of uncapped component KL divergences.

Pooling-weight warmup. For the headline configurations with target weight $\lambda = 1$, the coefficient used in equation 5 increases linearly from zero to one during the first $5{,}000$ optimizer steps:

$$\lambda(t) = \min\left\{\frac{t}{5{,}000}, 1\right\}, \qquad t \geq 0. \tag{66}$$

It remains one from step $5{,}000$ onward. Here t indexes training updates, separately from the generative time index used inside the backbone. Optimizer learning-rate schedules are specified separately.

Backbone-specific operations. BACKBONELOSS and BACKBONESAMPLE encapsulate the forecasting objective and generation procedure. The algorithms use a common notation for these operations; the architecture-specific conditioning interface determines how g_e, encoder states M_e, and raw history enter each backbone (Appendix B.1). Flow matching uses velocity regression with the auxiliary clean-target term in equation 59; diffusion uses forward noising, Min-SNR-weighted v-prediction, and DDIM sampling. The deterministic Transformer, GRU, SSM, and TCN families use pointwise MAE and a single forward pass. Each probabilistic sampling call draws fresh initial noise in addition to the independently resampled latent. A deterministic backbone is noiseless conditional on its sampled latent, but the latent draw remains stochastic. Appendix B.1 specifies the architectures, forecasting losses, and numerical samplers.

Algorithm 1 summarizes one training step.

Algorithm 1 CLASP training step. A plain backbone bypasses this module and optimizes its native forecasting objective (Appendix B.1).

Require: Minibatch $\mathcal{B}$ of (x_e, y_e, m_e, c_e), with metadata and cluster labels optional; current pooling weight $\lambda(t)$, other configured regularization weights, and $\kappa_z = \infty$

for $e \in \mathcal{B}$ **do**
  Encode $h_e \leftarrow f_\theta(x_e, m_e)$ and retain sequence states M_e
  if cluster labels are provided **then**
    $r_e \leftarrow \mathrm{onehot}(c_e)$; $\ell_{\mathrm{cluster},e} \leftarrow 0$
  else
    $r_e \leftarrow \pi_\omega(h_e, m_e)$; $\ell_{\mathrm{cluster},e} \leftarrow D_{\mathrm{KL}}(r_e \| p_c)$
  end if
  $u_e \leftarrow \sum_k r_{ek} u_k$
  for $k = 1, \ldots, K$ **do**
    $(\mu_{e,k}, \Sigma_{e,k}) \leftarrow g_\psi(h_e, u_k, m_e)$
    $q_{e,k} \leftarrow \mathcal{N}(\mu_{e,k}, \Sigma_{e,k})$; draw $z_{e,k} \sim q_{e,k}$ independently over k
  end for
  $z_e \leftarrow \sum_k r_{ek} z_{e,k}$; $g_e \leftarrow [h_e, u_e, z_e]$
  $\ell_{\mathrm{pool},e} \leftarrow \sum_k r_{ek} D_{\mathrm{KL}}(q_{e,k} \| P_k)$ ▷ uncapped diagnostic
  $\ell^{\mathrm{train}}_{\mathrm{pool},e} \leftarrow \sum_k r_{ek} \min\{D_{\mathrm{KL}}(q_{e,k} \| P_k), \kappa_z\}$ ▷ optimized penalty
  $\ell_{\mathrm{forecast},e} \leftarrow \textsc{BackboneLoss}(y_e, x_e; M_e, g_e)$ ▷ backbone-specific
end for
Average the per-window forecast, training-pooling, and categorical KL terms over $\mathcal{B}$
$\bar{r} \leftarrow |\mathcal{B}|^{-1} \sum_{e \in \mathcal{B}} r_e$
Update trainable parameters using equation 62 with the current pooling weight $\lambda(t)$
Retain $\mathcal{L}_{\mathrm{pool}} \leftarrow |\mathcal{B}|^{-1} \sum_{e \in \mathcal{B}} \ell_{\mathrm{pool},e}$ for diagnostics

B.3 CLASP Hyperparameter Details

Table 9 summarizes the main hyperparameters of the headline CLASP configurations: inferred-cluster diffusion on eICU and provided-cluster flow matching on WEATHER-5K. The table distinguishes their forecasting objectives and samplers as well as their clustering and regularization settings. For the inferred-cluster WEATHER-5K experiments, the router-loss weight is $\gamma = 1$ with $K = 4$ and $\eta_{\mathrm{ent}} = 0$.

Pooling weight and uncapped regularization. The target weight $\lambda = 1$ was selected using validation entities under the reported protocol; it is not privileged by the aggregate-KL theory. The held-out WEATHER-5K sweep in Table 25 provides supplementary sensitivity evidence: $\lambda = 1$ minimizes reported MSE at all three horizons and MAE at two of three. These results support the selected setting but do not establish a universally optimal weight.

Both eICU and WEATHER-5K use the full, uncapped component KL penalty, with $\kappa_z = \infty$ and $\mathcal{L}^{\mathrm{train}}_{\mathrm{pool}} = \mathcal{L}_{\mathrm{pool}}$. The eICU $[-5, 5]$ clip applies only to sampled latent coordinates inside the forecasting conditioning map; it does not cap the KL loss or redefine the Gaussian latent laws. The moment diagnostics in Appendix E.2.4 therefore concern uncapped training on both datasets, subject to the fixed-posterior and fixed-responsibility qualifications of Proposition 2.

Table 9: CLASP hyperparameters for the headline inferred-cluster eICU diffusion and provided-cluster WEATHER-5K flow-matching configurations. The inferred-cluster WEATHER-5K setting uses $K = 4$, $\gamma = 1$, and $\eta_{\text{ent}} = 0$. Latent sources, forecast counts, and readout are given in Table 8.

Hyperparameter	eICU	WEATHER-5K
Assignment mode	inferred	provided
Number of clusters K	4	5
History length	12	48
Input dimension	6	6
Metadata features	$\text{diag}_{\text{cardio}}$, $\text{diag}_{\text{pulm}}$, $\text{diag}_{\text{neuro}}$, $\text{diag}_{\text{gastro}}$	LONGITUDE, LATITUDE
Generative time grid T_{gen}	200	200
Path / schedule	cosine diffusion	straight, τ_t on uniform grid
Prediction target	diffusion v-target	flow velocity $v = \epsilon - y_e$
Sampler	DDIM, 50 steps	Euler, 50 steps
Loss	Min-SNR Huber $+0.1\,\text{MSE}$	Huber $+0.05\,\text{MAE}$
Context encoder	GRU, 2 layers	GRU, 2 layers
Velocity network	Transformer, 6 layers, 8 heads	Transformer, 6 layers, 8 heads
Hidden dimension	128	128
Latent dimension d_z	32	32
Target pooling weight λ	1, validation-selected	1, validation-selected
Pooling warmup	Linear $0 \to 1$ over the first 5,000 steps; then fixed at 1	
Router weight γ	1	0
Component KL cap κ_z	∞ (uncapped)	∞ (uncapped)
Batch entropy bonus η_{ent}	0.01	0
Latent conditioning clip	$[-5, 5]$	none
Batch size	128	128
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	3×10^{-4}
Weight decay	10^{-4}	10^{-4}
Max steps	30,000	45,000

B.4 Training Environment and Cost

All experiments run on a single-GPU CUDA workstation with one NVIDIA GeForce RTX 4090 with 24 GiB VRAM, Intel Xeon Platinum 8481C CPUs, and 754 GiB system memory; the reported experiments fit on a single 24 GiB-class GPU. For CLASP with the flow-matching backbone, the total training and inference times across the three WEATHER-5K horizons ($H \in \{24, 72, 168\}$ hours) are 13.6 and 19.9 hours, respectively. On eICU, the corresponding totals across the three horizons ($H \in \{6, 12, 24\}$ hours) are 10.5 hours for training and 45.5 hours for inference.

Table 8 records the forecast draw counts and sampling steps, and Appendix B.5 separates the training and deployment operation counts.

B.5 Scalability of Inferred Responsibilities

The main scalability concern is the number of entities. Inferred responsibilities and component posteriors are computed by amortized minibatch inference: the shared router π_ω and component-posterior head g_ψ operate directly on the current encoded context, without storing persistent per-entity assignments or latent states; the model maintains only cluster-level parameters. Let B_{batch} denote the minibatch size, K the number of clusters, and d_z the latent dimension, treating architecture widths as constants. The costs below describe the CLASP-specific routing and latent computation, not the total forecasting-model complexity. They exclude history encoding and backbone forecast generation, whose costs depend on history length, horizon, architecture, and numerical sampling steps.

Router complexity: The router maps each entity-window summary (h_e, m_e) to K logits followed by a softmax, once per window, so the per-minibatch routing cost is $O(B_{\text{batch}}K)$. Provided labels bypass this router.

Training latent computation: For each pair (e, k) the component-posterior head outputs a diagonal Gaussian with $2d_z$ parameters, so posterior evaluation costs $O(B_{\text{batch}}Kd_z)$; sampling $z_{e,k}$, the diagonal-Gaussian KL terms, and the responsibility-weighted aggregate $z_e = \sum_k r_{ek} z_{e,k}$ share the same scaling. The implementation evaluates all K cluster-conditioned posteriors in a single vectorized pass over a reshaped $[B_{\text{batch}} \times K, \cdot]$ batch.

Deployment latent computation: Deployment omits all component-posterior heads. Inferred routing is still evaluated once per window, and the implemented independent prototype draws and weighted aggregation cost $O(B_{\text{batch}}SKd_z)$ for S forecast trajectories. The prior-mean retrofits instead form μ_e^P in $O(B_{\text{batch}}Kd_z)$ without sampling the CLASP latent. Prototype-embedding aggregation also scales linearly with K at fixed embedding width. These costs are additional to the host's generation and readout costs; increasing K therefore affects deployment as well as training.

Memory: Persistent state consists of the cluster-level prototype parameters $\{\mu_k, \Sigma_k, u_k\}_{k=1}^K$, requiring $O(K(d_z+d_u))$ memory with diagonal covariances, or $O(Kd_z)$ in the reported configuration with $d_u = d_z = 32$; all entity-specific quantities are transient minibatch tensors. Memory for this persistent bank therefore does not grow with the total number of entities N; this statement does not include dataset storage.

Consequence: The latent module has $O(B_{\text{batch}}Kd_z)$ training overhead per minibatch and $O(Kd_z)$ persistent prototype storage, independent of N at fixed batch size. Processing more entities still requires more batches. CLASP is intended to scale with the number of entities N for moderate K, not to support arbitrarily many prototypes.

C Baseline Implementations

C.1 Implementation Sources and Protocol Adaptations

Table 10 summarizes implementation sources and adaptations. Model-specific settings are detailed below and in Table 12. The eICU D3VAE and TSDiff results retain the earlier runs; the WEATHER-5K adaptations do not imply that those eICU results were regenerated under the same configuration.

Fairness and adaptation scope. The common entity splits, histories, and forecast horizons define the evaluation task, while each baseline retains the documented model and sampling configuration below. Context padding, horizon resampling, and rolling-window adjustments accommodate the available inputs and output lengths. Other changes materially affect the evaluated predictor: TSDiff's cross-variable attention changes its architecture, and normalization, draw budgets, and invalid-draw filtering affect inference or evaluation. These differences are reported explicitly rather than treated as equivalent implementations. Here, *native* means the evaluated baseline without CLASP; it may include the adaptations in Table 10 and need not be an unmodified upstream implementation.

Table 10: Implementation sources and major adaptations for native baseline evaluation. The D3VAE and TSDiff adaptations listed here concern WEATHER-5K; their eICU results retain earlier runs. CLASP retrofit conditioning is described separately. These native models have no CLASP clustering, latent, or readout; sampling budgets appear in Table 11.

Baseline	Implementation source	Major adaptation	Reason
D3VAE (WEATHER)	PyTorch recomposition of NsDiff-vendored D3VAE components.	GRU encoding; 48-step conditioning resampled to the forecast horizon.	Match the fixed history and requested forecast length.
TSDiff (WEATHER)	Multivariate re-implementation of conditional TSDiff.	Cross-variable attention; global z-score; 25 DDIM steps and $S = 32$.	Support multiple channels and the evaluation sampling budget.
Timer-XL	Released OpenLTM implementation.	Pad observed history to 96 steps; use autoregressive calls for longer horizons.	Match the token length and requested forecast horizon.
NsDiff	Authors' code with local patches.	Rolling length 24/6 on WEATHER-5K/eICU; invalid-trajectory exclusion.	Keep the rolling window shorter than history and handle divergent draws.
TSFlow	Authors' code with local patches.	Disable hourly lag features; fix the prior context to 48 steps.	Accommodate a history shorter than the upstream lag span.

Sampling budgets. Table 11 separates the forecast draw count, inference steps, and sampler. A training diffusion grid is not an inference-step count: in particular, the evaluated WEATHER-5K D3VAE forecast uses the decoder mean and one score correction despite its 1,000-step training schedule. The retained eICU configurations are identified separately; their missing sampling details are not inferred from the updated WEATHER-5K implementations.

Normalization provenance. The main model and retrofits that reuse its data loaders fit global z-score statistics on training entities only and hold them fixed for validation and testing. Means and variances use population moments (division by n); test data never enter the fit. On WEATHER-5K, `data/weather5k.py` uses the 99 training stations, including for the NsDiff, TSDiff, and TSFlow retrofits. The WEATHER-5K D3VAE runner `clasp_d3vae.py` uses this same train-

Table 11: Sampling budgets for the evaluated probabilistic baselines. S is the requested number of forecasts per window; NsDiff excludes invalid draws. Steps refer to prediction, not the training diffusion grid. NR denotes a retained-run detail not recoverable from the available implementation records. The eICU D3VAE row describes the documented Paddle evaluator, with the shared preprocessing protocol specified below. Timer-XL is a deterministic point forecaster and is omitted here.

Baseline	Dataset	S	Inference steps	Sampler / forecast construction
D3VAE	WEATHER-5K	1	1 score step	Decoder mean, then score correction
D3VAE	eICU (earlier)	1	1 score step	Documented Paddle evaluator: one prediction call using the decoder mean
TSDiff	WEATHER-5K	32	25	DDIM, $\eta = 0$
TSDiff	eICU (earlier)	NR	NR	Earlier text reports 100 steps; actual flags and sampler unrecovered
NsDiff	Both	100	20	Non-stationary diffusion reverse chain
TSFlow	WEATHER-5K	100	32	Euler integration from the conditional OU prior, using EMA

only normalizer and does not load eICU. The TSDiff adaptation replaces upstream per-window mean scaling with this global z-score.

On eICU, `data/eicu.py` fits the six vital-sign channels on training patients only; diagnosis flags are not standardized. The main model and the NsDiff, TSDiff, and TSFlow loader integrations use this rule; loader support does not imply that a TSFlow eICU result is reported. Feature selection and imputation means are also training-only (Appendix D.3.1). The retained eICU D3VAE results follow the same preprocessing protocol as WEATHER-5K D3VAE: global z-score statistics are fitted on training entities only, using population moments, and held fixed for validation and testing. For eICU, the fit uses training patients and the evaluated targets are the six vital signs of the current benchmark. The actual sampling flags of the retained eICU TSDiff runs remain unrecovered, as detailed below.

Native WEATHER-5K NsDiff, TSFlow, and Timer-XL use the same global z-score statistics as their CLASP retrofits, fitted on the 99 training stations and held fixed for validation and testing. Timer-XL control runs use this same fit. Its RevIN module is a separate model-level operation. Native eICU Timer-XL uses training patients only; no Timer-XL CLASP retrofit is evaluated on eICU.

D3VAE: D3VAE (Li et al., 2022) combines a coupled diffusion process with a bidirectional variational auto-encoder and disentangled latents. On WEATHER-5K we run a re-implementation under the unified protocol of Appendix D.1: `baselines/D3VAE/d3vae_torch.py` recomposes the diffusion, encoder, score-network, and embedding components vendored with NsDiff. The evaluated configuration uses 1,000 training diffusion indices with the linear coupled schedule using β_{end}=0.1 and scale 0.1, an NVAE-style bidirectional VAE, a GRU temporal encoder, the 48-step conditioning linearly resampled to the horizon, and the composite loss $\mathcal{L} = \psi\,\mathcal{L}_{\text{ELBO}} + \lambda_1\,\mathcal{L}_{\text{score}} + \mathcal{L}_{\text{MSE}} - \gamma\,\text{TC}$ with ψ=0.5, λ_1=1.0, and γ=0.01. These are D3VAE-local objective coefficients: in particular, this γ weights total correlation, independently of CLASP's router weight γ in equation 5. The separate earlier eICU implementation is the Paddle entry point `baselines/D3VAE/main.py`. Its documented command also specifies 1,000 training indices and $\beta_{\text{end}} = 0.1$; omitting these flags instead selects the Paddle CLI defaults of 100 and 0.01. Neither training grid is a forecast-time reverse chain.

Score-loss implementation. In the evaluated Paddle-compatible path, the detached score-loss construction differentiates without creating a higher-order graph. The PyTorch recomposition reproduces this behavior with `score_create_graph=False`, explicitly selected by `--score-create-graph 0` in the wrapper. Consequently, this score-loss term supplies no gradient to the score-network parameters. The separately vendored `src/models/D3VAE.py` uses `create_graph=True` and is not the evaluated wrapper path. This is a property of the specified implementation and configuration, not a claim about every D3VAE implementation. The observation follows from code inspection; no post-backward score-network gradient test is reported.

For WEATHER-5K, calendar covariates are unused and training uses Adam at learning rate 5×10^{-3}, held for the first two pseudo-epochs of 1,500 steps and halved after each pseudo-epoch thereafter,

early stopping after five pseudo-epochs without validation improvement, batch size 16, and a 30,000-step cap. The reported forecast uses $S = 1$: the wrapper calls `predict(draw=False)` for the decoder mean, with one score correction of scale 0.001, rather than a reverse diffusion chain. The Paddle evaluator likewise calls `pred()` once and uses `output.mu`. This specifies the output estimator without asserting that all internal VAE latents are deterministic.

TSDiff: TSDiff (Kollovieh et al., 2023) is a diffusion-based forecasting framework with an unconditional model plus inference-time self-guidance and a conditional variant; we use the conditional variant with observation-mask features and no self-guidance. Since it was designed for univariate series, the WEATHER-5K rows are a multivariate re-implementation run under the unified protocol of Appendix D.1. The architecture uses one bidirectional S4 stream per variable with state size 128, 6 residual blocks, and hidden width 128; a cross-variable Transformer encoder layer after every block; and a learned 16-d variable-identity feature. Each Transformer layer uses 4-head self-attention over the six per-time-step variable tokens of width 128, a GELU feed-forward layer of width 512, and residual connections. The features are the context with zeros in the horizon and the observation mask, with no lag features or calendar encodings, and the protocol's training-only global z-score replaces the per-window mean scaling (Appendix C.1). Training follows the upstream optimizer recipe: ϵ-prediction L_2 loss over context and horizon without context re-imposition, a linear beta schedule from 10^{-4} to 0.1 over 100 steps, Adam at learning rate 10^{-3} with gradient clipping 0.5, a plateau schedule whose reduction condition was not reached in these runs, and no EMA. The protocol's training budget uses batch size 32 and 30,000 steps, with validation every 500 steps and early stopping patience 10. Sampling uses the upstream DDIM sampler with η=0 and 25 steps. The main WEATHER-5K evaluation and both the native and CLASP retrofit rows in Table 27 use S=32 draws. These are explicit wrapper overrides of the runner defaults ($S = 100$, DDPM, 100 sampling steps). The retained eICU results come from the previous submission. Although the earlier text reports 100 steps, the current launch script does not run TSDiff on eICU and no saved eICU result files recover its actual flags; the draw count, sampler, and step count therefore remain unverified. Defaults do not establish those earlier evaluation settings. A separate timing benchmark of the 100-step ancestral sampler with S=100 draws costs 4.2 s per window at H=24 on our GPU, more than ten times the cost of the chosen WEATHER-5K setting; this benchmark does not identify the retained eICU configuration.

Timer-XL: Timer-XL (Liu et al., 2025) is a long-context pre-trained Transformer for unified forecasting. **Zero-shot:** the pre-trained checkpoint with 8 layers, d_{model}=1024, 8 heads, and token length 96 is evaluated without weight updates. **Fine-tuning:** all parameters are adapted with the default recipe: Adam with learning rate 5×10^{-6}, cosine annealing with $T_{\max}$=10, batch size 256, early stopping patience 3, and RevIN normalization for 10 epochs.

The native WEATHER-5K Timer-XL rows use the same global z-score fit on the 99 training stations as the CLASP and control rows. The token-length and autoregressive horizon handling are summarized in Table 10; the CLASP retrofit uses the conditioning pathway described in Appendix C.2.

NsDiff: NsDiff (Ye et al., 2025) is a non-stationary diffusion forecaster that models a time-varying noise scale jointly with the conditional mean. We use the released model with the following configuration: a non-stationary Transformer conditioner with d_{model}=512, d_{ff}=1024, 2 encoder layers and 1 decoder layer; 20 diffusion steps with a linear beta schedule for β from 10^{-4} to 10^{-2}; a two-layer noise-scale network with 64 hidden units; and Adam at learning rate 10^{-3}. Training runs for 30,000 steps with batch size 32, and evaluation draws S=100 samples under the protocol of Appendix D.1. Its trailing-window estimator of the noise scale requires a window strictly shorter than the history, so we set the rolling length to 24 on WEATHER-5K with its 48-step context and 6 on eICU with the same 12-step context as the other eICU baselines. Its reverse chain occasionally diverges: draws with any normalized forecast coordinate exceeding 50 in absolute value, or with non-finite values, are treated as invalid and excluded from the aggregate. Approximate invalid-draw rates are 15% on WEATHER-5K and 10% on eICU.

TSFlow: TSFlow (Kollovieh et al., 2025) is a conditional flow-matching forecaster with Gaussian-process priors, and is the closest published relative of our flow-matching instantiation. The inherited hourly conditional recipe comprises an S4 backbone with 3 residual blocks, hidden width 64, and step embedding 64; an Ornstein–Uhlenbeck prior kernel with γ=1; 32 Euler steps with random

coupling; AdamW at learning rate 10^{-3} with gradient clipping 0.5; and EMA decay 0.9999. Our reported training budget is 30,000 steps with batch size 64. Two context-driven adaptations accommodate the 48-step history: hourly lag features are disabled because the upstream lag set needs roughly 672 steps, and the GP prior's context is fixed to all 48 observed steps at every horizon. The evaluated upstream implementation constructs this prior only for horizons that are multiples of the 24-hour period, so TSFlow is reported on WEATHER-5K only. A separate implementation correction copies the S4 kernels' integer state buffer from the online network to the EMA network before evaluation, identically for native and CLASP-enhanced sampling.

Table 12: Baseline training hyperparameters on eICU and WEATHER-5K. The time-grid/solver row distinguishes training discretization from reverse sampling; Table 11 gives the inference budgets separately. The eICU TSDiff step count is retained as previously reported, but its evaluation flags could not be recovered.

	D3VAE		**TSDiff**		**Timer-XL FT**		**NsDiff**	**TSFlow**
	eICU	Weather	eICU	Weather	eICU	Weather	eICU / Weather	Weather
Optimizer	Adam		Adam		Adam		Adam	AdamW
Learning rate	5×10^{-3}		10^{-3}		5×10^{-6}		10^{-3}	10^{-3}
LR schedule	Halving/epoch	Halving/1.5k-step pseudo-ep.	ReduceOnPlateau		Cosine annealing		constant	constant
Loss	Composite		MSE		MSE		KL + MSE	CFM
Training length	20 ep.	30k steps	50 ep.	30k steps	10 ep.	10 ep.	30k steps	30k steps
Batch size	128	16	128	32	256	256	32	64
Early stop	5	5 pseudo-ep.	5	10 val.	3	3	—	—
Time grid / solver	1,000-step training grid		100 steps (unverified)	100-step grid 25 DDIM steps	—		20 reverse	32 Euler
Noise schedule / prior	Linear		Linear		—		Linear	OU GP, γ=1
Hidden dim	128		128		1,024		512	64
Layers	2		6		8		2 enc. + 1 dec.	3 S4
Other	—	GRU enc.	cross-var. attn.	cross-var. attn., 4-head, 128	RevIN		rolling len. 6 / 24	EMA 0.9999
Context length	12	48	12	48	12	48	12 / 48	48
Prediction horizons	6, 12, 24 / 24, 72, 168		6, 12, 24 / 24, 72, 168		6, 12, 24 / 24, 72, 168		6, 12, 24 / 24, 72, 168	24, 72, 168

C.2 CLASP Retrofit Interfaces

Retrofit variants: The native comparison omits the CLASP module and readout according to the plain-backbone convention in Appendix B.1, while retaining each host's own encoders, latent variables, and loss. Table 27 extends the basic CLASP backbone pipelines in Figures 3–8 to five baseline forecasters on WEATHER-5K with provided zones ($K = 5$). Figures 9–13 show the corresponding retrofits; below, we describe their differences from the basic pipelines. Each + CLASP variant adds the latent module of Algorithm 1, trains it jointly with the host under the host's forecasting objective plus the pooling penalty, and applies the validation-fitted anchor blend of equation 12. These comparisons evaluate the complete CLASP pipeline and do not isolate latent conditioning from the validation-fitted readout. Native and CLASP variants share the same training-fitted normalization (Appendix C.1). Table 13 summarizes the shared module's architecture-specific interfaces and inference procedures. Zero-initialized FiLM preserves the native backbone's output at initialization, before training and decoding. Under a shared random seed and identical preprocessed inputs, an identity gate verifies bit-exact agreement with the native output at initialization. This check concerns the conditioning interface. The Timer-XL FiLM condition vector is shared across channels and tokens.

Architecture-specific conditioning interfaces: As in the basic figures, the context encoder produces h_e, the prototype bank supplies $u_e = \sum_k r_{ek} u_k$, and the posterior heads parameterize $q_{e,k}$. Independent component-posterior draws aggregate to $z_e \sim q_e^{\text{agg}}$ during training. The resulting $g_e = [h_e, u_e, z_e]$ conditions the host through equation 11, with FiLM applied at the locations specified below. For these WEATHER-5K experiments, the responsibilities block uses the provided zone labels ($K = 5$), so r_e is one-hot and the entropy term is inactive. NsDiff, TSFlow, and TSDiff redraw $z_e \sim P_e^{\text{agg}}$ for each sampled trajectory. Timer-XL and D3VAE instead use the prototype-mean CLASP latent at inference and produce one forecast (S=1). These two retrofit diagrams explicitly use $z_e = \mu_e^P = \sum_k r_{ek}\mu_k$, the aggregate prior mean. This preserves each host's single-forecast protocol without introducing additional CLASP latent-sampling variability (Appendix B.1); D3VAE retains its host-native decoder-mean prediction path and score correction.

Table 13: CLASP retrofit interfaces and inference on WEATHER-5K with provided zones ($K = 5$). Each host uses the shared representation $g_e = [h_e, u_e, z_e]$ through its listed interface. The test-time CLASP latent uses P_e^{agg} or its mean μ_e^P, as listed. "Blend" is the fixed validation-fitted two-anchor readout, followed by the arithmetic mean of the forecasts (the single forecast when $S = 1$); NsDiff retains its invalid-draw exclusion rule. These are complete-pipeline comparisons, including conditioning and readout. Training latent sources also appear in Table 8.

Backbone	Conditioning mechanism	Test latent	S	Sampling steps / calls	Readout
NsDiff	FiLM after first three conditional-MLP denoiser layers	$z_e \sim P_e^{\text{agg}}$	100	20 non-stationary reverse steps	Blend
TSFlow	FiLM at every S4 block input	$z_e \sim P_e^{\text{agg}}$	100	32 Euler steps using EMA	Blend
Timer-XL	FiLM on decoder token embeddings	$z_e = \mu_e^P$	1	1 call; 2 at $H = 168$	Blend
D3VAE	FiLM on temporal/value conditioning before horizon resampling	$z_e = \mu_e^P$	1	Decoder mean and 1 score step	Blend
TSDiff	FiLM at every S4 denoising block	$z_e \sim P_e^{\text{agg}}$	32	25 DDIM steps, $\eta = 0$	Blend

NsDiff (Figure 9): Relative to the diffusion pipeline in Figure 4, NsDiff uses a conditional MLP denoiser, with FiLM applied after each of its first three layers. The host additionally supplies a conditional mean predictor and a noise-scale network driven by a 24-step trailing-window variance estimate. Its 20-step non-stationary reverse process replaces the basic pipeline's 50-step DDIM sampler; the S=100 forecast aggregation and CLASP readout are retained. The native model and retrofit share the same training-fitted z-score normalizer (Appendix C.1).

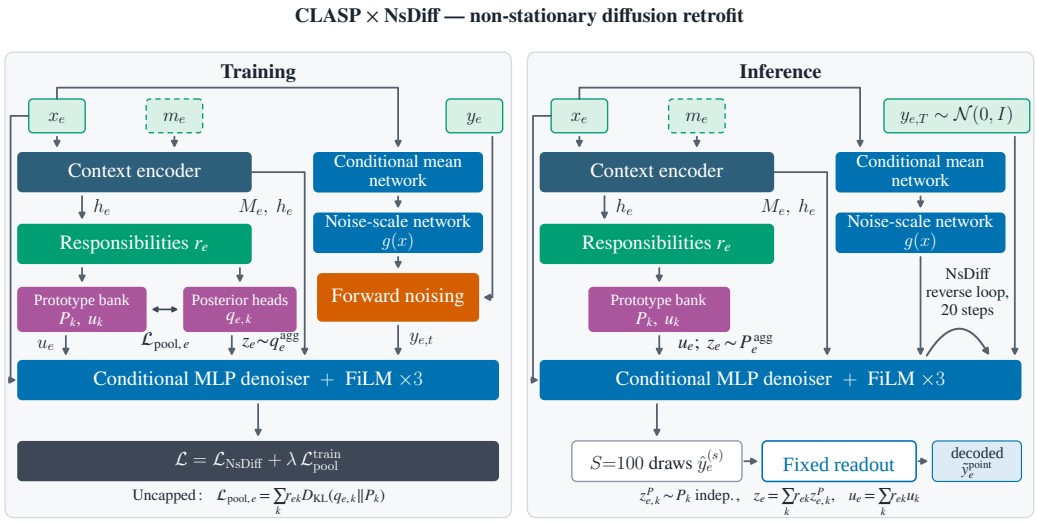

Figure 9: CLASP retrofitted onto NsDiff. The conditioning pathway is that of Figure 4; the host contributes its conditional mean and noise-scale networks, FiLM modulates the first three layers of its conditional MLP denoiser, and sampling uses NsDiff's 20-step non-stationary reverse loop.

TSFlow (Figure 10): TSFlow follows the flow-matching pipeline in Figure 3, using a three-block S4 velocity network with cross-channel attention and FiLM at every block input. The initial state comes from a history-conditioned Ornstein–Uhlenbeck Gaussian process in place of the standard Gaussian prior. Sampling uses 32 Euler steps through the exponential moving average (EMA) network and S=100 draws, matching the native TSFlow configuration in Table 1. The retrofit applies

CLASP conditioning and decoding with the same training-fitted normalizer as native TSFlow; the basic flow-matching pipeline uses 50 Euler steps.

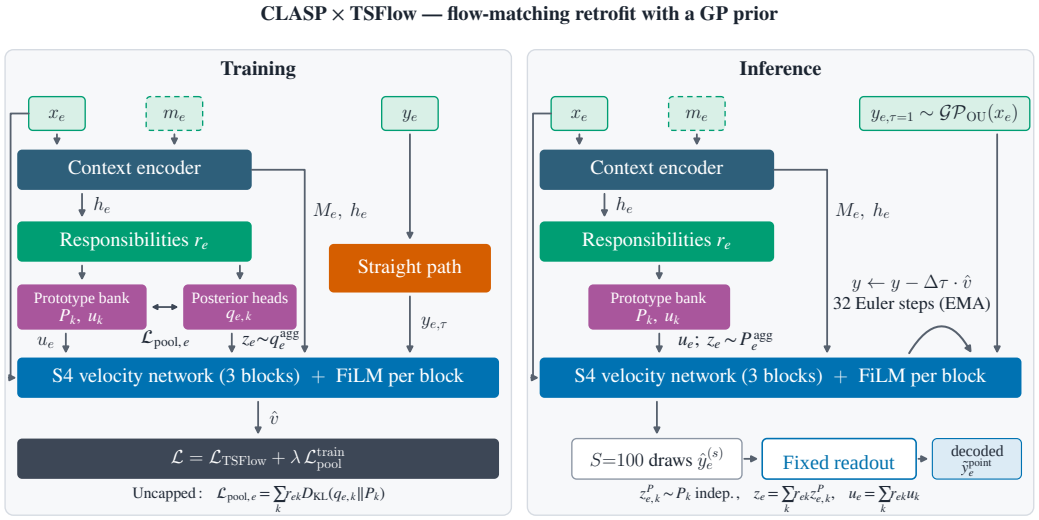

Figure 10: CLASP retrofitted onto TSFlow. FiLM modulates the input of each S4 block of the velocity network; the initial state comes from an Ornstein–Uhlenbeck Gaussian-process prior conditioned on the history, and the probability-flow ODE is integrated with 32 Euler steps through the exponential moving average.

Timer-XL (Figure 11): The Transformer pipeline in Figure 5 provides the basic layout. Timer-XL applies FiLM to the pretrained decoder-only model's token embeddings in place of the basic forecaster's learned prefix token. Its 48-step history is replicate-padded to the 96-step token length, and H=168 requires two autoregressive calls. Both pipelines use S=1; Timer-XL uses the prototype-mean CLASP latent instead of a prior draw. Training continues the fine-tuned checkpoint used for the Timer-XL FT results in Table 1. The native Timer-XL row in Table 27 repeats those FT results, while the +CLASP row reports the model with CLASP conditioning and decoding. The native, CLASP, and control runners share the same z-score statistics fitted on the 99 training stations (Appendix C.1).

D3VAE (Figure 12): D3VAE reuses the g_e conditioning and CLASP readout interfaces of Figures 3–8, while retaining its own coupled-diffusion and bidirectional-VAE forecasting blocks. The injection point is the conditioning sequence formed by concatenating the temporal encoding and value embedding: FiLM acts before this sequence is resampled from 48 steps to the forecast horizon. Inference fixes the CLASP latent to the prototype mean; the host returns the decoder mean with a single score correction (S=1), without an additional decoder-output draw.

TSDiff (Figure 13): TSDiff follows the diffusion pipeline in Figure 4, replacing the DiT denoiser with bidirectional S4 blocks and cross-variable attention, with FiLM applied at every block. It retains Gaussian-noise initialization and DDIM sampling, using 25 reverse steps and S=32 draws for the retrofit comparison, compared with 50 steps and S=100 in the basic pipeline.

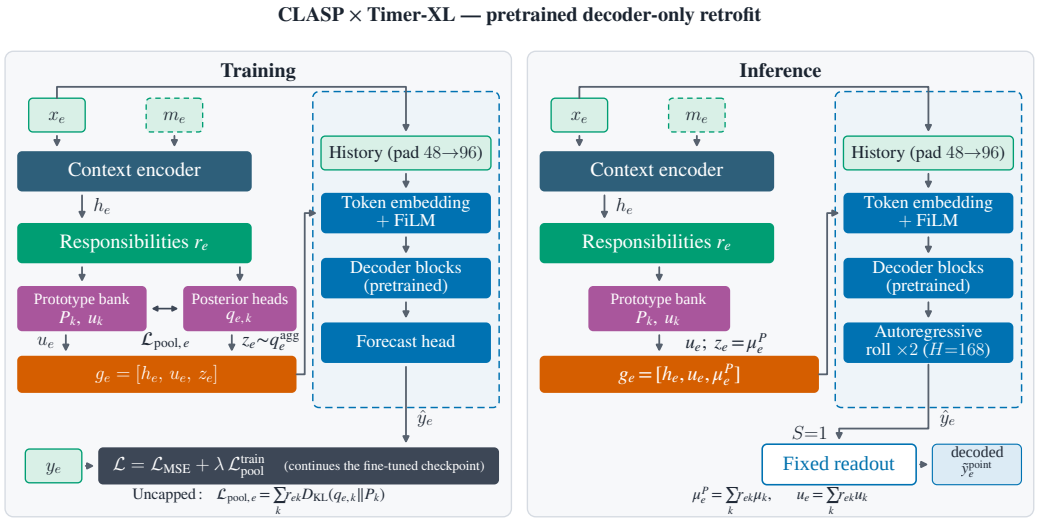

Figure 11: CLASP retrofitted onto Timer-XL. FiLM modulates the token embedding of the pre-trained decoder-only model, the history is replicate-padded from 48 to 96 steps, and H=168 is produced by two autoregressive calls from a deterministic prototype-mean latent.

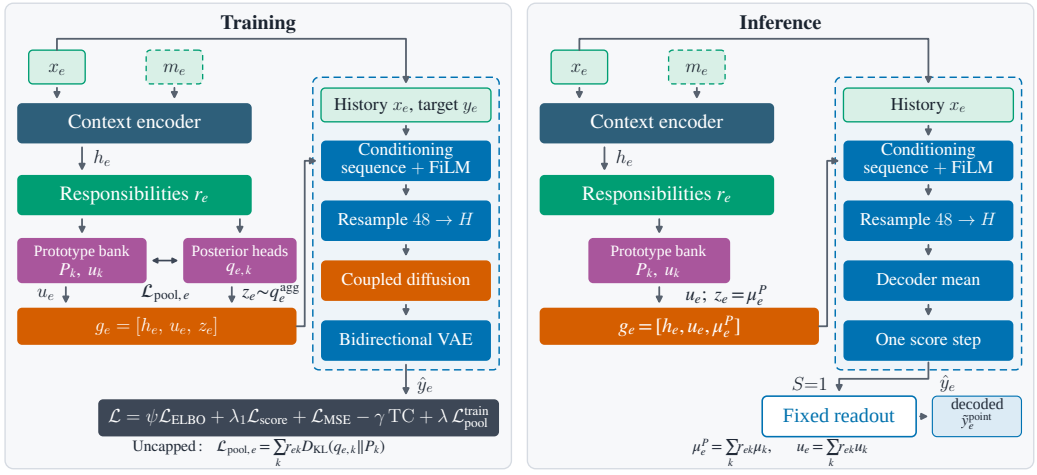

Figure 12: CLASP retrofitted onto D3VAE. FiLM modulates the conditioning sequence built from the temporal encoding and the value embedding, before it is resampled from 48 steps to the horizon; the host keeps its coupled diffusion, bidirectional variational auto-encoder and single score step. The displayed γ is D3VAE's total-correlation coefficient, not CLASP's router weight.

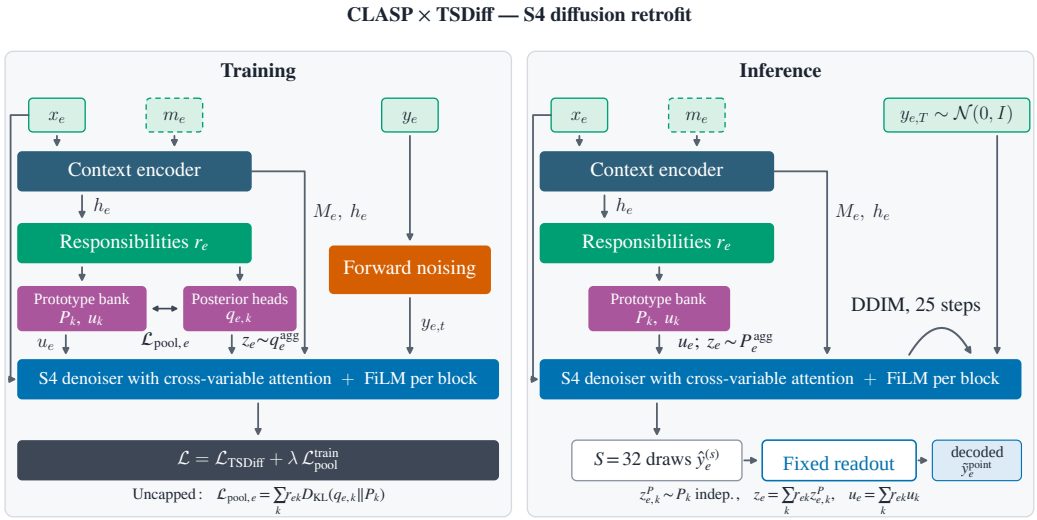

Figure 13: CLASP retrofitted onto TSDiff. Relative to Figure 4, the denoiser uses S4 blocks with cross-variable attention and FiLM at every block; sampling uses 25 DDIM steps and $S = 32$ draws, matching the native TSDiff comparison.

D Datasets and Experimental Protocol

D.1 Unseen-Entity Evaluation Protocol

Entity splits and windows. The balanced WEATHER-5K subset contains 25 stations from each of five geographic zones. Three stations per zone are held out for testing, giving 15 test entities. Within each zone, the three stations with the smallest station identifiers are assigned to the test set, so the holdout is deterministic by sorted ID rather than a random draw. Of the remaining 110 stations, 11 are reserved for validation and 99 for training. Training entities determine the learned model parameters. Table 15 summarizes the zone-wise training/validation pool and test split before the validation holdout. The eICU partition contains 76,834 training, 8,529 validation, and 7,923 test patients, with no patient shared across splits.

What validation determines. Validation entities support checkpoint selection and early stopping, selection of the target pooling weight $\lambda = 1$ (Appendix B.3), and fitting of the lead-specific forecast-readout coefficients. Readout fitting includes both the prototype-specific profiles and the pooled global profile used for low-support fallback and shrinkage; the plain-backbone readout control uses the global profile alone. The 200-window fallback threshold and 0.5 shrinkage weight are fixed procedure settings, not parameters fitted to test data. Normalization statistics use training entities only.

Before test evaluation, all selected checkpoints, normalization statistics, prototype and router parameters, and readout profiles are fixed. Nothing is fitted or refitted on test entities. Computing a test window's encoder states, responsibilities, history-based anchors, or readout-profile index is an application of this fixed prediction rule, rather than a parameter update.

Window construction. Entities are assigned to disjoint splits before their forecast windows are constructed; the train/validation/test partition is by identity, not by successive time blocks of the same entity. For a preprocessed trajectory segment $(v_{e,1}, \ldots, v_{e,L_e})$, a forecast origin t gives context $x_e(t) = v_{e,t-T+1:t}$ and target $y_e(t) = v_{e,t+1:t+H}$, requiring a complete $T + H$ segment. For eICU, each such segment belongs to a single ICU stay: both the context and the entire forecast target must remain within that stay, even when a patient has multiple stays. All stays of a patient belong to the same entity split. WEATHER-5K uses $T = 48$ and $H \in \{24, 72, 168\}$; eICU uses $T = 12$ and $H \in \{6, 12, 24\}$, in hourly steps. Figure 14 separates the identity split from this within-entity timeline. Training and validation windows use stride 12 on both datasets. eICU test windows also use stride 12; the WEATHER-5K forecasting comparisons retain stride-24 test windows. Temporal overlap does not turn these windows into independent observations. The non-overlapping initial/later blocks in Appendix E.4.3 belong to a separate fixed-predictor diagnostic, not to the train/validation/test split.

Normalization. CLASP and the retrofits using its loaders fit global z-score statistics on training entities only: 99 stations on WEATHER-5K and training patients on eICU. Means and variances use population moments (division by n), with no test data in the fit. The eICU fit standardizes the six vitals; diagnosis indicators remain unstandardized. Feature selection and imputation means also use training patients only. The WEATHER-5K D3VAE and TSDiff adaptations use the same train-only protocol; the retained eICU D3VAE results likewise use training-patient statistics and the six-vital target set. Native eICU Timer-XL also uses training-patient statistics.

Native WEATHER-5K NsDiff, TSFlow, and Timer-XL share the same 99-training-station z-score fit as their CLASP retrofits; Timer-XL control runs also use this fit. This global fit is distinct from Timer-XL's model-level RevIN operation. Appendix C.1 gives the dataset-loader and implementation-specific provenance, including the NsDiff and TSFlow integrations. Normalization is fixed before testing, including for baselines that inherit upstream code.

Evaluation configurations and point forecasts. Table 1 uses flow-matching CLASP with provided WEATHER-5K zones ($K = 5$), as do the six-backbone and external-retrofit comparisons. The additional WEATHER-5K main comparison, component ablations, and inference-source analysis instead use inferred responsibilities ($K = 4$), except where a control explicitly varies K or label availability. eICU uses inferred responsibilities ($K = 4$); its four diagnosis indicators are metadata,

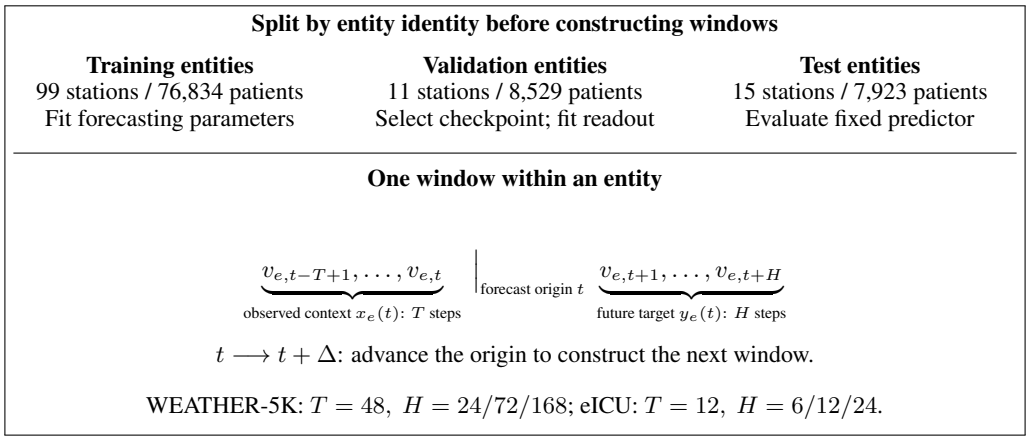

Figure 14: Unseen-entity evaluation and forecast-window construction (schematic, not to scale). Counts are stations for WEATHER-5K and patients for eICU. Every context and target belongs to its entity's assigned split; eICU contexts and targets additionally remain within one ICU stay. The target is observed for scoring but is not an input to the forecast. The stride Δ is 12 for training and validation on both datasets and for eICU testing, and 24 for WEATHER-5K testing.

not cluster labels. In the headline WEATHER-5K configuration, each window produces $S = 100$ trajectories using 50 forward-Euler integration steps per trajectory. The point forecast is the ordinary mean of all anchor-blended draws. Table 2 reports inferred clustering with $K = 4$ and uses the same blend-and-mean readout, with the domain-specific anchors defined below. The inferred-cluster WEATHER-5K comparison reuses the baseline rows from the provided-cluster evaluation. Baseline-specific implementations and sampling settings are given in Appendix C.1. The backbone comparison in Table 4 uses the same stride-24 WEATHER-5K test windows. Plain variants follow Appendix B.1 and return unmodified forecasts; full CLASP combines latent conditioning with validation-fitted readout and sample-mean aggregation. Deterministic families use one forward pass and one latent draw per window, whereas DiT and flow matching generate $S = 100$ draws, with the CLASP variants averaging all blended draws. Plain DiT and flow matching use the arithmetic mean of their unblended draws; the latter reuses the raw plain-backbone reference in Table 3. These comparisons are interpreted within the backbone experiment rather than across differing configurations. The separate WEATHER-5K fixed-predictor evaluation uses aggregate posterior sampling with the 16-draw Monte Carlo protocol in Appendix E.4.2; the forecasting comparisons use aggregate deployment prior sampling by default. The latent-source controls in Table 6 explicitly replace it with aggregate posterior sampling or the aggregate deployment prior mean. The prototype-mean retrofits follow their architecture-specific inference rules in Appendix C.2.

Seeds. Unless otherwise stated, forecasting results are reported as the mean over five independent training runs with different random seeds. Table 21 reports the per-seed MAE and MSE for the headline flow-matching CLASP model on WEATHER-5K at each horizon, together with the across-seed standard deviation. Across the five seeds the standard deviations are below 0.001 for both metrics. This observed spread describes optimization randomness for these runs, rather than uncertainty in population-level performance; the individual values are listed in full. Repeated-seed training-support diagnostics are summarized in Table 39.

Station-level uncertainty. Training-seed variation and variation across the 15 held-out stations measure different sources of variability. Appendix E.1.3 reports both, including the per-station and per-zone aggregation rules; neither summary alone establishes population-level uncertainty or paired significance between methods.

D.1.1 Forecast readout and inference

Context-based anchors. Both anchors use only the preprocessed history values available for the current forecast. Persistence repeats $x_{e,T}$ across all leads. For WEATHER-5K, define $\mathcal{P}_\ell = \{t \in$

$\{1, \dots, T\} : t \equiv T + \ell \pmod{24}\}$. The climatology anchor at lead ℓ is $y^{\text{clim}}_{e,\ell} = |\mathcal{P}_\ell|^{-1} \sum_{t \in \mathcal{P}_\ell} x_{e,t}$, averaging the two matching history values when $T = 48$. The eICU readout uses the context-mean anchor $y^{\text{clim}}_{e,\ell} = T^{-1} \sum_{t=1}^{T} x_{e,t}$ with $T = 12$, repeated at every lead. Thus its short history does not require an unobserved full diurnal cycle.

Each λ_j denotes a prototype- and lead-specific profile $(\lambda_{j,\hat{c}_e,\ell})_{\ell=1}^{H}$, broadcast across channels, with $\hat{c}_e = c_e$ for provided labels and $\hat{c}_e = \arg\max_k r_{ek}$ for inferred assignments. Products in equation 12 are elementwise, and

$$\lambda_{1,k,\ell}, \lambda_{2,k,\ell} \geq 0, \qquad \lambda_{1,k,\ell} + \lambda_{2,k,\ell} \leq 1 \quad \text{for all } k, \ell. \tag{67}$$

The profiles are fitted on validation entities, satisfy equation 67, and remain fixed during testing. These profiles are distinct from the training-objective weights in equation 5.

Anchor blending and validation fitting. For each validation window, the regression base is the raw sample mean $\overline{\widehat{y}}_e = S^{-1} \sum_s \widehat{y}^{(s)}_e$. At each prototype and lead, least squares regresses $y_e - \overline{\widehat{y}}_e$ on the two anchor residuals $y^{\text{pers}}_e - \overline{\widehat{y}}_e$ and $y^{\text{clim}}_e - \overline{\widehat{y}}_e$. Sufficient statistics sum equally over validation windows and channels, using the configured evaluation space (original units by default). Non-finite residual entries are excluded from these sums. The two-anchor normal equations are solved in float32, with their determinant floored from below at 10^{-8}; a single enabled anchor uses the corresponding scalar regression with the same denominator floor and sets the disabled coefficient to zero.

The fitting order is least squares, clamp, low-support fallback, then shrinkage. Given an unconstrained coefficient pair $(\widehat{\lambda}_1, \widehat{\lambda}_2)$, the clamp computes

$$a_j = \min\{1, \max\{0, \widehat{\lambda}_j\}\}, \qquad \lambda^{\text{clamp}}_j = \frac{a_j}{\max\{1, a_1 + a_2\}}, \quad j \in \{1, 2\}.$$

This produces feasible blend weights rather than solving constrained least squares exactly. A global profile is fitted from the pooled validation sufficient statistics and clamped in the same way. Prototypes with fewer than 200 validation windows receive this global profile. Every prototype profile is then averaged equally with the global profile (shrinkage weight 0.5). No final clamp is applied: both inputs are feasible and their convex average satisfies equation 67. Support counts include all assigned validation windows, even if some residual entries are non-finite.

Profiles are fitted only on validation entities and held fixed on test entities. Profile selection uses c_e for provided labels and $\hat{c}_e = \arg\max_k r_{ek}$ for inferred assignments; the selected lead-specific weights are broadcast across channels. This index selects only the readout profile: the latent construction retains the full responsibility vector and uses P^{agg}_e (or its mean in the stated retrofits), without categorical component selection. WEATHER-5K validation includes three Tropical stations and two in each other zone, but the fallback criterion uses the number of assigned windows. The same affine blend applies to each probabilistic trajectory or to a deterministic forecast. Raw configurations bypass it. For the plain-backbone readout controls in Tables 3 and 23, the lead-specific profile is fitted globally by pooling all validation windows, using the same anchors and coefficient-fitting procedure. This single profile is shared across entities, without cluster assignments or router-based profile selection.

Point aggregation. For decoded forecasts, the point estimator averages all blended draws at each forecast lead ℓ and target channel j:

$$\widetilde{y}^{\text{point}}_{e,\ell j} = \frac{1}{S} \sum_{s=1}^{S} \widetilde{y}^{(s)}_{e,\ell j}. \tag{68}$$

The anchors and coefficients are identical across draws for a given window, so averaging commutes with the affine blend in equation 12. For $S = 1$, this returns the single blended forecast. "Raw CLASP" and "no readout" both mean that the anchor blend is bypassed; neither name changes the latent-conditioned forecast draws. All reported raw CLASP point forecasts use the arithmetic sample mean: the WEATHER-5K controls in Tables 3 and 23 and the eICU row in Table 24. The plain DiT and flow-matching rows underlying Table 4 also average their raw draws. Thus the reported raw-versus-decoded comparisons keep point aggregation fixed and change the anchor blend. For $S = 1$, the raw point forecast is the single unblended output.

Predictive distributions and intervals. Headline MAE/MSE use the validation-fitted anchor blend followed by arithmetic-mean aggregation. CRPS and empirical coverage instead use raw, unblended predictive draws. Interval construction is a separate operation from both point aggregation and anchor blending. When requested for a probabilistic ensemble with $S > 1$, interval endpoints are the elementwise P10 and P90 of the raw, unblended draws, using linear quantile interpolation. Deterministic configurations and single forecasts produce no intervals. Appendix E.1.2 reports empirical coverage against observed targets; the nominal central-80% interval does not ensure 80% empirical coverage. Algorithm 2 combines latent sampling, optional blending, point aggregation, and the separate raw-interval readout for unfiltered ensembles. NsDiff excludes invalid draws before aggregation as described in Appendix C.1. The aggregate-posterior source control in Table 6 substitutes $q_{e,k}$ for P_k in the sampling branch while retaining the comparison's forecast-sampling budget and readout.

Algorithm 2 CLASP inference with experiment-specific forecast readout.

Require: Trained model; history x_e; optional metadata m_e and label c_e
Require: Draw count S; readout and interval options; fitted blend profiles when readout is enabled
Require: Raw point estimator A_{raw}: arithmetic sample mean for all reported raw-forecast comparisons
Require: CLASP latent mode: aggregate-prior sample (default) or aggregate-prior mean (Timer-XL and D3VAE retrofits)

Encode $h_e \leftarrow f_\theta(x_e, m_e)$ and retain sequence states M_e
$r_e \leftarrow \text{onehot}(c_e)$ or $\pi_\omega(h_e, m_e)$; $u_e \leftarrow \sum_k r_{ek} u_k$
if readout is enabled **then**
    Select the profile using $\hat{c}_e = c_e$ or $\hat{c}_e = \arg\max_k r_{ek}$ and form the context-based anchors
                                                ▷ Readout index only; latent weights remain r_e
end if
for $s = 1, \ldots, S$ **do**
    if CLASP latent mode is aggregate-prior mean **then**
        $z_e^{(s)} \leftarrow \mu_e^P = \sum_k r_{ek} \mu_k$
    else
        Draw $z_{e,k}^{(s)} \sim P_k$ independently over k and s
        $z_e^{(s)} \leftarrow \sum_k r_{ek} z_{e,k}^{(s)}$ ▷ $z_e^{(s)} \sim P_e^{\text{agg}}$
    end if
    $g_e^{(s)} \leftarrow [h_e, u_e, z_e^{(s)}]$
    $\widehat{y}_e^{(s)} \leftarrow \textsc{BackboneSample}(x_e; M_e, g_e^{(s)})$ ▷ backbone-specific
    if readout is enabled **then**
        Form $\widetilde{y}_e^{(s)}$ using equation 12
    end if
end for
if readout is enabled **then**
    $\widetilde{y}_e^{\text{point}} \leftarrow S^{-1} \sum_{s=1}^S \widetilde{y}_e^{(s)}$ ▷ all blended draws; equation 68
else
    $\widehat{y}_e^{\text{point}} \leftarrow \mathsf{A}_{\text{raw}}(\{\widehat{y}_e^{(s)}\}_{s=1}^S)$
end if
Select $\widetilde{y}_e^{\text{point}}$ as the point output if readout is enabled, otherwise $\widehat{y}_e^{\text{point}}$
if the backbone is probabilistic, $S > 1$, and intervals are requested **then**
    for $p \in \{0.1, 0.9\}$ **do**
        $\widehat{q}_{p,e}^{\text{raw}} \leftarrow \text{Quantile}_p\left(\{\widehat{y}_e^{(s)}\}_{s=1}^S\right)$ ▷ all raw draws
    end for
    return point output and $[\widehat{q}_{0.1,e}^{\text{raw}}, \widehat{q}_{0.9,e}^{\text{raw}}]$
else
    return point output
end if

D.2 WEATHER-5K DETAILS

The provided-zone experiments use $K = 5$ and one-hot responsibilities from the geographic grouping below. The inferred experiments retain the same station split but use a learned $K = 4$ router, unless a control varies K; their responsibilities are not the five zone labels. Appendix D.1 identifies the configuration of each main comparison, ablation, and retrofit study.

D.2.1 CIRCULAR ENCODING OF WIND DIRECTION

Wind direction is defined on a circle rather than on the real line, so direct prediction in degrees is poorly aligned with Euclidean losses: 1° and 359° are nearly identical directions, yet their raw difference is 358°, creating an artificial discontinuity at the $0^\circ/360^\circ$ boundary. With $\theta = \frac{\pi}{180}$ WND_ANGLE, we encode wind direction by the circular embedding

$$\text{WND_ANGLE_sin} = \sin\theta, \qquad \text{WND_ANGLE_cos} = \cos\theta, \tag{69}$$

which places wind direction on the unit circle: for example, $1^\circ \mapsto (0.0175,\ 0.9998)$ and $359^\circ \mapsto (-0.0175,\ 0.9998)$ are close, correctly reflecting their directional similarity. We predict and evaluate WND_ANGLE_sin and WND_ANGLE_cos directly in this transformed space, keeping training and evaluation consistent with the circular representation.

D.2.2 SAMPLING AND GROUPING

For the provided-cluster experiments we partition stations into five broad geographic zones: tropical, dry, temperate, continental, and polar. This choice is intended to be interpretable rather than taxonomic: large-scale weather dynamics vary systematically with latitude, so these zones provide a simple, meaningful grouping with broadly distinct climatic regimes. We construct a balanced subset by sampling the same number of stations from each zone and reserving disjoint training/validation and test entities. The 110-station training/validation pool is subsequently split into 99 training and 11 validation stations as in Appendix D.1. Table 14 summarizes the five-zone grouping, Table 15 the split sizes, and Table 16 the exact station IDs. Figure 15 visualizes the sampled stations: the subset preserves broad longitudinal coverage within each latitude band, and the fixed-cluster construction is visually intuitive: tropical and dry stations concentrate in lower-latitude belts, temperate and continental stations occupy mid- to high-latitude regions, and polar stations appear at the highest latitudes.

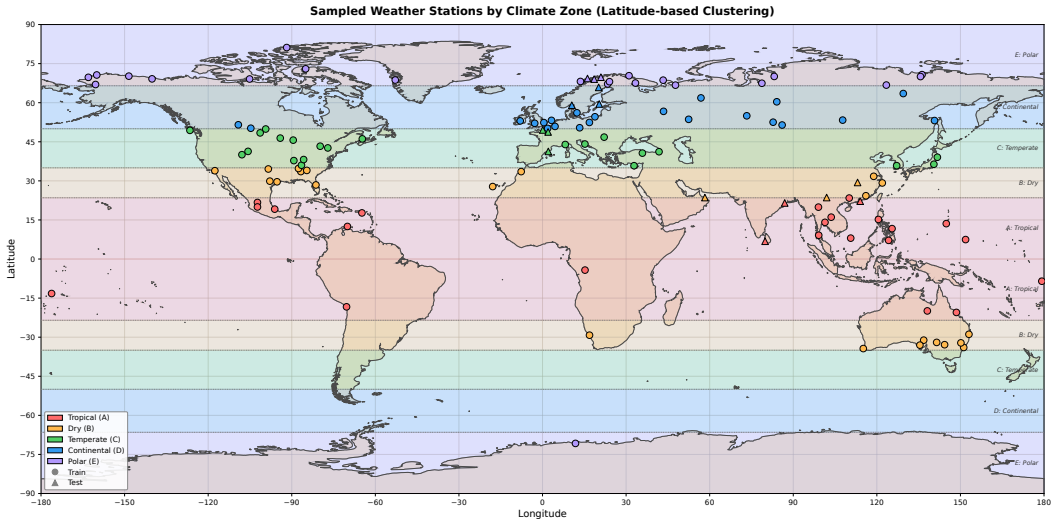

Figure 15: Geographic distribution of WEATHER-5K stations in the provided-cluster experiments. Colors identify the five latitude-based zones, and marker shapes distinguish the training/validation pool from test stations; the pool is subsequently split into 99 training and 11 validation entities.

D.2.3 LICENSE AND ACCESS

WEATHER-5K is distributed through its public GitHub repository at https://github.com/taohan10200/WEATHER-5K under the MIT License, with dataset resources, code, and license information openly accessible; no restricted-access training or data-use agreement is required. We use the benchmark under this license and report only aggregate results derived from the released data.

Table 14: Geographic zones defining the provided WEATHER-5K clusters.

Zone	Description	Absolute latitude range
Tropical	Low-latitude stations near the equatorial belt	$\lvert\text{lat}\rvert \leq 23.5^\circ$
Dry	Subtropical dry-zone stations	$23.5^\circ < \lvert\text{lat}\rvert \leq 35^\circ$
Temperate	Mid-latitude temperate stations	$35^\circ < \lvert\text{lat}\rvert \leq 50^\circ$
Continental	High-latitude continental stations	$50^\circ < \lvert\text{lat}\rvert \leq 66.5^\circ$
Polar	Polar and near-polar stations	$\lvert\text{lat}\rvert > 66.5^\circ$

Table 15: Balanced WEATHER-5K subset used in the provided-cluster experiments.

Zone	Training/validation pool	Test stations	Total
Tropical	22	3	25
Dry	22	3	25
Temperate	22	3	25
Continental	22	3	25
Polar	22	3	25
Total	110	15	125

D.3 eICU Details

D.3.1 Preprocessing pipeline

We construct the eICU benchmark in several stages, from raw ICU stays to dense patient-level multivariate time series. CLASP uses inferred responsibilities with $K = 4$. The four overlapping cardiovascular, pulmonary, neurologic, and gastrointestinal diagnosis indicators are metadata supplied to the model, not labels for these inferred clusters. The 12-hour context predicts the six vitals at horizons 6, 12, and 24 hours.

Hourly stay-level trajectories: We retain only stays of at least 24 hours. This cohort filter does not guarantee a complete 12-hour context plus a 24-hour target for every retained stay; forecast windows additionally require their full context and target segment within that same stay. Irregular clinical observations are aligned to an hourly grid from ICU admission to discharge: repeated periodic bedside vitals are averaged within each hour, while sporadic measurements, such as aperiodic blood pressures and laboratory tests, are represented by the most recent value associated with the hour. Values, missingness indicators, and recency information are all preserved at this stage.

Patient-level entities: All ICU stays of the same patient are concatenated in temporal order, yielding one entity per patient, possibly spanning multiple stays with boundaries retained explicitly. Window construction respects these boundaries: no training, validation, or test window crosses between stays, and its context and target are drawn from the same stay. Before diagnosis filtering this yields 104,897 patient-level entities, with an average of 1.27 ICU stays and 115.3 hours per entity.

Diagnosis-restricted cohort: We restrict to patients with at least one diagnosis in broad cardiovascular, pulmonary, neurologic, or gastrointestinal categories, producing 93,286 unique patients; the train, validation, and test splits contain 76,834, 8,529, and 7,923 entities, as reported in Table 17. The partition is defined by patient identity, so no patient appears in more than one split.

Feature densification and selection: Feature availability is estimated on the training split only; of 53 candidate variables, 24 sufficiently dense features are retained. Missing values are imputed by forward-filling within each stay and then filling remaining gaps with training-set means. From the dense representation we construct a value-only benchmark: among the 24 features, only six have a missingness rate below 20%: SaO_2, HR, RR, NIBP-S, NIBP-D, and NIBP-M. To avoid relying on forward-filled plateaus, we restrict the final modeling set to these six core vitals.

Table 16: Exact station IDs in the balanced WEATHER-5K subset, grouped by geographic zone, with the 99 training and 11 validation stations labeled separately. Within each zone, the test set comprises the three smallest station IDs, a deterministic holdout by sorted ID. Validation is stratified across zones: the three largest remaining IDs in the Tropical zone and the two largest remaining IDs in each other zone are reserved for validation.

Zone	Split	Station IDs
Tropical	Train	48302099999, 48404099999, 48430099999, 48551099999, 48919099999, 59254099999, 64450099999, 76571099999, 76662099999, 76692099999, 78551011624, 78982000411, 85406099999, 91218041414, 91334040505, 91643080705, 91753099999, 94255099999, 94365099999
	Validation	98327099999, 98553099999, 98746099999
	Test	42895099999, 43467099999, 45044099999
Dry	Train	58238099999, 58569099999, 59117099999, 60001099999, 60155099999, 68408099999, 72033353175, 72205012815, 72228013876, 72244012918, 72253912979, 72323513896, 72357503950, 74781263813, 94573099999, 94601099999, 94659099999, 94691099999, 94767099999, 95654099999
	Validation	95697099999, 95754099999
	Test	41256099999, 56966099999, 57584099999
Temperate	Train	14428099999, 15117099999, 16153099999, 17045099999, 17085099999, 17501099999, 47146099999, 47512099999, 47629099999, 71170099999, 71437099999, 71705099999, 71894099999, 72327313827, 72423093821, 72433693810, 72467494050, 72519454778, 72564524022, 72655594938
	Validation	72741504803, 72767594011
	Test	07028099999, 07145099999, 08181099999
Continental	Train	03522099999, 03577035046, 03966099999, 06168099999, 06252099999, 06451099999, 07005099999, 11438099999, 12135099999, 12330099999, 23812099999, 23966099999, 24656099999, 27453099999, 28698099999, 28806099999, 29937099999, 30637099999, 31369099999, 36058099999
	Validation	71129099999, 71569099999
	Test	01482099999, 02154099999, 02979099999
Polar	Train	01099099999, 01141099999, 02095099999, 02820099999, 04220099999, 20978099999, 21931099999, 21937099999, 22165099999, 22213099999, 22383099999, 23256099999, 24343099999, 70030027503, 70063727406, 70119600102, 70121026624, 71288099999, 71592099999, 71872099999
	Validation	71978099999, 89512099999
	Test	01010099999, 01023099999, 01046099999

Population scope and diagnostic subset. The minimum-stay and diagnosis filters above restrict the population represented by the 93,286-patient benchmark; it is not the full eICU database. The fixed-predictor sample-size analysis in Appendix E.4.3 makes a further, separate restriction: 32 long test stays are selected with seed zero from 503 eligible stays supporting eight initial-block and two later-block non-overlapping windows. This is a diagnostic cohort, distinct from the 7,923-patient forecasting test population. Its concentration calculations do not establish a full-test-population risk certificate.

D.3.2 Diagnosis overlap and split statistics

Figure 16 shows the overlap among the four diagnosis groups used to construct the cohort. The single-category regions remain large, but the pairwise, three-way, and four-way intersections are substantial: many patients exhibit multiple clinically relevant conditions at once, so the categories are overlapping descriptors rather than a mutually exclusive partition. Using them as provided clusters would discard this cross-category structure, which is why eICU serves as the inferred-cluster benchmark: diagnosis indicators enter only as metadata m_e, while the cluster structure itself is inferred by the router.

D.3.3 License and use agreement

Access to the eICU Collaborative Research Database at https://physionet.org/content/eicu-crd/2.0/ is governed by the PhysioNet Credentialed Health Data License and Use Agreement: access is restricted to approved users who complete the required human-subjects and HIPAA training, the data may be used only for lawful scientific research, re-identification attempts are prohibited, and appropriate safeguards are required. We use eICU only in de-identified

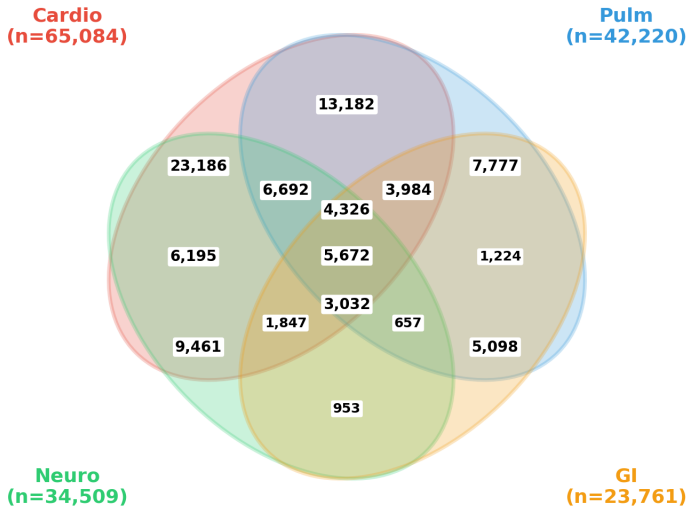

Figure 16: Patient-level overlap of the cardiovascular, pulmonary, neurologic, and gastrointestinal diagnosis categories used to define the eICU subset.

Table 17: Patient counts in each eICU split after preprocessing and diagnosis-based subset construction.

Split	Unique patient entities
Train	76,834
Validation	8,529
Test	7,923
Total	93,286

form, release no raw restricted records, and report only aggregate results and derived benchmark statistics; the experiment code is released in accordance with the agreement while respecting the access restrictions on the underlying credentialed data.

D.4 Data Use and Broader Impacts

This work may have positive impact in forecasting domains such as environmental monitoring and healthcare by improving prediction across related entities. However, in high-stakes settings, inaccurate forecasts or poorly calibrated predictive uncertainty may lead users to place excessive confidence in model outputs: in clinical decision support, an overconfident forecast could incorrectly suggest that a patient's vitals are stable, and in environmental monitoring it could underestimate the severity of an upcoming event, affecting downstream planning or resource allocation if the model is used without sufficient validation and human oversight. In addition, the learned shared latent structure may reflect biases, imbalances, or missingness patterns in the training data, which is particularly relevant for decision support on heterogeneous populations. The method should therefore be carefully validated, evaluated across subgroups, and used with human oversight in high-stakes applications.

E Additional Experimental Results

The results below address three empirical questions: whether CLASP improves forecasting for unseen entities, how its latent mechanism and optional readout contribute, and how its usefulness varies across forecasting backbones and population structures. Appendices E.1, E.2, and E.3 organize the evidence in that order. Appendix E.4 then relates the measured latent divergences to the theoretical analysis and keeps the conditional fixed-predictor bounds and reference diagnostics distinct from forecasting comparisons. All analyses retain their stated dataset, backbone, clustering, readout, and sampling configurations.

Reading the protocols. Table 8 specifies the latent law, forecast count S, sampler, and readout for each CLASP backbone; Table 13 specializes these choices to the external-model retrofits. Native-baseline sampling is given in Table 11; native models have no CLASP assignment, prototype count, or latent source. The linked forecasting tables use P_e^{agg} at deployment unless a row specifies a prior mean, posterior samples, or a control that removes or replaces the latent. Raw point forecasts are arithmetic sample means before readout, decoded points are means after the fixed validation-fitted blend, and full CLASP includes the reported latent conditioning and readout configuration (Appendix D.1.1). Distributional metrics instead use raw, unblended draws. Latent/reference-divergence tables do not evaluate forecasts, so S and forecast readout are inapplicable. Fixed-predictor calculations use their separately stated trajectory count M and loss, rather than the point-forecast protocol. Missing details of retained auxiliary runs are identified locally and are not supplied by these protocol references.

Forecasting evidence and latent diagnostics. The forecasting comparisons in Appendices E.1 and E.3 corroborate the main-text performance patterns within their reported protocols. Repeated table entries expand the reporting of those results and are not independent replications. Appendix E.2 probes attribution through controlled forecasting comparisons. Prototype moment residuals and aggregate KL values diagnose learned latent structure; they are not forecasting-risk guarantees. The separate concentration calculations in Appendix E.4 concern only the explicitly identified fixed predictors, risks, and sampling assumptions.

E.1 Unseen-Entity Forecasting Quality and Variability

The per-channel and per-vital results show where the overall forecasting gains occur. Raw-draw distributional metrics assess a separate aspect of forecast quality, while training-seed and held-out-station summaries characterize different sources of variability. These analyses complement the unseen-entity comparisons in Section 4.1.

E.1.1 Per-channel and per-vital results

Tables 18 and 19 decompose the overall results in Tables 1 and 2 by channel. The provided-cluster Overall values in Table 18 repeat Table 1; its inferred-cluster CLASP values repeat Table 2. CLASP channel errors use the same $S{=}100$ decoded draws; NsDiff and TSFlow channel errors use a second $S{=}100$ sampling pass over the same windows. Their channel means therefore differ from the Overall column by sampling noise in the third decimal.

In Table 18, inferred versus provided CLASP changes overall MAE/MSE by $-0.9\%/-1.2\%$ at $H = 24$, $-0.4\%/-1.3\%$ at $H = 72$, and $+0.3\%/-0.9\%$ at $H = 168$. The provided-cluster error is the denominator; negative values indicate smaller errors.

E.1.2 Predictive Distribution Quality

Table 20 evaluates raw, unblended draws. Flow-matching CLASP has coverage closest to nominal 0.80 in five of six settings (TSFlow is closest on WEATHER-5K at $H = 72$), never undercovers by more than 0.04, and has the lowest CRPS and energy score throughout, followed by CLASP diffusion. These distributional results do not identify the mechanism behind the point-forecast gains. Empirical coverage measures observed targets inside the central 80% sample interval; that interval alone does not ensure nominal coverage. CRPS and energy score additionally assess calibration and sharpness of the full predictive distribution.

Table 18: Per-channel WEATHER-5K errors for the provided-zone ($K = 5$) comparison in Table 1 and inferred CLASP ($K = 4$) in Table 2. Entries are MAE | MSE; lower is better. Bold and underlining rank the Overall values within the provided-cluster comparison; the final block gives inferred-cluster CLASP. Both CLASP configurations use flow matching, aggregate-prior sampling with $S = 100$, and the full validation-fitted readout (Table 8).

Model	Horizon	TMP	DEW	WND_sin	WND_cos	WND_RATE	SLP	Overall
D3VAE	24	3.910 | 26.073	4.243 | 29.906	0.330 | 0.176	0.356 | 0.203	1.258 | 2.959	3.738 | 24.006	2.306 | 13.887
	72	5.140 | 45.380	6.224 | 65.677	0.468 | 0.340	0.490 | 0.365	1.836 | 6.327	5.549 | 56.210	3.285 | 29.050
	168	5.034 | 42.976	6.054 | 63.001	0.450 | 0.333	0.461 | 0.345	1.700 | 5.115	5.085 | 46.999	3.131 | 26.462
TSDiff	24	2.265 | 11.944	2.824 | 20.302	0.260 | 0.106	0.290 | 0.131	0.916 | 1.548	3.055 | 15.814	1.602 | 8.308
	72	3.087 | 22.372	3.899 | 36.402	0.330 | 0.173	0.324 | 0.168	1.048 | 2.203	5.236 | 43.130	2.321 | 17.408
	168	4.574 | 41.289	4.680 | 44.088	0.327 | 0.172	0.346 | 0.190	1.126 | 2.545	4.821 | 39.611	2.646 | 21.316
Timer-XL ZS	24	2.040 | 9.080	1.950 | 8.857	0.496 | 0.428	0.500 | 0.430	1.603 | 5.490	3.416 | 28.647	1.668 | 8.822
	72	2.658 | 14.904	2.717 | 16.075	0.573 | 0.537	0.574 | 0.537	1.878 | 7.426	5.546 | 72.212	2.324 | 18.615
	168	3.038 | 18.846	3.108 | 20.131	0.608 | 0.586	0.606 | 0.580	1.975 | 8.112	6.664 | 99.029	2.667 | 24.547
Timer-XL FT	24	2.102 | 9.132	1.929 | 8.281	0.504 | 0.416	0.506 | 0.410	1.620 | 5.469	3.190 | 22.709	1.642 | 7.736
	72	2.610 | 13.971	2.597 | 14.467	0.563 | 0.495	0.562 | 0.487	1.828 | 6.976	5.003 | 57.311	2.194 | 15.618
	168	2.994 | 18.073	3.039 | 19.137	0.598 | 0.546	0.593 | 0.532	1.928 | 7.714	6.317 | 89.001	2.578 | 22.501
NsDiff	24	1.841 | 7.413	1.967 | 8.375	0.483 | 0.379	0.483 | 0.369	1.522 | 4.876	2.572 | 16.720	1.478 | 6.355
	72	2.374 | 12.246	2.556 | 13.952	0.557 | 0.467	0.551 | 0.445	1.772 | 6.400	4.407 | 46.887	2.036 | 13.400
	168	2.789 | 16.385	3.064 | 18.979	0.590 | 0.501	0.582 | 0.481	1.843 | 7.052	5.739 | 74.157	2.435 | 19.591
TSFlow	24	1.701 | 7.056	1.787 | 7.931	0.563 | 1.344	0.566 | 1.334	1.579 | 5.608	2.367 | 15.115	1.427 | 6.398
	72	2.535 | 12.370	2.685 | 13.628	0.645 | 1.416	0.651 | 1.414	1.835 | 6.290	4.689 | 48.356	2.173 | 13.912
	168	2.778 | 16.127	2.945 | 17.852	0.677 | 1.453	0.671 | 1.434	1.908 | 7.730	5.578 | 69.094	2.426 | 18.948
CLASP provided	24	1.584 | 6.185	1.667 | 6.916	0.472 | 0.366	0.461 | 0.347	1.491 | 4.739	2.070 | 12.112	**1.291** | **5.111**
	72	2.197 | 10.805	2.338 | 12.296	0.535 | 0.435	0.524 | 0.414	1.714 | 6.121	4.106 | 41.020	**1.902** | **11.848**
	168	2.634 | 14.689	2.804 | 16.613	0.574 | 0.474	0.562 | 0.456	1.831 | 6.944	5.336 | 63.981	**2.290** | **17.193**
CLASP inferred, K=4	24	1.578 | 6.152	1.660 | 6.795	0.464 | 0.379	0.458 | 0.367	1.487 | 4.639	2.033 | 11.962	1.280 | 5.049
	72	2.170 | 10.676	2.315 | 12.129	0.534 | 0.455	0.516 | 0.433	1.694 | 6.056	4.135 | 40.397	1.894 | 11.691
	168	2.641 | 14.458	2.843 | 16.440	0.565 | 0.492	0.554 | 0.470	1.816 | 6.867	5.369 | 63.513	2.298 | 17.040

All methods under-cover at longer horizons. The Huber velocity and auxiliary ℓ_1 objective in equation 59 does not directly optimize calibration; velocity error and finite-step Euler integration may affect quantiles, but their contributions are not isolated. The raw-draw coverage cannot be attributed to the affine readout in equation 12 or Table 23's readout variants. Ideal flow matching recovers the target distribution with the correct velocity field and exact integration (Lipman et al., 2023); this does not guarantee calibration for the fitted model and numerical solver.

E.1.3 Variability Across Training Runs and Held-Out Entities

Table 21 reports the five training runs underlying Table 1; Table 22 summarizes their 15 held-out stations. Training-seed variation and between-station variation describe different sources of uncertainty. The latter bears on unseen-station generalization, but neither summary alone establishes population-level uncertainty in differences between methods.

Table 19: Per-vital eICU errors corresponding to Table 2; CLASP uses DiT diffusion, inferred responsibilities ($K = 4$), and the aggregate-prior, $S = 100$, full-readout protocol in Table 8. Entries are MAE | MSE (lower is better); bold and underlining mark the best and second-best Overall values. The CLASP entries are unweighted macro-averages over the four inferred clusters of Table 33 (equal cluster weighting); the balanced evaluation assigns equal window denominators per cluster, so this macro-average coincides with pooled patient/window performance.

Model	Horizon	SaO_2	HR	RR	NIBP-S	NIBP-D	NIBP-M	Overall
D3VAE	6	1.581 | 6.737	8.454 | 136.948	12.389 | 300.470	11.430 | 256.020	7.973 | 126.759	8.549 | 144.520	8.396 | 161.909
	12	2.182 | 5.863	10.038 | 108.572	22.447 | 532.735	15.681 | 260.798	10.721 | 124.664	12.107 | 157.066	12.196 | 198.283
	24	1.879 | 6.053	10.367 | 152.070	20.489 | 597.123	14.873 | 306.898	9.568 | 130.419	8.140 | 162.177	10.886 | 225.790
TSDiff	6	1.842 | 9.150	8.501 | 134.727	3.308 | 24.297	14.316 | 356.588	10.242 | 202.681	10.503 | 201.927	8.119 | 154.895
	12	1.952 | 10.124	9.155 | 157.064	3.872 | 33.734	14.564 | 369.607	9.717 | 182.670	10.257 | 193.441	8.253 | 157.774
	24	2.213 | 12.980	10.576 | 202.248	3.643 | 29.400	16.677 | 474.191	11.263 | 237.846	11.833 | 249.658	9.367 | 201.054
Timer-XL ZS	6	1.488 | 6.790	6.326 | 80.743	2.337 | 13.498	11.768 | 252.968	8.247 | 140.815	8.482 | 140.862	6.441 | 105.946
	12	1.707 | 7.723	7.939 | 118.527	2.792 | 18.431	13.129 | 302.029	8.915 | 155.681	9.280 | 160.201	7.294 | 127.099
	24	1.940 | 8.660	9.951 | 173.448	3.367 | 25.631	14.831 | 370.672	9.818 | 178.466	10.336 | 189.003	8.374 | 157.647
Timer-XL FT	6	1.692 | 9.508	7.780 | 115.239	2.738 | 17.301	12.466 | 278.894	8.639 | 151.094	8.847 | 151.577	7.027 | 120.602
	12	1.839 | 10.474	8.690 | 141.842	3.027 | 20.821	13.342 | 314.049	9.043 | 161.725	9.371 | 165.879	7.552 | 135.798
	24	1.972 | 11.027	9.714 | 174.664	3.356 | 25.464	14.357 | 358.026	9.478 | 173.859	9.930 | 182.366	8.135 | 154.234
NsDiff	6	1.411 | 6.867	5.922 | 74.239	6.181 | 76.665	10.984 | 224.055	7.688 | 124.603	7.850 | 123.917	6.673 | 105.058
	12	1.562 | 7.822	6.972 | 100.198	6.293 | 76.251	11.823 | 255.692	8.064 | 134.996	8.429 | 137.870	7.191 | 118.805
	24	1.671 | 8.202	8.068 | 129.829	5.570 | 63.003	12.741 | 291.379	8.476 | 146.322	8.927 | 151.628	7.576 | 131.727
CLASP	6	1.366 | 7.458	5.635 | 69.753	2.202 | 12.358	10.621 | 215.788	7.370 | 121.903	7.613 | 120.486	**5.801** | **91.291**
	12	1.491 | 7.772	6.682 | 93.477	2.493 | 15.560	11.502 | 246.582	7.790 | 131.520	8.140 | 133.268	**6.350** | **104.696**
	24	1.604 | 8.068	7.753 | 120.526	2.813 | 19.607	12.471 | 283.410	8.258 | 143.077	8.719 | 148.624	**6.936** | **120.552**

Table 20: Probabilistic evaluation of the generative models of Tables 1 and 2: empirical P10–P90 coverage (nominal 0.80; value closest to nominal in **bold**), continuous ranked probability score (CRPS, lower better), and energy score (lower better). Columns follow $H = 24, 72, 168$ on WEATHER-5K and $H = 6, 12, 24$ on eICU. Bold marks the best value within each metric, dataset, and horizon. All metrics use raw, unblended draws. CLASP uses provided WEATHER-5K zones ($K = 5$) and inferred eICU responsibilities ($K = 4$). CLASP backbone rows use $S = 100$ aggregate-prior draws with the corresponding sampler in Table 8; baseline budgets appear in Table 11.

Dataset	Model	Per horizon		
Coverage (nominal 0.80)				
WEATHER-5K	NsDiff	0.787	0.745	0.714
	TSFlow	0.749	**0.793**	0.741
	CLASP diffusion	0.772	0.758	0.739
	CLASP flow matching	**0.796**	0.789	**0.776**
eICU	NsDiff	0.833	0.796	0.780
	CLASP diffusion	0.781	0.772	0.765
	CLASP flow matching	**0.804**	**0.798**	**0.784**
CRPS (lower better)				
WEATHER-5K	NsDiff	1.085	1.583	1.912
	TSFlow	1.052	1.547	1.881
	CLASP diffusion	1.023	1.498	1.832
	CLASP flow matching	**0.981**	**1.442**	**1.761**
eICU	NsDiff	4.823	5.412	5.891
	CLASP diffusion	4.412	4.987	5.463
	CLASP flow matching	**4.198**	**4.753**	**5.201**
Energy score (lower better)				
WEATHER-5K	NsDiff	1.842	2.713	3.295
	TSFlow	1.793	2.651	3.218
	CLASP diffusion	1.738	2.581	3.142
	CLASP flow matching	**1.674**	**2.493**	**3.038**
eICU	NsDiff	38.451	43.218	47.583
	CLASP diffusion	35.012	39.478	43.521
	CLASP flow matching	**33.481**	**37.692**	**41.547**

Table 21: WEATHER-5K flow-matching CLASP with provided zones ($K = 5$) and the aggregate-prior, $S = 100$, full-readout protocol of Table 8: MAE (top) and MSE (bottom) from five independently trained checkpoints, not repeated evaluations of one checkpoint. Mean and Std are across-run summaries; Mean reproduces Table 1. Per-seed values and standard deviations use four decimals. Both metrics have standard deviation below 0.001; this observed spread does not identify its cause.

	Per-seed						
Horizon	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Mean	Std
MAE							
24	1.2906	1.2913	1.2908	1.2915	1.2908	1.291	0.0004
72	1.9021	1.9030	1.9020	1.9025	1.9024	1.902	0.0004
168	2.2897	2.2904	2.2905	2.2901	2.2903	2.290	0.0003
MSE							
24	5.1101	5.1113	5.1106	5.1116	5.1104	5.111	0.0006
72	11.8488	11.8479	11.8477	11.8486	11.8480	11.848	0.0005
168	17.1934	17.1928	17.1927	17.1935	17.1926	17.193	0.0004

Table 22: WEATHER-5K flow-matching CLASP with provided zones ($K = 5$) and the aggregate-prior, $S = 100$, full-readout protocol of Table 8: per-station MAE mean, standard deviation, and min–max range across 15 held-out stations, plus the range of unweighted per-zone means (three stations each). The station mean and zone range agree at reported precision with Tables 1 and 31. Agreement is not automatic: zone means here weight stations equally, whereas cluster Overall errors pool zone windows. The station spread exceeds Table 21's seed spread; these are descriptive variability summaries, not confidence intervals or paired method differences.

Horizon	Station mean MAE	Station std MAE	Station range (min–max)	Per-zone mean range
24	1.291	0.320	0.724–1.734	0.814–1.594
72	1.902	0.568	0.944–2.615	1.074–2.477
168	2.290	0.701	1.171–3.176	1.291–3.046

E.2 Contributions of Latent Partial Pooling and Forecast Readout

The controls below address six attribution questions and include complementary conditioning controls and prototype-moment diagnostics:

1. What happens without pooling ($\lambda = 0$)?
2. What happens without decoding (the forecast readout)?
3. What changes between deterministic and probabilistic prototypes?
4. What changes from $K = 1$ to multiple prototypes? Appendix E.3.3 gives the broader K sweep.
5. How does the inference latent source affect forecasts?
6. How sensitive are forecasts to the alignment weight λ?

These comparisons expand Table 3; reused results do not provide independent confirmation. The WEATHER-5K plain-backbone and CLASP rows, each with and without the validation-fitted readout, provide the matched 2×2 comparison in Table 3. The separate zero-weight control jointly removes alignment and prototype-prior learning, leaving the deployment prior untrained.

Evaluation protocol. WEATHER-5K experiments use the flow-matching backbone and the stride-24 test windows of Table 1. The component ablations, latent-source comparison, readout study, alignment-weight sweep, and the zero-weight and full-CLASP variants in the comparison with the plain backbone below use inferred clusters with $K = 4$ and router-loss weight $\gamma = 1$ unless the variant explicitly changes clustering or label availability. All share the full-CLASP reference from Table 2 and follow Appendix D.1. On eICU, MAE and MSE are averaged over the six vitals in original units, and comparisons are restricted to the corresponding protocol block. The cluster-label availability ablation applies only to WEATHER-5K, which has provided geographic-zone labels; eICU uses router-inferred assignments, with four overlapping diagnosis indicators serving as metadata rather than cluster labels. The aggregate-KL diagnostic in Appendix E.4.1 uses the separate frozen-$K = 5$ checkpoint and window protocol specified there.

E.2.1 Component ablations across datasets

Tables 23 and 24 group controls by the component changed: prototype representation and learning, cluster structure, conditioning pathways, or forecast readout. The structural controls change the model used for training; the readout controls change how a fitted model's forecasts are summarized. The WEATHER-5K experiments use flow matching and the eICU experiments use diffusion. The full reference uses inferred responsibilities over $K = 4$ prototypes, with the dataset splits, horizons, and evaluation procedures specified in Appendix D.1. Except for the readout controls, forecasts use the complete validation-fitted readout. The controls below retain the backbone family and unaffected conditioning pathways; removing modules or variance parameters need not preserve the exact parameter count.

Without pooling: zero alignment weight ($\lambda = 0$). The pooling coefficient is zero throughout training, while the forecasting loss, router regularization, context encoder, and prototype embeddings remain active. Training continues to sample from the aggregate posterior, and inference continues to sample from the aggregate deployment prior. Thus z_e is retained. Because the prototype means and covariance parameters receive data-dependent gradients only through the pooling term, this control removes both posterior–prototype alignment and learning of the inference prior. The uncapped pooling objective is multiplied by zero on both datasets. The row is therefore labeled "prior untrained"; its difference from full CLASP measures the joint contribution of alignment and prior learning.

Without decoding: raw forecasts and readout controls. These rows retain the fitted model and deployment latent rule. "Raw CLASP (no readout)" disables the anchor blend. The WEATHER-5K controls in Tables 3 and 23 and the eICU control in Table 24 use the arithmetic mean of all raw draws, keeping point aggregation fixed relative to decoded forecasts (Appendix D.1.1). "Decoded CLASP (persistence)" enables only the persistence anchor, which repeats the last observed

value, and fixes the climatology coefficient to zero. The full-CLASP reference uses both persistence and climatology/context-mean anchors. Blend profiles are fitted on validation entities and fixed for testing; decoded point forecasts average all blended draws. The dataset-specific second anchor is defined in Appendix D.1.1. No decoded samples are trimmed, and P10–P90 intervals are computed separately from the raw, unblended draws. The plain-backbone readout uses a global profile, whereas CLASP uses prototype-specific profiles (Appendix D.1.1); the readout-enabled model contrast therefore also includes profile specialization.

Deterministic versus probabilistic prototypes. This control replaces the Gaussian component posteriors and prototype priors with deterministic entity vectors $h_{e,k}$ and prototype vectors μ_k, respectively. It replaces the responsibility-weighted KL by

$$\mathcal{L}^{\mathrm{det}}_{\mathrm{pool},e} = \sum_k r_{ek} \|h_{e,k} - \mu_k\|_2^2.$$

The deterministic entity and prototype vectors are learned jointly with the backbone. Their responsibility-weighted sums supply the conditioning latent during training and deployment, respectively, following the same entity-to-prototype replacement as the probabilistic model. The context encoder, router, prototype embeddings, latent dimension, and forecasting backbone are retained. Latent Gaussian sampling is removed, while the backbone's flow or diffusion sampling remains stochastic. This changes both representation and alignment loss; it is distinct from taking the prior mean only at inference in Table 6 and does not isolate covariance modeling alone.

Single versus multiple prototypes ($K = 1$). Every entity has $r_{e1} = 1$, one context-dependent posterior $q_{e,1}$, one shared prototype prior P_1, and prototype embedding u_1. The Gaussian latent and its alignment penalty remain, but there is no choice among clusters and the categorical router KL is zero. This preserves population-wide latent partial pooling while removing cluster specialization.

Direct label-vector conditioning. The one-hot zone and diagnosis-indicator controls replace CLASP's prototype latent mechanism with a deterministic label vector supplied to the forecasting backbone alongside the history representation. They omit the learned responsibility router, prototype-conditioned Gaussian posteriors, prototype priors, and their alignment and routing losses. The history encoder, forecasting backbone, and forecast-readout protocol are retained. On WEATHER-5K, "one-hot zone conditioning" uses the five-dimensional vector $\mathrm{onehot}(c_e)$ for the provided geographic zone. It differs from provided-cluster CLASP, which retains the Gaussian latent mechanism and uses the zone label to select a prototype. On eICU, "diagnosis-indicator conditioning" uses the four binary diagnosis indicators directly. Multiple entries may equal one; the indicators are neither converted into a mutually exclusive cluster label nor normalized into prototype responsibilities. These controls compare direct access to population descriptors with learned latent partial pooling.

Partial cluster-label availability (50% labels). This WEATHER-5K control retains CLASP and uses $K = 5$, corresponding to the five provided geographic zones. Labels are hidden for approximately half of the training stations, with the same mask applied to every window from a station. Labeled training stations use $r_e = \mathrm{onehot}(c_e)$; unlabeled training stations use $r_e = \pi_\omega(h_e, m_e)$. Both groups contribute to forecasting and posterior–prototype alignment. The context inputs and coordinate metadata remain available. Validation and test labels are retained and supply one-hot assignments, without fitting to test entities. Thus label masking changes training supervision rather than deployment label availability. This row also differs in prototype count from the inferred-$K = 4$ reference, so their difference does not isolate label availability alone.

Without latent conditioning (z_e). The Gaussian latent branch and its pooling penalty are removed, and the backbone receives $g_e = [h_e, u_e]$. The context encoder, responsibility router, and prototype embeddings still provide history-dependent and cluster-dependent conditioning. This control removes the sampled latent pathway itself, unlike $\lambda = 0$, which retains that pathway and its train–deployment replacement.

Without encoder-state conditioning (M_e). This control removes the backbone's cross-attention access to the per-step encoder states M_e. The context encoder remains active: its summary h_e still

supports routing, posterior construction, and the conditioning vector $g_e = [h_e, u_e, z_e]$. Prepended raw-history tokens are retained. The control therefore removes access to the learned time-resolved states while preserving the other history and latent conditioning routes.

Without metadata (m_e). Static metadata inputs are removed from the context encoder, router, and component-posterior encoder. These inputs are longitude and latitude on WEATHER-5K and the four overlapping diagnosis indicators on eICU. History values and the configured missingness and recency channels remain available. Responsibilities are still inferred, now without static metadata; this tests the additional information supplied by metadata rather than removing the router or the observed history.

WEATHER-5K. Table 23 consolidates the main-text controls and additional component results with inferred clusters ($K = 4$ by default). Inferred CLASP is the reference for the component and readout controls; the provided-label CLASP row reproduces the $K = 5$ model from Table 1. The matched readout-only control (plain backbone with and without the validation-fitted readout) brackets the $\lambda = 0$ control at longer horizons: injecting an untrained latent ($\lambda = 0$) is worse than the plain backbone with readout at $H = 24$ and slightly worse at $H = 72, 168$, consistent with the contribution of alignment and prototype-prior learning. At $H = 72$, the MAE difference is only 0.001. The $\lambda = 0$ control has the largest $H = 24$ errors. The controls without latent conditioning, encoder states, or metadata have higher errors than full CLASP at every horizon; encoder-state removal has the smallest gap among these three conditioning-pathway ablations. Direct one-hot zone conditioning has higher errors than the inferred-cluster reference. The single-prototype control retains global pooling and tests cluster specialization; $K = 4$ improves both metrics at every horizon. The 50%-label variant ($K = 5$) has higher errors than the inferred-$K = 4$ reference, a comparison that changes both training-label availability and prototype count. Decoding improves both metrics over raw draws at every horizon, with the largest MAE gain at $H = 168$. Both full-CLASP rows use the complete two-anchor readout.

eICU. Within the matched protocol blocks of Table 24, full CLASP reduces MAE by 21.2–21.8% and MSE by 26.9–28.8% relative to $\lambda = 0$, supporting the joint contribution of alignment and prototype-prior learning. Relative to deterministic prototypes, reductions are 2.4–2.5% in MAE and 4.9–5.4% in MSE; this comparison changes both representation and alignment loss. Full CLASP also improves over the cluster-structure and conditioning controls, with encoder-state removal causing the largest deterioration among conditioning variants. Readout gains depend on horizon and metric: persistence blending gives the lowest MAE and MSE at $H = 6$, while full readout minimizes MAE at $H = 12, 24$ without minimizing MSE throughout. The top reference applies to the three structural-control blocks and is repeated in the readout block with that block's minima highlighted.

Comparison with the plain backbone. The matched WEATHER-5K comparison in Table 3, reproduced in Table 23, evaluates the plain backbone and CLASP both with and without the validation-fitted readout. Enabling the readout on the plain backbone reduces MAE by 1.1–3.1% and MSE by 1.8–3.1% across the three horizons. With the readout enabled for both models, full CLASP reduces MAE by 12.5–17.7% and MSE by 16.4–24.4% relative to the plain backbone. The separate $\lambda = 0$ control retains latent conditioning and the readout while leaving the deployment prior untrained; full CLASP reduces its MAE by 13.6–21.4%. These reused results distinguish model and readout effects within this setting rather than providing independent evidence or a unique additive attribution. The zero-weight contrast jointly evaluates alignment and prior learning.

E.2.2 Which latent source is used at inference?

This subsection specifies the controls reported in Table 6 of Section 4.5. They use held-out WEATHER-5K stations, the flow-matching backbone, inferred clusters ($K = 4$), and the stride-24 test-window protocol of Appendix D.1. The component-posterior parameters $(\mu_{e,k}, \Sigma_{e,k}) = g_\psi(h_e, u_k, m_e)$ use only observed history, prototype embeddings, and metadata, so posterior-based inference requires no future targets. Within each fitted model and horizon, the encoder, router, prototype parameters, conditioning pathways, forecasting backbone, and validation-fitted readout remain

Table 23: Complete WEATHER-5K ablations on the flow-matching backbone, including the controls from Table 3. CLASP variants use inferred clusters ($K = 4$) unless stated otherwise; direct zone conditioning omits the prototype latent mechanism, and the 50%-label control uses $K = 5$ with labels hidden only during training. Lower is better; bold marks column minima. Raw and decoded point forecasts both use arithmetic means (Appendix D.1.1). Full CLASP includes both persistence and climatology anchors. Sampling uses $S = 100$ and 50 Euler steps (Table 8); row labels identify changes to the latent or readout.

	$H = 24$		$H = 72$		$H = 168$	
Variant	MAE	MSE	MAE	MSE	MAE	MSE
CLASP (provided labels, $K = 5$)	1.291	5.111	1.902	11.848	**2.290**	17.193
CLASP (inferred, $K = 4$)	**1.280**	**5.049**	**1.894**	**11.691**	2.298	**17.040**
Matched readout-only control						
Plain backbone (raw)	1.555	6.801	2.350	15.275	2.710	21.035
Plain backbone (validation-fitted readout)	1.538	6.678	2.301	14.835	2.626	20.383
Prototype representation and learning						
$\lambda = 0$ (prior untrained)	1.629	7.269	2.302	15.266	2.659	20.636
Deterministic prototypes	1.318	5.342	1.931	12.084	2.324	17.488
Cluster structure						
Single prototype ($K = 1$)	1.347	5.606	1.908	11.732	2.329	17.341
One-hot zone conditioning	1.355	5.702	1.931	12.184	2.345	17.514
50% training labels ($K = 5$)	1.318	5.302	1.905	11.902	2.318	17.302
Conditioning pathways						
Without latent z_e	1.450	6.142	2.079	13.309	2.499	19.184
Without encoder states M_e	1.358	5.742	1.974	12.461	2.361	17.812
Without metadata m_e	1.398	5.951	2.037	13.344	2.452	19.205
Readout variants						
Raw CLASP (no readout)	1.294	5.181	1.935	11.899	2.371	17.536
Decoded CLASP (persistence)	1.293	5.114	1.922	11.882	2.338	17.416

fixed; each window retains the same responsibilities. The three measured variants are

$$\text{Posterior samples:} \quad z_e \sim q_e^{\text{agg}},$$
$$\text{Prior mean:} \quad z_e = \mu_e^P = \sum_k r_{ek}\mu_k,$$
$$\text{Prior samples:} \quad z_e \sim P_e^{\text{agg}}.$$

The sampled variants use the independent component aggregation of equation 6 and equation 7. All variants retain $S = 100$ trajectories with 50 Euler steps and the complete two-anchor readout. Fixing the latent to the prior mean removes latent sampling, while the flow-matching backbone retains its generative noise. The readout averages all blended trajectories without trimming (Appendix D.1.1).

The improvements in Table 6 support the partial-pooling interpretation of replacing entity-specific posterior deviations with shared prototype structure. This replacement changes both latent means and covariances; it need not reduce covariance and is not a scalar variance-shrinkage operation. The prior-sample advantage over the prior mean suggests that latent variability can remain useful under the fixed readout. These observations neither identify posterior miscalibration nor isolate shrinkage as the causal mechanism. No posterior-mean result is available, so the experiment does not provide the complete mean-versus-sample comparison for both latent sources.

E.2.3 Sensitivity to the alignment weight λ

Table 25 varies λ in the mean pooling penalty of equation 4 for held-out WEATHER-5K forecasting with the flow-matching backbone and inferred clusters ($K = 4$). The context encoder, metadata, latent dimensions, router, conditioning pathways, backbone, readout, and all other hyperparameters remain fixed. As in the zero-weight control above, the prototype prior parameters (μ_k, Σ_k), distinct from u_k, receive their training signal only through $\lambda\mathcal{L}_{\text{pool}}$. Thus the sweep jointly varies alignment and prior learning; its component penalty is distinct from the reference complexity $\mathcal{C}_e^{\text{ref}}(Q_e, r_e)$.

Table 24: eICU component ablations with the diffusion backbone and inferred clusters ($K = 4$ unless varied). Diagnosis-indicator conditioning uses direct label vectors in place of the prototype latent mechanism. MAE and MSE are averaged over six vitals in original units. The top CLASP row is the reference for the prototype, cluster-structure, and conditioning blocks; it is repeated in the readout block. Compare each control with its block's reference; bold marks the minima within each such comparison block, including ties. Raw and decoded point forecasts both use arithmetic means (Appendix D.1.1). Full CLASP includes both persistence and context-mean anchors. Sampling uses $S = 100$ and 50 DDIM steps (Table 8); row labels identify changes to the latent or readout.

	$H = 6$		$H = 12$		$H = 24$	
Variant	MAE	MSE	MAE	MSE	MAE	MSE
CLASP	**5.801**	**91.291**	**6.350**	**104.696**	**6.936**	**120.552**
Prototype representation and learning						
$\lambda = 0$ (prior untrained)	7.405	128.140	8.118	143.270	8.802	165.310
Deterministic prototypes	5.952	96.205	6.507	110.148	7.104	127.402
Cluster structure						
Single prototype ($K = 1$)	5.850	96.805	6.500	113.214	7.330	126.821
Diagnosis-indicator conditioning	6.081	98.452	6.581	110.346	7.058	126.733
Conditioning pathways						
Without latent z_e	6.462	105.510	7.034	113.265	7.472	138.413
Without encoder states M_e	7.390	127.190	7.450	136.430	8.217	143.140
Without metadata m_e	6.450	103.974	6.741	121.592	7.167	134.292
Readout variants						
Full CLASP	5.801	91.291	**6.350**	104.696	**6.936**	120.552
Raw CLASP (no readout)	5.781	95.314	6.510	103.986	7.323	**120.283**
Decoded CLASP (persistence)	**5.753**	**91.013**	6.547	**102.854**	7.256	125.254

The deployed $\lambda = 1$ was selected on validation entities; this held-out sweep provides supplementary sensitivity evidence. In this sweep, $\lambda = 1$ minimizes reported MSE at all horizons and MAE at $H = 24, 72$; $\lambda = 0.5$ has lower MAE at $H = 168$ (2.294 versus 2.298). At $H = 24$, $\lambda = 1$ lowers MAE by 21.4% relative to $\lambda = 0$, while $\lambda = 5$ raises it by 2.7% relative to $\lambda = 1$. These results favor an intermediate tested weight for joint alignment and prior learning, but do not isolate their contributions or establish a tighter certificate. The $\lambda = 0, 1$ rows reuse Table 3.

Table 25: Prototype-alignment weight sweep on WEATHER-5K with inferred clusters ($K = 4$), flow matching, aggregate-prior sampling ($S = 100$, 50 Euler steps), and the full readout (Table 8). Architecture and other hyperparameters are fixed, but λ jointly scales posterior alignment and the training signal for the prototype prior. Bold marks column minima.

	$H = 24$		$H = 72$		$H = 168$	
λ	MAE	MSE	MAE	MSE	MAE	MSE
0	1.629	7.269	2.302	15.266	2.659	20.636
0.1	1.322	5.388	1.928	12.015	2.312	17.402
0.5	1.301	5.184	1.908	11.879	**2.294**	17.238
1	**1.280**	**5.049**	**1.894**	**11.691**	2.298	**17.040**
2	1.296	5.149	1.907	11.881	2.296	17.241
5	1.314	5.267	1.923	12.094	2.318	17.502

E.2.4 Diagnostic: Prototype Moment Alignment

Proposition 2 characterizes the conditional minimizer of the raw, uncapped component pooling objective. The logged training diagnostics compare each prototype with responsibility-weighted moments of its own component posteriors on the current minibatch, rather than with moments of the aggregate q_e^{agg} over the full entity population. These residuals diagnose empirical moment alignment and do not bound forecasting risk or attribute forecasting gains to a unique mechanism.

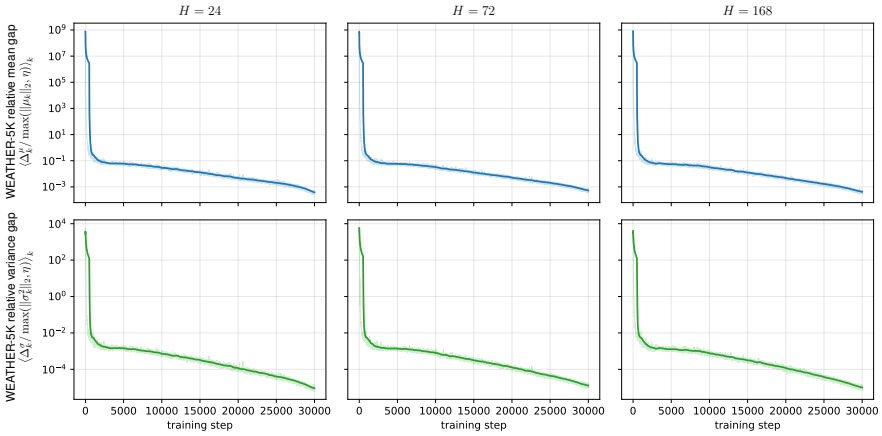

Figure 17: Current-minibatch residuals during WEATHER-5K training with provided zones ($K = 5$): mean G_μ in the top panels and variance G_σ in the bottom panels.

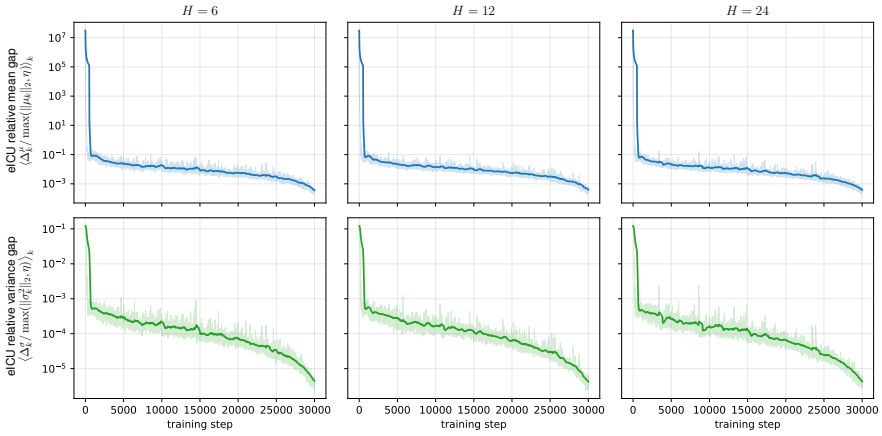

Figure 18: Current-minibatch residuals during eICU training with inferred responsibilities ($K = 4$): mean G_μ in the top panels and variance G_σ in the bottom panels.

At optimizer iteration t, let $\mathcal{B}_t$ be the current minibatch, $N^{\text{eff}}_{k,\mathcal{B}}(t) = \sum_{e \in \mathcal{B}_t} r_{ek}(t)$, and $\nu_k(t) = \max\{N^{\text{eff}}_{k,\mathcal{B}}(t), \eta\}$, where $\eta = 10^{-8}$. Define the diagonal variance vectors $\sigma^2_{e,k}(t) = \operatorname{diag} \Sigma_{e,k}(t)$ and $\sigma^2_k(t) = \operatorname{diag} \Sigma_k(t)$. The moments used in the logged residuals are

$$\tilde{\mu}_{k,\mathcal{B}}(t) = \frac{1}{\nu_k(t)} \sum_{e \in \mathcal{B}_t} r_{ek}(t) \mu_{e,k}(t),$$

$$V_{k,\mathcal{B}}(t) = \frac{1}{\nu_k(t)} \sum_{e \in \mathcal{B}_t} r_{ek}(t) \left[\sigma^2_{e,k}(t) + \big(\mu_{e,k}(t) - \mu_k(t)\big)^2 \right],$$

where squares are elementwise. In particular, the variance residual is centered at the *current prototype mean* $\mu_k(t)$, as in the implementation. The exact diagonal optimizer target of Proposition 2 would instead center that term at the responsibility-weighted target mean; the two agree when the prototype mean matches its target. Define the residual norms

$$\Delta^{\mu}_k(t) = \|\mu_k(t) - \tilde{\mu}_{k,\mathcal{B}}(t)\|_2, \qquad \Delta^{\sigma}_k(t) = \|\sigma^2_k(t) - V_{k,\mathcal{B}}(t)\|_2.$$

The plotted relative gaps are

$$G_\mu(t) = \frac{1}{K}\sum_{k=1}^{K} \frac{\Delta_k^\mu(t)}{\max\{\|\mu_k(t)\|_2, \eta\}},$$

$$G_\sigma(t) = \frac{1}{K}\sum_{k=1}^{K} \frac{\Delta_k^\sigma(t)}{\max\{\|\sigma_k^2(t)\|_2, \eta\}}.$$

Figures suppress the iteration argument t and write $\langle a_k \rangle_k = K^{-1}\sum_k a_k$ for the prototype average. The numerical floors apply to empty or very small cluster masses and to the two relative-error denominators. These are current-minibatch stationarity diagnostics, not full-population optimizer distances.

Figure 17 shows the diagnostics for provided-cluster WEATHER-5K at $H \in \{24, 72, 168\}$, with fixed one-hot assignments and uncapped pooling. The relative residuals decrease over training at each horizon. Figure 18 shows the eICU run with inferred responsibilities at $H \in \{6, 12, 24\}$, also using uncapped pooling. Its responsibilities evolve during training, so Proposition 2 describes a conditional target for fixed posteriors and responsibilities, rather than a guaranteed endpoint of joint optimization. The decreasing residuals on both datasets describe empirical moment alignment; neither plot establishes convergence to an exact full-population minimizer.

E.3 Transfer Across Backbones and Population Structures

Within-backbone comparisons and external-model retrofits evaluate the transferability of the complete CLASP pipeline. Prototype-count and training-support sweeps examine its dependence on population structure, followed by an overview of inferred groups in Appendix E.3.5 and detailed WEATHER-5K and eICU cluster results. The relevant architectural comparison is within each backbone family; the enhanced variants include both latent conditioning and validation-fitted readout. Provided geographic assignments and inferred responsibilities remain separate experimental settings.

E.3.1 Within-Backbone Forecasting Results

Table 26 retains the full absolute-error comparison summarized by the percentage reductions in Table 4. Each pair uses the same backbone family, WEATHER-5K test protocol, and provided cluster setting. The enhanced variant adds CLASP latent conditioning and validation-fitted readout; these results therefore evaluate their combined contribution. Plain variants use the convention in Appendix B.1. The plain flow-matching baseline is the raw plain-backbone reference in Table 3; its enhanced counterpart here uses provided clusters ($K = 5$), whereas Table 3 uses inferred responsibilities ($K = 4$).

E.3.2 CLASP Retrofits Across Forecasting Backbones

This comparison evaluates CLASP retrofits of five external forecasters. Its native rows follow the plain-backbone convention in Appendix B.1, retaining all host-native machinery. Appendix C.2 gives the architecture-specific conditioning interfaces, pipeline diagrams, and latent-source conventions. The comparisons below are within each forecasting family.

Evaluation details: The retrofits use the WEATHER-5K test protocol of Table 1. The native models' results in Table 27 are the same as those reported in Table 1. D3VAE and TSDiff use the unified-protocol re-implementations in Appendix C.1.

Results: Table 27 shows lower MAE at every reported horizon for all five forecasters. MSE also decreases except for NsDiff at H=168, where it rises by 1.2% and MAE is nearly unchanged (2.435 to 2.434). At H=24, MAE reductions range from 4.7% for NsDiff to 32.8% for D3VAE. TSFlow reduces MAE/MSE by 11.6%/24.9% at H=24 and 14.7%/20.0% at H=168. For Timer-XL, MAE falls by 23.0% and 10.5% at these two horizons relative to the Timer-XL FT baseline in Table 1; this comparison includes continued fine-tuning with the CLASP module and decoding. The results support the transferability of the complete CLASP retrofit, with gains that vary across backbones,

Table 26: Absolute errors underlying Table 4: within-backbone comparisons on unseen WEATHER-5K stations with provided zones ($K = 5$). Each pair compares a plain backbone with the same backbone enhanced by CLASP conditioning and validation-fitted readout. Latent sources, S, and samplers are specified by backbone in Table 8; plain variants retain the same forecasting budget without a CLASP latent or readout. Lower is better; bold marks the better value within each pair.

		$H = 24$		$H = 72$		$H = 168$	
Family	Variant	MAE	MSE	MAE	MSE	MAE	MSE
Transformer	plain	1.498	6.782	2.065	14.181	2.470	20.995
	+CLASP	**1.448**	**6.299**	**1.999**	**13.101**	**2.365**	**18.644**
GRU	plain	1.544	6.630	2.562	15.253	2.798	22.340
	+CLASP	**1.331**	**5.411**	**1.906**	**12.080**	**2.286**	**17.613**
SSM	plain	1.542	7.061	2.113	14.743	2.519	21.783
	+CLASP	**1.471**	**6.430**	**2.010**	**13.180**	**2.373**	**18.784**
TCN	plain	1.536	7.178	2.092	14.524	2.494	21.395
	+CLASP	**1.398**	**5.789**	**1.942**	**12.348**	**2.309**	**17.846**
DiT	plain	1.425	6.327	2.136	14.843	2.505	20.841
	+CLASP	**1.307**	**5.236**	**1.915**	**12.013**	**2.291**	**17.322**
FM	plain	1.555	6.801	2.350	15.275	2.710	21.035
	+CLASP	**1.291**	**5.111**	**1.902**	**11.848**	**2.290**	**17.193**

horizons, and metrics. All native and CLASP variants use the same training-fitted normalization; these comparisons assess the combined conditioning and forecast-readout pipeline.

Table 27: CLASP retrofits of five external forecasters on WEATHER-5K with provided zones ($K = 5$). The +CLASP rows include latent conditioning and the reported validation-fitted readout. Table 13 gives the latent source, S, and sampler for each retrofit, and Table 11 gives the native probabilistic sampling budgets. Native and CLASP variants use identical training-only normalization (Appendix C.1). Lower MAE and MSE are better; bold marks the best variant within each baseline and column.

		$H = 24$		$H = 72$		$H = 168$	
Baseline	Variant	MAE	MSE	MAE	MSE	MAE	MSE
D3VAE	native	2.306	13.887	3.285	29.050	3.131	26.462
	+ CLASP	**1.550**	**7.291**	**2.083**	**14.304**	**2.486**	**20.699**
TSDiff	native	1.602	8.308	2.321	17.408	2.646	21.316
	+ CLASP	**1.431**	**6.119**	**2.082**	**13.745**	**2.394**	**19.108**
NsDiff	native	1.478	6.355	2.036	13.400	2.435	**19.591**
	+ CLASP	**1.408**	**5.921**	**1.995**	**13.115**	**2.434**	19.835
Timer-XL	native	1.642	7.736	2.194	15.618	2.578	22.501
	+ CLASP	**1.265**	**4.868**	**1.902**	**12.092**	**2.308**	**17.817**
TSFlow	native	1.427	6.398	2.173	13.912	2.426	18.948
	+ CLASP	**1.261**	**4.807**	**1.873**	**11.648**	**2.070**	**15.155**

E.3.3 Sensitivity to the Number of Prototypes

We sweep $K \in \{1, 2, 4, 6, 8, 10\}$ on eICU. Figure 19 averages MAE and MSE in original units over $H \in \{6, 12, 24\}$; Table 28 provides the per-horizon values. The single-prototype setting has the largest horizon-averaged errors, and K=4 is best on both averaged metrics. The K=4 sensitivity-sweep values are 5.801, 6.370, and 6.936 for MAE and 91.291, 105.917, and 120.552 for MSE, yielding the horizon averages 6.369 and 105.920 displayed in Figure 19. They agree with Table 2 at $H = 6$ and $H = 24$. At $H = 12$, the main comparison reports 6.350/104.696 from its selected-

model evaluation, whereas the sensitivity sweep reports $6.370/105.917$; the latter is the value used in both Table 28 and the Figure 19 average. No single K dominates every individual horizon.

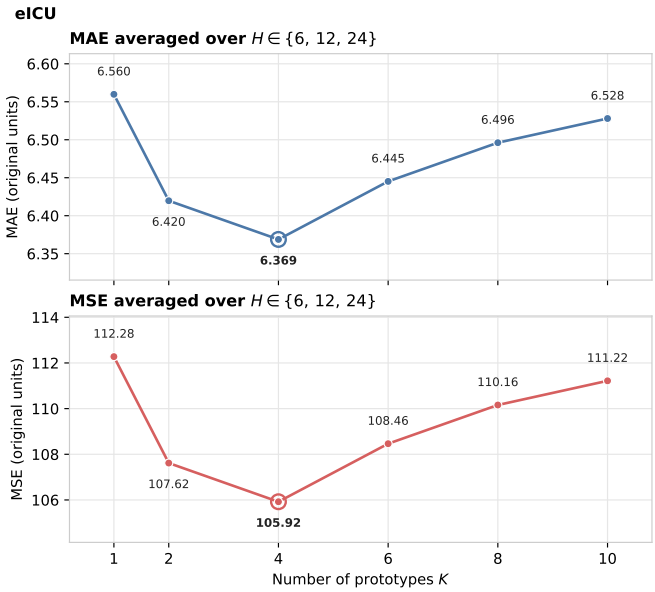

Figure 19: eICU prototype-count sensitivity with inferred responsibilities: horizon-averaged MAE above and MSE below (lower is better). The circle marks K=4.

Table 28: eICU sensitivity to the number of inferred clusters K. The DiT diffusion backbone uses aggregate-prior sampling with $S = 100$ and the full readout (Table 8); K varies by row. Entries are MAE | MSE and are the per-horizon sweep values underlying Figure 19; bold marks the best value per horizon and metric.

K	$H = 6$	$H = 12$	$H = 24$
1	5.850 | 96.805	6.500 | 113.214	7.330 | 126.821
2	5.881 | 91.756	6.253 | 103.620	7.126 | 127.484
4	**5.801** | **91.291**	6.370 | 105.917	**6.936** | **120.552**
6	5.804 | 96.478	6.578 | **103.460**	6.953 | 125.442
8	6.060 | 94.800	**6.130** | 109.160	7.298 | 126.520
10	6.060 | 102.858	6.220 | 109.220	7.304 | 121.582

Table 29: WEATHER-5K sensitivity to the number of inferred clusters K. The flow-matching backbone uses aggregate-prior sampling with $S = 100$ and the full readout (Table 8); K varies by row. Entries are MAE | MSE; bold marks the best value per horizon and metric.

K	$H = 24$	$H = 72$	$H = 168$
1	1.347 | 5.606	1.908 | 11.732	2.329 | 17.341
2	1.304 | 5.151	1.959 | 12.055	2.311 | 17.123
4	**1.280** | **5.049**	**1.894** | **11.691**	**2.298** | **17.040**
6	1.409 | 5.951	1.896 | 11.712	2.354 | 17.549
8	1.328 | 5.359	2.003 | 12.421	2.414 | 18.124
10	1.295 | 5.107	1.932 | 11.903	2.347 | 17.774

WEATHER-5K inferred-cluster sweep: Although the main WEATHER-5K experiments use provided geographic clusters, an inferred-cluster variant tests whether useful station structure can

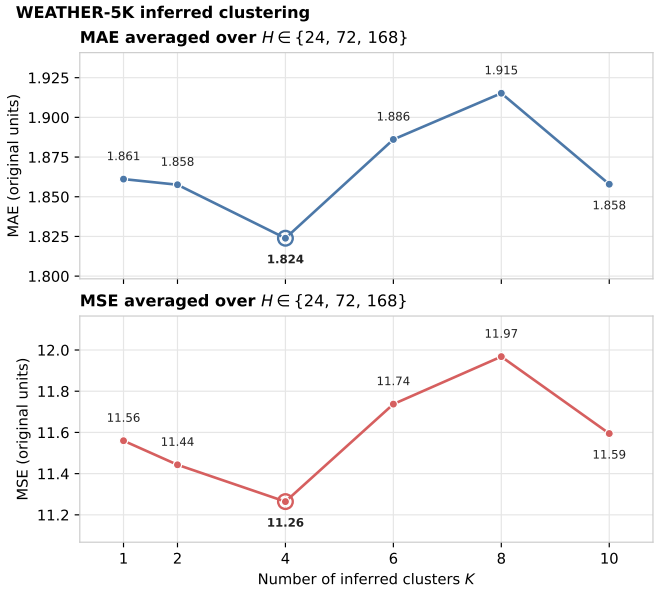

Figure 20: WEATHER-5K inferred-cluster sensitivity: horizon-averaged MAE above and MSE below (lower is better). The circle marks $K{=}4$.

be inferred without labels. Table 29 and Figure 20 report the sweep over $K \in \{1, 2, 4, 6, 8, 10\}$. Performance is stable across moderate K, with $K{=}4$ best on average. Figure 20 averages MAE and MSE over $H \in \{24, 72, 168\}$, and the circle marks this best average setting. The $K{=}1$ row is a controlled no-clustering variant, which retains the context encoder, component-posterior head, and a single global prior, and collapsing to it degrades the 24-hour horizon by 5.2% MAE and 11.0% MSE relative to $K{=}4$, while longer horizons remain within about 2%: the multi-cluster structure contributes most at short horizons. Table 18 compares provided clusters with the $K{=}4$ inferred-cluster variant: Overall MAE and MSE under inferred responsibilities remain within about 2% of those under provided geographic assignments at every horizon. Figure 21 visualizes the learned clusters. With balanced sizes 37/29/30/29, these spatially coherent groups capture longitudinal as well as latitudinal structure.

E.3.4 Partial Pooling Under Limited Training Support

Table 30 evaluates held-out WEATHER-5K forecasting at $H = 24$ as the number of training stations per prototype varies. It compares three matched-capacity variants: per-entity priors without sharing, a single global prior ($K = 1$), and CLASP with shared prototypes. This comparison tests whether cluster-specific sharing improves forecasting beyond either no pooling or population-wide pooling.

The sweep uses a separately balanced training pool of 100 stations, with 20 per zone, rather than the main 99-train/11-validation split. The 20-per-prototype row is therefore a matched-capacity reference, not the exact main-split training set. This sweep and the alignment-weight sweep in Table 25 use independent configurations and do not share runs. The retained sweep does not specify the backbone, test-time latent source, draw count, readout, or assignment of a per-entity prior to an unseen station. These missing details limit reproducibility and the interpretation of the no-pooling control; headline settings should not be inferred from numerical agreement.

CLASP achieves the lowest MAE and MSE at every tested support level, with the largest gains at two training stations per prototype: MAE decreases by 23.2% relative to no pooling and by 7.1% relative to the global prior. Its advantage persists at the largest tested support level, although the margins are smaller than at the lowest support. These results support prototype-based partial pooling as a candidate means of sharing information under limited training support. The incomplete control specification prevents a definitive attribution of these differences to the pooling mechanism alone.

Table 30: Held-out WEATHER-5K forecasting at $H = 24$ versus training stations per prototype. Per-entity uses no shared prior; Global uses one prior ($K = 1$); CLASP uses shared prototypes for the five provided zones ($K = 5$). CLASP performs best at every tested support level, with the largest gains at the lowest support. The sweep uses a separately balanced 100-station training pool, distinct from the main 99-train/11-validation split. Backbone, latent sampling, S, readout, and unseen-station handling for the per-entity control are not documented for this retained sweep.

	Per-entity (no pool)		Global ($K = 1$)		CLASP	
Stations/proto	MAE	MSE	MAE	MSE	MAE	MSE
2	1.812	9.735	1.498	6.814	**1.391**	**6.023**
5	1.564	7.690	1.402	6.154	**1.318**	**5.372**
10	1.421	6.702	1.355	5.712	**1.305**	**5.244**
20	1.348	6.118	1.347	5.606	**1.291**	**5.111**

E.3.5 Cluster Analysis

Figure 21 shows four inferred WEATHER-5K groups containing 37, 29, 30, and 29 stations across the full 125-station subset. Their geographic organization illustrates interpretable structure but is partly expected because routing uses both observed history and longitude/latitude metadata; it does not demonstrate independent discovery of geography. The provided latitude zones are an external reference, not ground-truth clusters that the inferred groups must recover. Table 18 compares forecasting under provided assignments and inferred responsibilities.

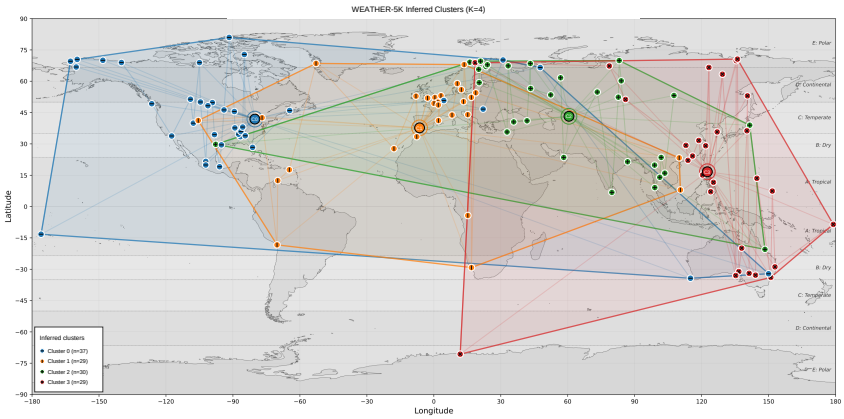

Figure 21: Inferred WEATHER-5K groups ($K = 4$) across all 125 stations, including training, validation, and test entities, overlaid on provided latitude-based zones (Figure 15). Routing uses observed histories and longitude/latitude metadata; the overlay illustrates spatial structure rather than recovery of ground-truth clusters.

On eICU, overlapping diagnosis indicators are metadata inputs, not cluster labels. Cluster-wise errors and feature visualizations in Appendix E.3.7 describe learned forecasting regimes but do not establish clinically meaningful phenotypes. The metadata-removal ablation changes multiple conditioning pathways and therefore does not isolate the router's dependence on history versus metadata.

Appendices E.3.6 and E.3.3 report within-cluster heterogeneity, a separate imbalanced-training comparison, and prototype-count sensitivity. Appendix E.3.4 compares per-entity, global, and shared-prototype priors under limited training support using a separate balanced training pool.

E.3.6 Cluster-wise forecasting on WEATHER-5K

Provided-cluster forecasting: Figure 22 summarizes the provided-cluster WEATHER-5K results. The left-panel boxes span mean $\pm$ standard deviation across $H \in \{24, 72, 168\}$, with a line at the mean. Solid boxes use the balanced flow-matching model with readout under the protocol of Table 1; dashed boxes use the imbalanced diffusion runs of Table 32. The middle panel

projects station variability features—the standard deviations of TMP, DEW, WND_RATE, and SLP over each station's series—using PCA. The right panel measures heterogeneity by average pairwise distance in normalized feature space. Figure 23 gives the eICU analogue.

Tropical and dry stations have lower errors, while continental and polar stations are harder to forecast; the relative ordering of the latter two varies with horizon. The broad grouping appears in both training configurations, but their different backbones preclude attributing error differences solely to training imbalance. The PCA regions partially overlap, and the heterogeneity measure increases from tropical to polar. Together, the panels show geographic structure alongside substantial variation within clusters, motivating shared prototypes with entity-specific latent adaptation.

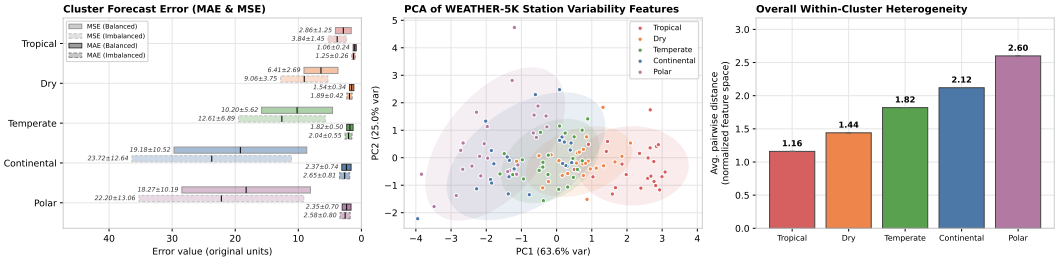

Figure 22: WEATHER-5K analysis with provided zones ($K = 5$): forecast error (left), variability-feature PCA (middle), and within-cluster heterogeneity (right). Boxes show mean ± standard deviation across horizons; solid and dashed boxes denote balanced flow-matching and imbalanced diffusion runs.

Table 31: Cluster-wise WEATHER-5K results for the balanced flow-matching configuration with provided zones ($K = 5$). The aggregate-prior, $S = 100$, full-readout protocol is the same as Table 1, specified in Table 8. Entries are MAE | MSE; lower is better.

Cluster	Horizon	TMP	DEW	WND_sin	WND_cos	WND_RATE	SLP	Overall
Tropical	24	1.016 | 2.285	1.015 | 2.618	0.458 | 0.364	0.463 | 0.332	1.008 | 2.443	0.922 | 1.626	0.814 | 1.611
	72	1.358 | 3.716	1.377 | 4.893	0.505 | 0.418	0.500 | 0.375	1.156 | 3.139	1.545 | 4.657	1.074 | 2.866
	168	1.720 | 5.632	1.787 | 7.379	0.542 | 0.454	0.530 | 0.411	1.237 | 3.501	1.932 | 7.296	1.291 | 4.112
Dry	24	1.638 | 5.622	1.862 | 9.003	0.561 | 0.456	0.500 | 0.377	1.133 | 2.529	1.394 | 3.902	1.181 | 3.648
	72	2.271 | 10.101	2.469 | 13.860	0.587 | 0.495	0.534 | 0.421	1.196 | 2.752	2.451 | 11.708	1.585 | 6.556
	168	2.786 | 14.635	2.866 | 17.595	0.613 | 0.525	0.561 | 0.455	1.284 | 3.157	3.060 | 17.785	1.862 | 9.025
Temperate	24	1.544 | 4.931	1.681 | 5.679	0.443 | 0.300	0.464 | 0.340	1.394 | 3.668	2.147 | 10.943	1.279 | 4.310
	72	2.145 | 8.611	2.357 | 10.245	0.509 | 0.384	0.525 | 0.417	1.535 | 4.435	4.397 | 40.536	1.911 | 10.771
	168	2.463 | 11.082	2.747 | 13.193	0.547 | 0.433	0.566 | 0.465	1.612 | 4.793	5.707 | 63.073	2.274 | 15.506
Continental	24	1.735 | 9.686	1.953 | 10.257	0.411 | 0.313	0.420 | 0.319	2.170 | 9.307	2.833 | 20.199	1.587 | 8.347
	72	2.450 | 16.097	2.829 | 18.302	0.512 | 0.417	0.527 | 0.426	2.625 | 12.781	5.919 | 71.035	2.477 | 19.843
	168	2.966 | 21.627	3.442 | 25.498	0.570 | 0.473	0.590 | 0.498	2.848 | 14.953	7.860 | 113.100	3.046 | 29.358
Polar	24	1.987 | 8.400	1.824 | 7.022	0.489 | 0.395	0.458 | 0.366	1.751 | 5.749	3.052 | 23.890	1.594 | 7.637
	72	2.760 | 15.501	2.659 | 14.179	0.560 | 0.459	0.532 | 0.428	2.060 | 7.496	6.217 | 77.163	2.465 | 19.204
	168	3.235 | 20.470	3.177 | 19.399	0.596 | 0.485	0.561 | 0.449	2.175 | 8.314	8.121 | 118.650	2.978 | 27.961

Table 31 reports the full cluster-wise results of the main model on the balanced test split, with 22 training/validation-pool stations and 3 test stations per zone before the 99/11 training/validation division. It uses flow matching with readout and the ordinary mean of all S=100 blended draws, following Table 1. A consistent difficulty ordering emerges: tropical and dry are easiest, temperate is intermediate, and continental and polar are hardest. This ordering is already visible at 24 hours and becomes more pronounced at longer horizons. Error growth is not uniform across variables: for tropical and dry clusters long-horizon degradation remains moderate, while for temperate, continental, and polar clusters the largest deterioration is concentrated in SLP and, to a lesser extent, TMP and DEW. For example, polar SLP MAE grows from 3.05 at H=24 to 8.12 at H=168. Table 32 retains the training-imbalance robustness check from the diffusion instantiation, with training sizes of Tropical 10, Dry 14, Temperate 16, Continental 22, and Polar 22. Its test split is unchanged, but it uses the previous submission's diffusion protocol; no imbalanced flow-matching run exists. The retained summary does not identify the test-time latent source, draw count, sampler steps, or

readout. Its errors are therefore not directly comparable to Table 31. Tropical and dry remain easier than continental and polar in this separate configuration.

Table 32: Cluster-wise WEATHER-5K results with provided zones ($K = 5$) and imbalanced diffusion training; s is the training-station count per zone. The retained run's latent source, S, sampling steps, and readout are unspecified; this is a separate auxiliary comparison. Entries are MAE | MSE; lower is better.

Cluster, size	Horizon	TMP	DEW	WND_sin	WND_cos	WND_RATE	SLP	Overall
Tropical, s=10	24	1.318 | 3.778	1.234 | 4.165	0.491 | 0.440	0.529 | 0.493	1.171 | 3.289	1.160 | 2.409	0.984 | 2.429
	72	1.642 | 4.976	1.595 | 5.927	0.561 | 0.515	0.569 | 0.515	1.344 | 4.331	1.818 | 6.332	1.255 | 3.766
	168	1.893 | 6.648	2.035 | 8.865	0.588 | 0.560	0.595 | 0.567	1.436 | 4.913	2.465 | 10.387	1.502 | 5.323
Dry, s=14	24	1.928 | 7.333	2.253 | 12.020	0.661 | 0.717	0.563 | 0.534	1.332 | 3.139	1.789 | 6.032	1.421 | 4.962
	72	2.735 | 13.982	3.153 | 20.990	0.752 | 0.889	0.676 | 0.671	1.564 | 3.917	3.251 | 18.831	2.022 | 9.880
	168	2.943 | 16.244	3.565 | 25.111	0.715 | 0.752	0.702 | 0.740	1.403 | 3.450	3.996 | 27.662	2.221 | 12.326
Temperate, s=16	24	1.738 | 5.786	1.961 | 7.198	0.462 | 0.355	0.523 | 0.566	1.604 | 5.004	2.497 | 14.841	1.464 | 5.625
	72	2.224 | 7.861	2.549 | 11.519	0.533 | 0.441	0.604 | 0.681	1.738 | 5.738	4.916 | 50.632	2.094 | 12.812
	168	2.595 | 12.012	3.097 | 16.207	0.567 | 0.484	0.654 | 0.756	1.847 | 6.563	6.555 | 80.333	2.552 | 19.393
Continental, s=22	24	2.038 | 12.738	2.193 | 12.603	0.429 | 0.390	0.451 | 0.404	2.457 | 12.937	3.245 | 25.878	1.802 | 10.825
	72	2.773 | 19.431	3.050 | 20.686	0.543 | 0.526	0.586 | 0.526	3.008 | 18.161	6.571 | 86.122	2.755 | 24.242
	168	3.357 | 27.483	3.749 | 30.430	0.600 | 0.576	0.671 | 0.728	3.302 | 21.024	8.740 | 136.350	3.403 | 36.098
Polar, s=22	24	2.140 | 9.569	2.030 | 8.431	0.512 | 0.468	0.499 | 0.485	1.885 | 6.767	3.316 | 27.749	1.730 | 8.912
	72	2.928 | 17.361	2.878 | 16.261	0.593 | 0.546	0.580 | 0.588	2.201 | 9.015	6.840 | 92.129	2.670 | 22.650
	168	3.498 | 23.891	3.502 | 23.304	0.635 | 0.587	0.652 | 0.672	2.380 | 10.732	9.288 | 150.995	3.326 | 35.030

E.3.7 CLUSTER-WISE FORECASTING ON EICU

Table 33 and Figure 23 summarize the forecasting behavior of the inferred clusters on eICU with K=4. The cluster-wise MAE and MSE remain relatively balanced, without any cluster exhibiting extremely low errors or severe degradation across horizons, indicating that the learned router partitions the patients into reasonably coherent groups rather than collapsing the difficult cases into one cluster. The left panel of Figure 23 shows cluster 1 is the easiest group on average and cluster 2 the most challenging, with clusters 0 and 3 in between. The PCA projection in the middle panel reveals a meaningful coarse organization: the four clusters occupy different regions of the feature plane with moderate overlap near the center. The within-cluster heterogeneity plot in the right panel is consistent: cluster 1 has the lowest average normalized pairwise distance and cluster 2 the highest, matching the forecasting difficulty ordering. Together these observations suggest the inferred clusters describe overlapping forecasting regimes while maintaining a balanced distribution of difficulty.

Cluster IDs refer to the inferred groups. In Figure 23, error boxes summarize $H \in \{6, 12, 24\}$ with a line at the mean and boxes extending one standard deviation on either side. Only one training configuration is shown, as eICU has no imbalanced-training counterpart. The middle panel uses 300 patients per cluster for the PCA of latent features; the right panel computes average pairwise distances in normalized feature space. The errors are more tightly grouped than in the WEATHER-5K analysis of Figure 22.

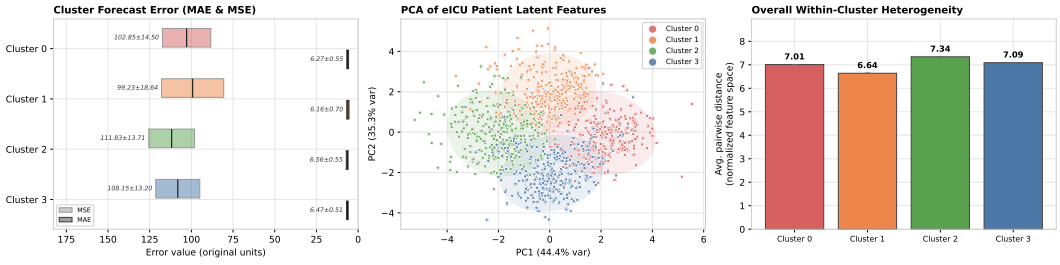

Figure 23: eICU inferred-cluster analysis (K=4): forecast error (left), latent-feature PCA (middle), and within-cluster heterogeneity (right). Boxes show mean ± standard deviation across horizons.

Table 33: Cluster-wise eICU results with inferred clusters (K=4). The DiT diffusion backbone uses aggregate-prior sampling with $S = 100$ and the full readout (Table 8). Entries are MAE | MSE; lower is better. The per-vital CLASP entries in Table 19 are unweighted macro-averages over these four clusters (equal cluster weighting), which coincides with pooled patient/window performance under the balanced evaluation's equal per-cluster window denominators.

Cluster	Horizon	SaO_2	HR	RR	NIBP-S	NIBP-D	NIBP-M	Overall
0	6	1.408 | 7.517	5.443 | 64.252	2.147 | 12.297	10.527 | 207.928	7.117 | 115.670	7.277 | 110.822	5.653 | 86.414
	12	1.397 | 6.542	6.651 | 92.762	2.568 | 17.419	11.940 | 264.001	7.773 | 133.239	8.299 | 135.642	6.438 | 108.268
	24	1.571 | 8.124	7.452 | 111.083	2.783 | 17.949	11.967 | 271.719	7.978 | 133.431	8.538 | 140.872	6.715 | 113.863
1	6	1.296 | 6.507	5.484 | 66.520	2.179 | 11.155	9.711 | 187.956	7.005 | 110.632	7.274 | 108.168	5.491 | 81.823
	12	1.387 | 8.070	6.500 | 85.133	2.404 | 14.064	10.932 | 231.968	7.652 | 121.747	7.724 | 120.867	6.100 | 96.975
	24	1.583 | 7.398	7.789 | 119.772	2.808 | 22.380	12.407 | 276.829	8.240 | 139.465	8.442 | 147.538	6.878 | 118.897
2	6	1.503 | 9.351	5.653 | 74.554	2.271 | 13.606	11.332 | 234.568	7.585 | 135.778	8.110 | 137.103	6.076 | 100.827
	12	1.599 | 8.751	6.776 | 98.267	2.632 | 16.305	11.428 | 246.503	7.823 | 137.058	8.433 | 137.924	6.448 | 107.468
	24	1.668 | 9.436	8.019 | 125.266	2.915 | 20.908	12.668 | 301.311	8.589 | 150.925	9.053 | 155.253	7.152 | 127.183
3	6	1.256 | 6.457	5.960 | 73.687	2.211 | 12.374	10.914 | 232.700	7.772 | 125.532	7.790 | 125.852	5.984 | 96.100
	12	1.580 | 7.726	6.800 | 97.748	2.368 | 14.452	11.709 | 243.856	7.911 | 134.037	8.105 | 138.640	6.412 | 106.076
	24	1.594 | 7.315	7.751 | 125.982	2.745 | 17.191	12.843 | 283.782	8.224 | 148.486	8.842 | 150.834	7.000 | 122.265

E.4 Aggregate Latent, Fixed-Predictor, and Reference Diagnostics

These analyses connect the probabilistic latent partial-pooling mechanism to its theoretical interpretation. The aggregate-Gaussian diagnostic measures the training–deployment latent gap and its one-hot correspondence to uncapped pooling. The subsequent fixed-predictor bounds and reference diagnostics use their stated sampling and predictor conditions; they do not isolate the cause of forecasting gains or certify the aggregate diagnostic by numerical substitution.

E.4.1 Aggregate latent divergence diagnostics

The diagnostic in Table 34 uses one frozen flow-matching checkpoint with $K = 5$ and $H = 24$, evaluated on the same 256 windows from each of 15 held-out WEATHER-5K stations. Only the responsibility rule changes; component posteriors and prototype priors remain fixed. It is separate from the inferred-$K = 4$ forecasting comparison in Table 6. Gaussian aggregate KL, uncapped weighted pooling, and the unweighted component-KL sum are computed per window and then averaged over windows within each station and equally across stations.

Table 34: Aggregate posterior–prior KL and uncapped pooling diagnostics for Section 3.4, separate from the $K = 4$ inference-source comparison. One frozen flow-matching checkpoint ($K = 5$, $H = 24$) uses the same 256 windows from each of 15 held-out WEATHER-5K stations. Only responsibilities change; component laws remain fixed. Entries are mean divergences in nats, averaged over windows and stations. The component sum $\mathcal{C}_e^{\mathrm{comp}}$ is computed from per-window component KLs under the same station-balanced protocol; because the component laws are fixed, it is identical across responsibility rules. All columns concern the Gaussian component/aggregate construction; no categorical-mixture KL is reported here.

Responsibility rule	Aggregate KL $\mathcal{C}_e^{\mathrm{agg}}$	Weighted pooling $\mathcal{L}_{\mathrm{pool},e}$	Component sum $\mathcal{C}_e^{\mathrm{comp}}$
Provided one-hot weights	0.97	0.97	6.84
Router-inferred soft weights	1.21	1.38	6.84

Under one-hot weights, aggregate KL and uncapped pooling both equal 0.97 nats, consistent with Corollary 1. With soft weights, aggregate KL is 1.21 nats and weighted pooling is 1.38 nats. This distinction motivates evaluating the aggregate laws directly; weighted pooling is not a numerical substitute for aggregate KL. These quantities are distinct from categorical mixture KL and the reference-complexity measurements in Appendix E.4.3. The observed ordering establishes neither a general pooling upper bound nor a forecasting-risk guarantee.

The station-balanced mean of the unweighted component sum $\mathcal{C}_e^{\mathrm{comp}} = \sum_k D_{\mathrm{KL}}(q_{e,k} \| P_k)$ is 6.84 nats under both rules. Its invariance follows from keeping the component laws fixed. With the

same averaging protocol, the reported aggregate means satisfy the data-processing inequality of Theorem 1: $0.97 \leq 6.84$ and $1.21 \leq 6.84$. Table 34 thus checks the averaged relation alongside the one-hot correspondence; these summary means do not establish that every per-window numerical calculation satisfies the inequality. The component sum is distinct from the mixture-reference scores and optional diagnostic slack used in the subsequent fixed-predictor calculations. The reference-law certificates and their applicability conditions remain in Appendices A.2.5 and A.4.5.

E.4.2 Held-Out Fixed-Predictor Concentration Evaluation

Table 35: Conditional bounds on $R^{\mathrm{fix}}_{\mathrm{env}}$ for a frozen diffusion predictor on 15 held-out WEATHER-5K stations (provided zones, $K = 5$, $H = 24$). The rule uses aggregate-posterior latent draws and 25-step DDIM trajectories; the empirical column averages the bounded normalized trajectory loss in equation 70, with $M = 16$ draws per window. Across-entity and Within-entity terms account for finite sampling under the stated i.i.d. assumptions. Both rows retain optional nonnegative diagnostic slack; the second adds the reported prototype score. The bounds concern this posterior-sampling rule, not the deployed prior-sampling predictor or ensemble-point error.

Diagnostic slack	Mean loss	Across-entity	Within-entity	$R^{\mathrm{fix}}_{\mathrm{env}}$ bound
C_ε	0.229	0.437	0.237	0.903
$C_\varepsilon + a_{\mathrm{proto}}$	0.229	0.437	0.241	0.907

Predictor and loss. Table 35 evaluates a frozen diffusion predictor on 15 held-out WEATHER-5K stations at $H = 24$. Its encoder, router, prototype parameters, backbone, normalization, and sampling rule are fixed before these entities are observed. The complete randomized prediction rule $\mathsf{F}(x, m_e; \xi)$ is therefore fixed as in Remark 3, although it recomputes the aggregate posterior for each context. There is no single Gaussian Q_e shared across all windows. The saved empirical loss averages $M = 16$ independent aggregate posterior draws and 25-step DDIM trajectories per window. Readout use is not documented for this retained calculation, so its forecasting rule cannot be identified with either the raw or decoded headline evaluator solely from these summaries. The bounded trajectory loss is

$$\ell(\widehat{y}, y) = \frac{1}{Hd} \sum_{t=1}^{H} \sum_{j=1}^{d} \min\left\{ \frac{|\widehat{y}_{tj} - y_{tj}|}{2}, 1 \right\}, \tag{70}$$

where d is the number of target channels, targets and predictions use training-split normalization, and the clipping scale is fixed before evaluation. This sampled-trajectory loss differs from the ensemble-point errors and interval coverage in the main forecasting tables.

For each window and its fresh random seeds, define

$$L^{(M)}_{e,i} = \frac{1}{M} \sum_{s=1}^{M} \ell\Big(\mathsf{F}(x^{(i)}_e, m_e; \xi_{e,i,s}), y^{(i)}_e\Big), \qquad \widehat{R}^{\mathrm{fix}}_{e,\mathrm{MC}} = \frac{1}{n_e} \sum_{i=1}^{n_e} L^{(M)}_{e,i}. \tag{71}$$

A window and its M seeds form one augmented observation with loss in $[0, 1]$ and expectation R^{fix}_e from equation 53. Thus the effective sample size is n_e, not $n_e M$. The corresponding environment risk is $R^{\mathrm{fix}}_{\mathrm{env}} = \mathbb{E}_{e \sim \mathcal{T}} R^{\mathrm{fix}}_e$. It concerns the frozen posterior-sampling procedure, not the deployed prior-sampling rule or the loss of an ensemble-mean forecast.

Windows and confidence. Each complete context–target window spans 72 hours. The initial block contains 256 evaluated windows per station, spaced at least 273 hours apart, so $n_e = 256$ is the number of losses actually averaged in $\widehat{R}^{\mathrm{fix}}_{e,\mathrm{MC}}$. A chronological embargo separates this block from a later block with 64 evaluated windows per station, whose Monte Carlo mean loss is $\widehat{R}^{\mathrm{later}}_e$. We use total $\delta = 0.05$, split as $\delta/2$ across entities and $\delta/(2N)$ for each within-entity bound; the auxiliary per-entity analyses below retain the nominal $\delta = 0.05$ per entity and do not share this simultaneous guarantee. Interpreting both blocks as samples from the same D_e requires the stated within-entity sampling assumption. Temporal spacing and non-overlap do not establish independence or a common data distribution. The environment-risk interpretation additionally assumes

i.i.d. entity distributions and metadata, independently of the frozen predictor. The sorted-ID station holdout does not establish that environment sampling assumption, so the reported guarantees are conditional on it.

Two-level fixed-predictor bound (Table 35). For $N, n_e \geq 8$ and $0 < \delta < 1$, apply equation 43 on a singleton hypothesis space to the augmented windows at confidence $\delta/(2N)$ per entity, and to the i.i.d. entity risks at confidence $\delta/2$. The hypothesis KL is zero because the complete prediction rule is fixed independently of the certified entities. A union bound and Pinsker's inequality give, with probability at least $1 - \delta$,

$$R_{\text{env}}^{\text{fix}} \leq \frac{1}{N}\sum_{e} \widehat{R}_{e,\text{MC}}^{\text{fix}} + \sqrt{\frac{\ln(4\sqrt{N}/\delta)}{2N}} + \frac{1}{N}\sum_{e}\sqrt{\frac{\ln(4N\sqrt{n_e}/\delta)}{2n_e}}.$$

Bounding the last terms using $n_{\min}$ and $n_{\max}$ and adding any nonnegative slack $\bar{a}$ yields, on the same event,

$$R_{\text{env}}^{\text{fix}} \leq \frac{1}{N}\sum_{e=1}^{N} \widehat{R}_{e,\text{MC}}^{\text{fix}} + \sqrt{\frac{\ln\frac{4\sqrt{N}}{\delta}}{2N}} + \sqrt{\frac{\bar{a} + \ln\frac{4N\sqrt{n_{\max}}}{\delta}}{2n_{\min}}}. \tag{72}$$

This remains valid for data-dependent $\bar{a} \geq 0$: adding slack only increases the bound. The augmented-observation argument includes Monte Carlo randomness directly and needs no separate Monte Carlo correction. It is distinct from the latent-law argument in Theorem 5 and Corollary 3.

We use $N = 15$, $n_{\min} = n_{\max} = 256$, $\delta = 0.05$, $\varepsilon = 0.1$, $B = 32$, and $K = 5$, giving $C_\varepsilon = \varepsilon B + (K-1)\ln(5/\varepsilon) = 18.848$ nats. The two rows retain $\bar{a} = C_\varepsilon$ and $\bar{a} = C_\varepsilon + a_{\text{proto}}$, respectively, with the reported prototype score $a_{\text{proto}} = 0.97$ nats used solely as nonnegative slack. Window-dependent penalties such as equation 74 do not require a single latent law. These nonnegative additions retain the reported numerical comparisons; neither is required to certify the fixed predictor. In particular, B is a parameter of this diagnostic addition, not a verified bound on latent divergences. The retained score is not identified with $D_{\text{KL}}(Q_e \| P_{c_e})$ for a single entity-level law. Agreement with the reported weighted-mixture mean in Table 37 does not establish that the laws, checkpoints, or averaging procedures coincide. Table 35 reports the resulting terms and bounds: both bounds are below 1 and above the pooled later-block loss 0.239 (Table 36). The limited number of entities makes the across-entity term the dominant source of looseness.

Auxiliary per-entity bounds. For a matched vector of station-level sampled losses and nonnegative diagnostic additions, define the upper binary-kl endpoint and its nominal per-entity use by

$$\begin{aligned} \mathcal{U}_n(b, a; \delta) &= \sup\left\{ u \in [b, 1] : \text{kl}(b \| u) \leq \frac{a + \ln(2\sqrt{n}/\delta)}{n} \right\}, \\ U_e(a) &= \mathcal{U}_{n_e}(\widehat{R}_e^{\text{fix}}, a; \delta), \qquad a \geq 0, \end{aligned} \tag{73}$$

with $\delta = 0.05$ per entity. This inversion is nonlinear and must be applied before averaging: a mean empirical loss and a mean complexity do not determine a mean bound. The window-count sweep in Table 36 instead uses the separately specified Monte Carlo correction and simultaneous confidence allocation in Appendix E.4.3. The saved artifact does not contain the matched per-station single-draw losses and divergences used by an earlier auxiliary analysis, so Tables 37 and 38 report complexity diagnostics only rather than unreproducible mean bounds. The raw component diagnostic is

$$D_e^{\text{proto}} = \frac{1}{|\mathcal{W}_e|}\sum_{w \in \mathcal{W}_e}\sum_{k=1}^{K} r_{wk} D_{\text{KL}}(q_{w,k} \| P_k), \tag{74}$$

where $\mathcal{W}_e$ contains the evaluated initial-block windows. The component-KL diagnostic and simplex-cover penalty are optional nonnegative slack for the same fixed predictor: its context-dependent router requires no simplex-cover charge, and D_e^{proto} is not the cross-prototype complexity in equation 44. Since $U_e(a)$ is nondecreasing in a, these additions can only loosen the per-entity bound. They do not show that reducing the training penalty causes improved generalization. Exact KL values must also be distinguished from stabilized numerical diagnostics. Zero hypothesis KL does not assert a latent budget $B = 0$; a direct use of Theorem 4 still requires a prespecified divergence budget and the transfer term εB.

Reported diagnostics. The conditional environment-risk bound of Table 35 is non-vacuous when it is below $1 - 10^{-6}$; its agreement check is whether the pooled later-block loss $(1/N)\sum_e \widehat{R}_e^{\mathrm{later}}$ lies below the bound. For the auxiliary per-entity bounds, a bound is non-vacuous when $U_e < 1 - 10^{-6}$ (entities with $n_e < 8$ are assigned the vacuous value 1) and agreement is the fraction of stations satisfying $\widehat{R}_e^{\mathrm{later}} \le U_e$. Both checks are comparisons with finite later-block estimates, not guarantees that future empirical losses will fall below the bound or verifications of the sampling assumptions. Such per-entity bound and agreement summaries are reported only for analyses with retained matched station-level inputs. Appendix E.4.3 reports window-count analyses, training-support diagnostics, and eICU sample-size limitations.

E.4.3 Additional Fixed-Predictor and Reference Diagnostics

These analyses complement Table 35 through sample-size evaluations for a frozen prediction rule, reference-complexity comparisons for a fixed predictor, and training-station diagnostics. They examine the statistical cost of certification. Predictive ablations of prototype learning and limited training support are reported separately in Appendices E.2 and E.3.4.

Evaluation cohorts and sampling. The fixed-predictor evaluation uses the 15 held-out WEATHER-5K stations, each supporting 256 initial-block and 64 later-block windows, and 32 eICU stays selected with seed zero from the 503 test stays supporting eight initial and two later windows. The selected eICU stays form a long-stay cohort and are not representative of the entire test cohort. The WEATHER-5K calculation retains the provided-$K = 5$, $H = 24$ posterior-sampling diffusion protocol described in Appendix E.4.2. The eICU summary does not identify its backbone, horizon, assignment rule, K, latent source, numerical sampler, or readout. Its cohort, loss, and Monte Carlo budget below are documented, but these do not reconstruct the full prediction rule or establish that it matches the headline eICU model.

Here, "frozen" means that the trained prediction rule and its sampling procedure are fixed before observing the evaluation entities. The rule may recompute posterior laws and responsibilities for each window; a common entity-level Gaussian Q_e is not assumed. This evaluation targets R_e^{fix} and treats recorded reference scores only as optional nonnegative slack.

We reserve the last 20% of stride-one window starts for the later block and exclude $T_{\mathrm{ctx}} + H - 1$ starts between blocks. Within each block, we select evenly spaced windows from a grid with spacing $T_{\mathrm{ctx}} + H$, ensuring that selected contexts and targets share no raw rows. Smaller WEATHER-5K samples are nested subsets of the 256 initial windows, with empirical losses recomputed on each subset.

Monte Carlo correction and confidence allocation. The bounded loss is equation 70 on the saved training-normalized scale. Write $\widehat{R}_{e,\mathrm{MC}}$ for the sampled mean in equation 71. We use 16 independent latent and trajectory draws per WEATHER-5K window and 64 per eICU window, totaling 97,280 forecast trajectories. Before evaluation, we set $\varepsilon = 0.1$, $B = 32$, and total failure probability 0.05. We retain a conservative budget of at most 184 prespecified entity and sample-count comparisons, giving $\delta_{\mathrm{PB}} = 0.025/184$. The table reports 92 initial-block comparisons ($15 \times 4 + 32$). The remaining 0.025 covers 139 Monte Carlo estimates—these 92 initial-block estimates and 47 later-block estimates—giving $\delta_{\mathrm{MC}} = 0.025/139$.

Let $\ell_e^{\mathrm{av}}(x, y) = \mathbb{E}_\xi[\ell(\mathsf{F}(x, m_e; \xi), y)]$ and $\widehat{R}_e^{\mathrm{av}} = n_e^{-1}\sum_i \ell_e^{\mathrm{av}}(x_e^{(i)}, y_e^{(i)})$. Under the within-entity i.i.d. assumption, the singleton PAC-Bayes inequality gives

$$\mathrm{kl}(\widehat{R}_e^{\mathrm{av}} \| R_e^{\mathrm{fix}}) \le \frac{\ln(2\sqrt{n_e}/\delta_{\mathrm{PB}})}{n_e}$$

with failure probability δ_{PB}. Conditional on the observed windows and fixed prediction rule, Hoeffding's inequality for the $n_e M$ independent bounded draws gives $\widehat{R}_e^{\mathrm{av}} \le \widehat{R}_e^{+}$ with failure probability δ_{MC}, where

$$\widehat{R}_e^{+} = \min\left\{1, \widehat{R}_{e,\mathrm{MC}} + \sqrt{\frac{\ln(1/\delta_{\mathrm{MC}})}{2n_e M}}\right\}.$$

Denote the nonnegative reference score used for an entity and sample-count comparison by $d^{\mathrm{ref}}_{e,n}$. Retain the reported diagnostic addition

$$a_{e,n} = d^{\mathrm{ref}}_{e,n} + C_\varepsilon \geq 0, \qquad C_\varepsilon = \varepsilon B + (K-1)\ln(5/\varepsilon),$$

and report

$$U^{\mathrm{MC}}_e = \mathcal{U}_{n_e}(\widehat{R}^+_e, a_{e,n}; \delta_{\mathrm{PB}}).$$

The upper endpoint in equation 73 is nondecreasing in both its empirical-risk argument and its nonnegative slack. Thus the two concentration events imply $R^{\mathrm{fix}}_e \leq U^{\mathrm{MC}}_e$; if $\widehat{R}^+_e > R^{\mathrm{fix}}_e$, the conclusion is immediate. The scores may depend on the observed data, since adding them only loosens the zero-hypothesis-KL bound. Neither $d^{\mathrm{ref}}_{e,n}$ nor C_ε is required for the fixed-predictor guarantee, and neither is identified with a single-law latent complexity. In particular, B specifies diagnostic slack, not a verified divergence budget for these window-dependent laws.

This simultaneous confidence allocation applies to the fixed-predictor sample-size analysis below, conditionally on its sampling assumptions. Nested subsets and reused samples across comparisons do not invalidate the union bound. The subsequent reference comparisons report measured complexities only. Recomputing bounds for those comparisons requires matched per-station empirical losses and divergences, with nominal $\delta = 0.05$ per entity in equation 73; the Monte Carlo-averaged loss from this sweep is not a substitute.

Measured sample-size sensitivity. Table 36 reports subset-specific losses and conditional fixed-predictor bounds with the retained nonnegative reference scores. The mean WEATHER-5K bound decreases from 0.8955 at 32 windows to 0.5029 at 256 windows, while the initial-block loss remains near 0.23. This improvement reflects both the larger evaluated sample and the smaller Monte Carlo correction, which is applied before bound inversion. The later block remains fixed throughout the WEATHER-5K sweep.

These results show how additional windows sharpen the evaluation of the fixed predictor; they do not measure gains from retraining the prototypes. The averaged entity-level bounds and their confidence allocation differ from the environment-risk calculation in Table 35.

Table 36: Conditional bounds on the per-entity sampled-trajectory risk R^{fix}_e of the frozen randomized prediction rules in Appendix E.4.3, using the bounded normalized loss in equation 70. The cohorts contain 15 WEATHER-5K stations and 32 selected eICU long stays. Bounds include Monte Carlo correction ($M = 16$ and 64, respectively), optional nonnegative diagnostic slack, and the stated simultaneous confidence allocation under the within-entity sampling assumptions. Entries are entity means, including means of individual bounds U^{MC}_e, rather than environment-risk bounds. Excess is $U^{\mathrm{MC}}_e - \widehat{R}_{e,\mathrm{MC}}$. The retained eICU predictor settings and readout status are incomplete, as detailed in Appendix E.4.3.

Dataset	n_e	Initial loss	MC margin	Later loss	U^{MC}_e	Excess
WEATHER-5K	32	0.230	0.092	0.239	0.8955	0.666
WEATHER-5K	64	0.231	0.065	0.239	0.7586	0.528
WEATHER-5K	128	0.230	0.046	0.239	0.6176	0.388
WEATHER-5K	256	0.229	0.032	0.239	0.5029	0.274
eICU	8	0.227	0.092	0.235	0.9994	0.773

Every reported mean bound is below $1-10^{-6}$, and all reported later-block empirical means lie below their corresponding mean bounds, although the eICU bounds remain close to one. The simultaneous two-sided Monte Carlo radii for later-block loss are 0.067 on WEATHER-5K and 0.191 on eICU. This agreement is a descriptive comparison, not evidence establishing temporal independence or stationarity.

Reference-prior comparisons for a fixed predictor. The diagnostic addition in Table 35 motivates examining reference structure separately. Holding the predictor fixed, Table 37 compares a fitted single Gaussian $P^\star_{\mathrm{single}}$, the uniform categorical reference mixture $\overline{P} = P^{\mathrm{mix}}_{u_K} = K^{-1}\sum_k P_k$, where $u_K = (1/K, \ldots, 1/K)$, and the responsibility-weighted categorical reference mixture $P^{\mathrm{mix}}_{r_e} = \sum_k r_{ek} P_k$. The evaluation uses the 15 held-out WEATHER-5K stations at $H = 24$, with

$n_e = 256$ and $K = 5$. These comparisons hold the posterior law Q_e fixed and change its reference law. They do not replace the deployed latent law with a categorical mixture. Their mixture KLs are separate from $\mathcal{C}_e^{\mathrm{agg}}$; the aggregate Gaussian KL formula does not in general evaluate them.

The mean reference penalty decreases from 4.12 nats for the single Gaussian to 1.38 for the uniform mixture and 0.97 for the responsibility-weighted mixture. Thus, prototype structure provides a lower-complexity reference for the same entity posteriors in this experiment, with responsibility weighting yielding a further reduction over uniform weighting. This is an observed reduction in reference divergence for the fixed entity posteriors; it does not identify the mechanism behind forecasting gains.

Table 37: Categorical-mixture reference-complexity comparison on the 15 held-out WEATHER-5K stations ($H = 24$, $n_e = 256$, $K = 5$). The singleton row gives the zero-penalty baseline. Measured KL penalties can be used as the nonnegative arguments a_e in $U_e(a_e)$ from equation 73 only when paired with the corresponding station-level empirical losses. The mixture rows compare the fixed law Q_e with categorical reference mixtures, not the aggregate Gaussian KL $\mathcal{C}_e^{\mathrm{agg}}$. These are reference-divergence diagnostics; no forecast sampling budget or readout applies, and no forecasting-risk bound is reported. The weighted reference's assignment source and the source checkpoint are not identified in the retained summary.

Reference	Mean KL (nats)
Singleton ($a = 0$)	0.00
Single Gaussian $P^{\star}_{\mathrm{single}}$	4.12
Uniform categorical mixture $\overline{P}$	1.38
Weighted categorical mixture $P^{\mathrm{mix}}_{r_e}$	0.97

The ordering is empirical: Lemma 3 bounds $D_{\mathrm{KL}}(Q_e \| P^{\mathrm{mix}}_{r_e})$ by $\mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) = \sum_k r_{ek} D_{\mathrm{KL}}(Q_e \| P_k)$, with the same Q_e in every summand. This reference complexity is distinct from the componentwise pooling penalty, which uses $q_{e,k}$ in summand k, and the lemma does not guarantee superiority over an arbitrary fitted Gaussian. Moreover, the reported mean divergences do not supply numerical mean certificates without matched station-level empirical losses and divergences.

A fixed-prior PAC-Bayes interpretation additionally requires independence of the reference from the certified windows, including the fitted Gaussian parameters and mixture weights. Data-dependent weights require an appropriate correction or the fixed uniform-mixture construction below; freezing them after observing the certified sample does not establish independence.

Direct uniform-mixture analysis versus a simplex cover. For a prototype tuple fixed independently of the certified windows, the uniform mixture provides an entity-level reference bound without a simplex cover. Let $u_K = (1/K, \ldots, 1/K)$ and $\overline{P} = P^{\mathrm{mix}}_{u_K} = K^{-1} \sum_k P_k$. Marginalizing the joint laws $r_k Q$ and $u_{K,k} P_k$ gives

$$D_{\mathrm{KL}}(Q \| \overline{P}) \le D_{\mathrm{KL}}(r \| u_K) + \mathcal{C}_e^{\mathrm{ref}}(Q, r) \le \mathcal{C}_e^{\mathrm{ref}}(Q, r) + \ln K,$$

where $\mathcal{C}_e^{\mathrm{ref}}(Q, r) = \sum_k r_k D_{\mathrm{KL}}(Q \| P_k)$. Standard PAC-Bayes with the fixed prior $\overline{P}$ is uniform over posteriors Q_e, so this construction requires neither a simplex cover nor the divergence budget B.

Table 38 compares the direct mixture KL with the joint-law surrogate under the learned soft router responsibilities r_e, for which $D_{\mathrm{KL}}(r_e \| u_K) < \ln K$. The direct complexity is 1.38 nats, compared with 2.27 for the surrogate. The difference reflects additional conservatism from the surrogate rather than a change in the predictor or uniform-mixture reference.

These soft-router complexities are not interchangeable with the reported nonnegative diagnostic addition in Table 35. For a single latent law under one-hot responsibilities, the joint-law expression instead reduces to $D_{\mathrm{KL}}(Q_e \| P_{c_e}) + \ln K$; the fixed-predictor table does not establish a value for that single-law quantity.

For comparison, the full cover-based complexity is

$$a_e^{\mathrm{cover}} = \mathcal{C}_e^{\mathrm{ref}}(Q_e, r_e) + \varepsilon B + (K - 1) \ln(5/\varepsilon).$$

Table 38: Direct and joint-law categorical-reference complexities on the 15 held-out WEATHER-5K stations ($H = 24$, $n_e = 256$, $K = 5$). Both rows use the same categorical uniform-mixture reference $\overline{P} = P^{\mathrm{mix}}_{u_K}$, rather than P^{agg}_e. The joint-law surrogate uses learned soft router responsibilities, a different setting from the provided-cluster diagnostic in Table 35. These are reference-complexity diagnostics, not numerical forecasting-risk bounds; forecast sampling and readout are inapplicable. The source checkpoint is not identified.

Complexity a	Mean nats	Excess over direct
Direct $D_{\mathrm{KL}}(Q_e \| \overline{P})$	1.38	—
Joint-law $\mathcal{C}^{\mathrm{ref}}_e + D_{\mathrm{KL}}(r_e \| u_K)$	2.27	$+0.89$ nats

At $K = 5$, $\varepsilon = 0.1$, and $B = 32$, the additive cover penalty is $3.2 + 4 \ln 50 \simeq 18.848$ nats, exceeding $\ln 5$. Consequently, the direct uniform-mixture complexity is no larger than the full cover complexity for these settings. With matched empirical risks and confidence budgets, its kl-inverted entity-level bound is therefore no looser. This comparison distinguishes the cost of the prototype reference from conservatism introduced by the bound construction.

Training-support diagnostics. A separate diffusion-checkpoint sweep varies the number of training stations per prototype and evaluates each fitted predictor on those same training stations. Its diagnostic scores are $U_e(C_{\mathrm{net}} + D^{\mathrm{proto}}_e)$ with $n_e = 256$ and $\delta = 0.05$. Table 39 summarizes three seeds for two and five stations per prototype and two seeds for ten. Mean scores range from 0.451 to 0.459, with no monotonic improvement as training support increases. This WEATHER-5K summary does not specify its horizon, assignment rule, K, latent source, trajectory count, sampler, or readout; it therefore does not reconstruct the evaluated prediction rules.

Table 39: WEATHER-5K training-station diagnostic scores for diffusion predictors evaluated on the stations used to train them, reported as mean $\pm$ sample standard deviation across seeds. Scores are $U_e(C_{\mathrm{net}} + D^{\mathrm{proto}}_e)$ with $n_e = 256$ and $\delta = 0.05$; they are not held-out entity certificates.

Stations/prototype	Mean score	Median score	Non-vac. %	Agreement %
2	0.459 ± 0.010	0.461 ± 0.021	100.0 ± 0.0	100.0 ± 0.0
5	0.451 ± 0.010	0.453 ± 0.014	100.0 ± 0.0	100.0 ± 0.0
10	0.459 ± 0.004	0.463 ± 0.003	100.0 ± 0.0	100.0 ± 0.0

In Theorem 5, N counts i.i.d. certified entities for a posterior-producing rule fixed independently of them, not the training stations in this diagnostic sweep or the predictive sweep in Appendix E.3.4. Neither sweep therefore tests the hierarchical dependence on N. Likewise, the responsibility weights in Corollary 3 regroup reference divergences but do not imply an automatic certificate improvement from larger prototype support.

On eICU, 92.5% of the 68,774 evaluated training stays have fewer than eight non-overlapping windows. Assigning these stays the vacuous score one gives mean/median augmented scores of $0.992/1.000$ and 7.5% non-vacuity. These summaries reflect a sample-size limitation of the reporting convention, not a guarantee for held-out patients. The available full-cohort test logging pass averages overlapping windows and cannot supply a matched full-cohort certificate; the fixed-predictor sample-size evaluation instead uses disjoint windows from the explicitly selected long-stay subset described above.

F Extended Discussion

F.1 Additional Related Work

Time-series forecasting. Transformer forecasters, including Autoformer, PatchTST, iTransformer, and Timer-XL, develop expressive temporal representations (Wu et al., 2021; Nie et al., 2023; Liu et al., 2024; 2025). Diffusion models such as TimeGrad, D3VAE, and TSDiff (Rasul et al., 2021; Li et al., 2022; Kollovieh et al., 2023), and flow-matching approaches (Lipman et al., 2023; Liu et al., 2023; Kollovieh et al., 2025), provide mechanisms for probabilistic forecast generation. CLASP complements these forecasting architectures by regularizing history-dependent component posteriors toward shared prototype priors and injecting their aggregate latent through architecture-specific conditioning interfaces.

Cross-entity sharing and partial pooling. Global and global–local forecasting methods share information through joint training, shared parameters, or decompositions into common and series-specific components (Montero-Manso & Hyndman, 2021; Wang et al., 2019; Sen et al., 2019; Salinas et al., 2020). Entity embeddings provide entity-specific vector representations, while clustering-based approaches organize sharing among related series (Cini et al., 2023; Bandara et al., 2020). The Channel Clustering Module uses assignments to learned channel prototypes to condition forecasting backbones (Chen et al., 2024). CLASP explicitly aligns context-dependent entity distributions with shared prototype priors, extending vector-based conditioning to distributional latent partial pooling. Hierarchical Bayesian modeling provides the underlying pooling principle (Gelman, 2006; Gelman et al., 2013); our contribution is its neural realization for unseen-entity forecasting.

Latent adaptation. Neural Processes adapt predictions to observed context, with latent variants representing uncertainty across related prediction tasks (Garnelo et al., 2018; Kim et al., 2019; Singh et al., 2019; Kim et al., 2022). Neural Diffusion Processes extend process modeling through diffusion (Dutordoir et al., 2023). CLASP organizes context-dependent adaptation around multiple shared prototype priors and explicitly analyzes the posterior-to-prior latent replacement at deployment.

PAC-Bayes and meta-learning. Classical PAC-Bayes bounds relate generalization to posterior–prior complexity (McAllester, 1999; Maurer, 2004), while hierarchical and meta-learning extensions study priors learned across related tasks (Pentina & Lampert, 2014; Amit & Meir, 2018; Rothfuss et al., 2021). CLASP studies the aggregate Gaussian laws used in training and deployment. Its aggregate-KL analysis is distinct from the complementary hierarchical certificate for prototype reference-prior learning, which requires an independently fixed posterior-producing rule and the stated within- and across-entity sampling assumptions.

F.2 Population Structure, Computational Cost, and Transfer Limitations

Population structure and assignments. CLASP's latent partial-pooling benefit depends on meaningful shared structure: aligning entity posteriors with common references helps only when those references capture transferable dynamics. Gains may diminish in nearly homogeneous populations, where a single global reference may suffice. Smoothly varying heterogeneity may instead favor continuous structure over a finite prototype bank. Provided clusters impose an external grouping; the WEATHER-5K latitude-based zones may pool stations with different dynamics. Inferred clusters instead depend on the learned router and its inputs, including observed histories and coordinate or diagnosis metadata, and need not recover stable or scientifically meaningful groups. In either case, inappropriate sharing can suppress useful entity-specific deviations.

Prototype count and computation. The number of prototypes K controls the granularity of sharing. Too few prototypes may merge distinct regimes, while too many may reduce effective support per prototype. Although $K = 4$ attains the lowest horizon-averaged errors among the evaluated prototype counts, eICU optima vary by horizon and metric (Appendix E.3.3); this does not establish a universal choice of K. Training evaluates a component-posterior head for each context–prototype pair. For batch size B_{batch} and fixed network widths, diagonal-posterior construction, sampling, KL evaluation, and aggregation scale as $O(B_{\text{batch}} K d_z)$ (Appendix B.5). Deployment avoids the pos-

terior heads but retains prototype sampling and aggregation, plus routing for inferred assignments. With diagonal covariances, prototype-prior storage scales as $O(Kd_z)$ independently of entity count. The complete persistent bank, including prototype embeddings, requires $O(K(d_z + d_u))$ memory. This overhead may outweigh gains when population structure is weak. Adaptive prototype selection and more efficient latent computation could better balance regime resolution, statistical support, and computational cost.

Scope of empirical attribution. The matched 2×2 model/readout experiment shows that most gains persist without the validation-fitted forecast readout in the reported WEATHER-5K flow-matching configuration (Section 4.2). A corresponding factorial decomposition of latent conditioning and readout contributions is not established for other backbones. Moreover, the $\lambda = 0$ control jointly removes posterior–prototype alignment and prototype-prior learning (Appendix E.2.1), leaving their individual contributions unresolved. These controls support the combined mechanism within the evaluated configuration, rather than an attribution to alignment alone or a decomposition that holds across all backbones.

Evidence for population transfer. The provided-cluster WEATHER-5K test set contains only 15 held-out stations, three per zone, selected deterministically by sorted station ID (Appendix D.1). This limits evidence for how reliably the learned references transfer across station populations. Small training-seed variation does not resolve uncertainty from this limited entity sample; broader entity holdouts are needed to assess pooling across diverse populations and distribution shifts.

Scope of theoretical guarantees. The aggregate-KL analysis characterizes latent train–deployment discrepancy rather than forecasting risk. The PAC-Bayes guarantees require additional bounded-loss, sampling, and sample-independence assumptions (Appendix A.2.6). They do not automatically certify the implemented window-dependent, prior-sampled predictor. The separate fixed-predictor evaluations retain their own stated risks and sampling assumptions (Appendix E.4.2).

Clinical scope. The eICU experiments evaluate forecasting methodology, not clinical decision support. Sharing latent references across patients may obscure subgroup-specific dynamics or propagate biases associated with missingness and population imbalance. Aggregate forecasting gains therefore do not establish clinical utility, patient benefit, safety, or clinically meaningful subgroup discovery. Clinical use would require external validation, subgroup evaluation, calibration assessment, and human oversight, including checks that partial pooling preserves clinically relevant patient differences.