DNS-Calibrated Local Stochastic Transition Closure for PDE-Free Long-Horizon Turbulence Diffusion

Zhuoer Lin[1] **Wenwu He**[1,2†] **Congcong Liu**[3] **Zhuo-Xu Cui**[3†]
[1]School of Computing and Data Science, Fujian University of Technology
[2]Fujian Provincial Key Laboratory of Big Data Mining and Applications
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
zhuoerlin@smail.fjut.edu.cn, hwwhbb@163.com
{cc.liu, zx.cui}@siat.ac.cn

Abstract

PDE-free latent diffusion models can synthesize plausible turbulence fields, but visually realistic snapshots do not guarantee statistically reliable long-horizon rollouts. In CNF-based turbulence diffusion, we observe that long rollouts can drift in latent transition laws, temporal memory, and transport-sensitive diagnostics even when the decoder remains spatially expressive. We address this failure mode with a DNS-calibrated stochastic transition closure for long-horizon turbulence diffusion. The closure is fit offline in a coarse latent observable space and defines stochastic transition tubes calibrated from DNS-referenced latent trajectories. During diffusion training, the frozen closure regularizes the denoiser's predicted-clean latent transitions by penalizing deviations from these calibrated tubes, while leaving the CNF decoder and reverse diffusion sampler unchanged. The method targets transition-side temporal reliability without PDE residuals or sampling-time correction. Across DNS-referenced long-horizon protocols, the proposed method yields targeted gains in temporal transport and memory diagnostics, especially in high-drift settings, with decoded-field spectra and derivative statistics serving as physical-fidelity guardrails. These results support lightweight stochastic transition closure for improving temporal reliability in PDE-free turbulence diffusion. The code is publicly available at: https://github.com/MAiTL-Group/CoNFiLD-LSKP.

1 Introduction

Unconditional turbulence generation must reproduce fine-scale spatial structure while preserving long-horizon temporal, transport, derivative-sensitive, and intermittency-related statistics [Pope, 2000, Frisch, 1995, Kolmogorov, 1941, She and Leveque, 1994]. A rollout may contain visually plausible snapshots while drifting in the transition statistics that govern temporal memory, transport, and rare events. In the PDE-free setting, governing equations, simulator residuals, pressure projections, and time-stepping corrections are unavailable as training or sampling anchors.

Recent generative models for scientific flows include direct 3D flow generation, sequence-level diffusion for 4D flows, Lagrangian turbulence diffusion, conditional turbulence diffusion, and score-based statistical fluid generation [Lienen et al., 2024, Saydemir et al., 2024, Li et al., 2024, Gao et al., 2024, Molinaro et al., 2024]. Latent diffusion separates generative modeling in a compact latent space from high-dimensional decoding [Rombach et al., 2022]. CoNFiLD applies this principle to spatiotemporal turbulence using a coordinate-based conditional neural field (CNF) decoder and a latent diffusion model [Du et al., 2024]. Related studies further support CNF latents as compact representations for turbulence fields [Liu et al., 2025, Guo et al., 2025]. However, a compact and

†Corresponding authors: Wenwu He (hwwhbb@163.com) and Zhuo-Xu Cui (zx.cui@siat.ac.cn)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

expressive latent representation does not by itself ensure transition statistics consistent with direct numerical simulation (DNS) over long generated trajectories. We therefore study how to regularize long-horizon latent transition dynamics while preserving the PDE-free sampling pipeline.

The DNS-referenced diagnostics expose temporal-memory gaps, short-lag transport errors, and tail-window mismatch in long rollouts of the original CoNFiLD backbone (Appendix Figure 2). Event-conditioned spectral diagnostics in the same figure show additional field discrepancies across calm, typical, and burst windows. These observations motivate a training-time constraint on latent transition compatibility, alongside decoded-field statistical fidelity checks.

We propose a DNS-calibrated stochastic transition closure implemented as a frozen local stochastic Koopman-type prior (LSKP). The prior is fit offline on coarse observables of CNF training latents, with fitted operators and residual scales defining stochastic transition tubes. Validation diagnostics guide the choice between a single regime and multiple descriptor regimes. During diffusion training, it penalizes whitened residuals of predicted-clean latent transitions. With multiple regimes, selective weighting reduces contributions from ambiguous assignments. A separate noise gate restricts coupling to lower diffusion noise levels. At inference, the reverse diffusion sampler and CNF decoder are unchanged. Figure 1 summarizes the motivation and the training and inference pipeline.

We evaluate targeted transition and temporal-memory improvements within the CoNFiLD turbulence benchmark family. Each case uses its own fitted latent representation and prior, with testing on held-out data from the same physical setting. Temporal transport and memory are the primary diagnostics; spectra and derivative-sensitive statistics assess decoded-field fidelity.

The paper makes three contributions:

- We identify long-horizon latent transition drift as a key failure mode of PDE-free CNF-based turbulence diffusion using DNS-referenced temporal and field diagnostics.
- We introduce a frozen DNS-calibrated stochastic transition closure for predicted-clean latent transitions during diffusion training, keeping the CNF decoder, reverse diffusion sampler, and inference budget unchanged when the LSKP prior is added.
- We provide a conditional finite-time analysis and evaluation protocol linking tube discrepancy, predicted-clean proxy consistency, finite-time amplification, short-lag transport, and decoded-field physical-fidelity guardrails across the turbulence cases.

2 Related work

Generative modeling for turbulence and scientific flows. Data-driven turbulence generation has been studied through direct 3D flow generation [Lienen et al., 2024], 4D spatiotemporal sequence generation [Saydemir et al., 2024], Lagrangian trajectory synthesis [Li et al., 2024], conditional turbulence diffusion [Gao et al., 2024], and score-based statistical fluid generation [Molinaro et al., 2024]. These works demonstrate that generative models can capture nontrivial turbulent variability. We evaluate temporal consistency, tail-window transport, derivative-sensitive statistics, and event-conditioned behavior under the same long-horizon evaluation protocol.

Conditional neural fields and latent diffusion backbones. CoNFiLD is the closest backbone because it combines a CNF decoder with latent diffusion for PDE-free turbulence generation [Du et al., 2024]. Related work studies CNF representations for inflow synthesis [Liu et al., 2025] and representation studies [Guo et al., 2025]. Building on these representations, we add a transition prior while retaining the CNF decoder and the latent diffusion architecture.

Koopman learning and structured latent dynamics. Koopman operator theory and dynamic mode decomposition seek representations where nonlinear evolution is approximately linear, enabling interpretable spectra and tractable long-horizon analysis [Tu et al., 2014, Brunton and Kutz, 2019]; see Colbrook et al. [2025] for a recent data-driven overview. Neural variants learn such representations from data, including invariant-subspace and lifted-autoencoder models [Takeishi et al., 2017, Lusch et al., 2018], temporally consistent and Hamiltonian variants [Nayak et al., 2025, Zhang et al., 2024], and recent studies of representation quality, expressiveness, and stability [Cheng et al., 2026]. Coherent-prior diffusion for chaotic dynamics can also use reduced-order or deep Koopman priors

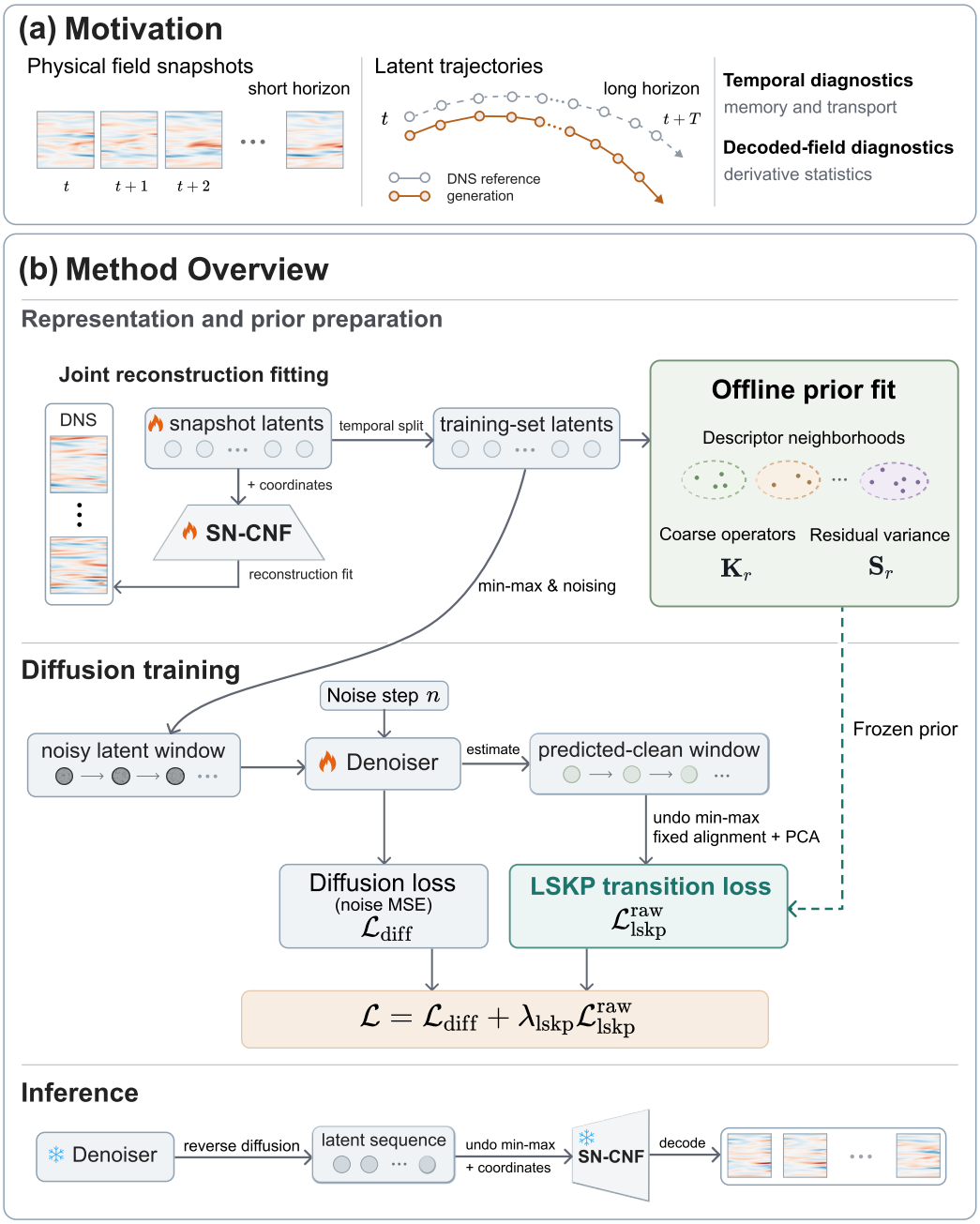

Figure 1: Motivation and method overview. (a) Plausible snapshots can accompany statistical drift in long rollouts. The trajectories are schematic; evaluation uses temporal and decoded-field diagnostics rather than pointwise trajectory matching. (b) Reconstruction jointly fits snapshot latents and a spectrally normalized CNF (SN-CNF). After temporal splitting, training-set latents supply offline prior fitting and diffusion training. Fitted operators and residual variances define a frozen prior that scores predicted-clean transitions while gradients update the denoiser. Clusters illustrate $R > 1$; the transition penalty also applies for $R = 1$. Inference retains the original reverse diffusion sampler and fixed coordinate-based decoder. Flames mark trainable components; snowflakes frozen ones.

as conditioning guidance for long-range forecasting [Nathaniel and Gentine, 2026]. Here, fitted Koopman-type operators and residual scales define a frozen scoring rule for short-lag transitions in an

existing CNF latent chart. The rule regularizes predicted-clean transitions during diffusion training, while latent diffusion remains responsible for generating the rollout.

Physics-guided diffusion and training-time structure. Diffusion and score-based models incorporate physical structure through PDE-informed losses [Bastek et al., 2025], physics-consistent sampling [Jacobsen et al., 2025], posterior-style sampling guidance [Chung et al., 2023], and physics-guided reinforcement learning for video diffusion [Deng and Peng, 2026]. LSKP places DNS-calibrated structure on predicted-clean latent transitions during PDE-free training. Sampling-time guidance changes inference compute and tuning; we isolate a training-time signal with unchanged sampling.

3 Method

LSKP augments the CoNFiLD backbone with a DNS-calibrated transition penalty during diffusion training. Spectral normalization (SN) regularizes the CNF decoder's latent-conditioning branch [Miyato et al., 2018]; the prior is fit offline on the resulting training latents. It scores predicted-clean transitions through fitted operators and residual scales in coarse observables.

3.1 Problem setup and design rationale

We are given turbulent field trajectories

$$\{\mathbf{y}^{(i)}_{0:T_i}(\cdot)\}_{i=1}^{n_{\mathrm{tr}}}, \tag{1}$$

where i labels trajectories, T_i is their final physical-time index, and n_{tr} is their number. We use t for physical time and omit (i) below. No governing equation, simulator residual, or analytic time-stepper is used during training. Following CoNFiLD [Du et al., 2024], each field snapshot $\mathbf{y}_t(\cdot) \in \mathcal{Y}$ has a latent code

$$\mathbf{x}_t \in \mathbb{R}^{d_x}, \qquad \mathbf{y}_t(\cdot) \approx \mathcal{G}(\mathbf{x}_t; \cdot). \tag{2}$$

The conditional neural field (CNF) decoder $\mathcal{G}$ is a shared coordinate-based SIREN network with sinusoidal activations. Each snapshot latent is optimized together with the decoder by alternating reconstruction updates, without an explicit encoder. The code modulates the decoder's hidden features, while spatial coordinates specify where field values are queried. The code coordinates depend on the decoder fitted jointly with them. The decoder is then fixed and the extracted codes are temporally partitioned (Appendix B.5). The training-set latents supply both offline prior fitting and diffusion training. Diffusion models the ordered latent sequence through windows

$$\mathbf{X}_{0:\ell_w} = (\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell_w}) \in \mathbb{R}^{(\ell_w+1)\times d_x}. \tag{3}$$

A window contains $\ell_w + 1$ snapshots and ℓ_w adjacent transitions. We regularize these adjacent transitions during diffusion training using a frozen DNS-calibrated prior.

Pairing a fitted transition operator with a residual scale gives a stochastic compatibility score in the inherited latent representation. Descriptor neighborhoods extend the construction to multiple operators where supported; a single regime uses the same scoring principle.

3.2 Offline LSKP construction

We use three turbulence datasets from CoNFiLD [Du et al., 2024]: Case1 is a stochastically forced irregular-pipe flow, Case2 is equilibrium turbulent inflow from a channel flow, and Case3 is non-equilibrium periodic-hill turbulence. LSKP is fit to clean SN-CNF training latents separately for each case and scores its held-out or generated transitions. Prior selection checks validation support for each regime, residual scales, operator magnitudes and whitening scales, and stable training coupling; test data are reserved for final evaluation. For a new case, start with one tube and add regimes or history while these checks remain healthy. The selected priors use eight regimes with history for Case1 and single-regime tubes for Case2/3. A single regime retains the fitted transition operator and residual whitening; Case1 additionally exercises localization, history, and selective weighting. Appendix B.5 reports the selection evidence; Tables 13 and 17 list the configurations.

The construction starts from a coarse observable

$$\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t), \qquad \mathbf{c}_t \in \mathbb{R}^{d_c}, \tag{4}$$

where $\mathcal{P}$ applies fixed training-split latent standardization followed by PCA. We keep this map fixed so that diffusion training uses stable coordinates for transition scoring. Jointly learning a lifting would also change the coordinates in which the prior is calibrated; the fixed map isolates the contribution of the transition prior. Appendix B.5 specifies the map, and Appendix B.6.1 quantifies variance retention and numerical conditioning. Local dynamics are indexed by a transition descriptor

$$\mathbf{q}_t = \mathcal{Q}(\mathbf{c}_t, \mathbf{c}_{t+1}, \ldots) \in \mathbb{R}^{d_q}, \tag{5}$$

This d_q-dimensional descriptor summarizes the transition being scored. Its base raw form is

$$\begin{aligned} \mathbf{q}_t^{\mathrm{raw}} &= \left[\mathbf{c}_t^\top,\ \mathbf{h}_t^\top,\ \delta_{e,t},\ \sqrt{\|\mathbf{h}_t\|_2^2 + \varepsilon_q}\right]^\top, \\ \mathbf{h}_t &:= \mathbf{c}_{t+1} - \mathbf{c}_t, \qquad \delta_{e,t} := \|\mathbf{c}_{t+1}\|_2^2 - \|\mathbf{c}_t\|_2^2. \end{aligned} \tag{6}$$

Here $\delta_{e,t}$ is a signed coarse-energy change, and ε_q stabilizes the increment norm. Coordinatewise standardization of the raw descriptor with fixed training statistics gives $\mathbf{q}_t$. Case1 also includes the previous increment $\mathbf{h}_{t-1}$ and the difference $\mathbf{h}_t - \mathbf{h}_{t-1}$. Appendix B.5 specifies the block order, window boundaries, dimensions, and numerical constants for both fitting and scoring.

Lloyd k-means in standardized descriptor space yields centers $\{\mathbf{v}_r\}_{r=1}^R$, where R is the number of regimes. Grouping and dynamical fitting are separate steps: Lloyd minimizes descriptor distances, and Appendix B.6.2 assesses the resulting transition fits against a global model and label controls. These regions are reproducible descriptor neighborhoods. Within each neighborhood r, we fit

$$\mathbf{c}_{t+1} \approx \mathbf{K}_r \mathbf{c}_t + \boldsymbol{\epsilon}_t^{\mathrm{fit}}, \qquad \mathbb{E}[\boldsymbol{\epsilon}_t^{\mathrm{fit}}] \approx \mathbf{0}, \quad \mathrm{Cov}(\boldsymbol{\epsilon}_t^{\mathrm{fit}}) \approx \mathbf{S}_r, \tag{7}$$

where $\mathbf{K}_r$ is a transition operator and $\mathbf{S}_r$ is the diagonal scoring matrix built from DNS-referenced residual variances, including the numerical stabilization specified in Appendix B.5. Together, $\mathbf{v}_r$, $\mathbf{K}_r$, and $\mathbf{S}_r$ define a local stochastic transition tube with mean transition $\mathbf{K}_r\mathbf{c}_t$ and residual geometry calibrated to the variability of DNS transitions in that regime. During diffusion training, whitened residual coordinates are scored within this geometry using squared or Huber penalties.

Appendix Figure 5 visualizes this mechanism: a predicted-clean descriptor query supplies the soft weights for both operator and residual variance interpolation, and the resulting whitened residual receives a coordinatewise training penalty with reliability weighting.

3.3 Training-time predicted-clean transition coupling

For diffusion, let $\mathbf{Z}_0$ denote a clean latent window after training-split min–max normalization, and let n index the diffusion noise step. With noise variances β_j, define $\bar{\alpha}_n = \prod_{j=1}^n (1-\beta_j)$. We use epsilon prediction:

$$\mathbf{Z}_n = \sqrt{\bar{\alpha}_n}\,\mathbf{Z}_0 + \sqrt{1-\bar{\alpha}_n}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{8}$$

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{\mathbf{Z}_0, n, \boldsymbol{\epsilon}}\left[\frac{\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{Z}_n, n)\|_F^2}{(\ell_w + 1) d_x}\right]. \tag{9}$$

Here n is sampled uniformly over the diffusion steps, $\boldsymbol{\epsilon}_\theta$ is the denoiser, and the Frobenius norm sums squared errors over all window entries. The predicted-clean normalized window is

$$\hat{\mathbf{Z}}_0^{(n)} = \frac{\mathbf{Z}_n - \sqrt{1-\bar{\alpha}_n}\,\boldsymbol{\epsilon}_\theta(\mathbf{Z}_n, n)}{\sqrt{\bar{\alpha}_n}}. \tag{10}$$

Undoing the diffusion normalization gives $\hat{\mathbf{X}}_{0:\ell_w}^{(n)}$. The frozen prior maps this window through its stored latent alignment and PCA to obtain $\hat{\mathbf{c}}_t$ and $\hat{\mathbf{q}}_t$, and scores the predicted-clean transitions. Writing $\alpha_r(\mathbf{q})$ for the assignment function, the training weights $\alpha_t(r) = \alpha_r(\hat{\mathbf{q}}_t)$ satisfy

$$\alpha_t(r) \propto \exp\!\big(-\tau \|\hat{\mathbf{q}}_t - \mathbf{v}_r\|_2^2\big), \qquad r = 1, \ldots, R, \tag{11}$$

with normalization over r, and the inverse temperature τ is selected during offline prior fitting. For $R > 1$, let $\boldsymbol{\alpha}_t = (\alpha_t(r))_{r=1}^R$, with $\alpha_{t,(1)}$ and $\alpha_{t,(2)}$ its two largest entries. Using the entropy $\mathrm{Ent}(\boldsymbol{\alpha}_t) = -\sum_r \alpha_t(r) \log \alpha_t(r)$ of the full assignment vector, we define

$$\rho_t = \alpha_{t,(1)}\big(\alpha_{t,(1)} - \alpha_{t,(2)}\big)\left(1 - \frac{\mathrm{Ent}(\boldsymbol{\alpha}_t)}{\log R}\right). \tag{12}$$

Confidence measures assignment ambiguity. Selective weighting combines it with latent activity; Appendix Tables 27–28 distinguish the marginal and joint diagnostics. With one regime, $\rho_t = 1$.

The softly interpolated local operator and residual scale are

$$\bar{\mathbf{K}}_t := \sum_{r=1}^{R} \alpha_t(r)\mathbf{K}_r, \qquad \bar{\mathbf{S}}_t := \sum_{r=1}^{R} \alpha_t(r)\mathbf{S}_r, \tag{13}$$

and the whitened predicted-clean residual and its scalar discrepancy are

$$\hat{\mathbf{r}}_t^{\mathrm{w}} := \bar{\mathbf{S}}_t^{-1/2}\big(\hat{\mathbf{c}}_{t+1} - \bar{\mathbf{K}}_t\hat{\mathbf{c}}_t\big), \qquad \hat{d}_t := \|\hat{\mathbf{r}}_t^{\mathrm{w}}\|_2. \tag{14}$$

The final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{lskp}}\,\mathcal{A}\big(w_t\,\varphi_t(\hat{\mathbf{r}}_t^{\mathrm{w}})\big), \tag{15}$$

where the superscript w denotes whitening. The penalty φ_t acts coordinatewise, and $\mathcal{A}$ takes the arithmetic mean over residual coordinates, adjacent transitions, and selected windows. Define the activity proxy $b_t = \sqrt{\|\hat{\mathbf{c}}_{t+1} - \hat{\mathbf{c}}_t\|_2^2 + \varepsilon_b}$, with numerical constant ε_b, and let b_* be the stored training-split 75th-percentile activity threshold. The final runs use

$$w_t = 0.65^{\mathbb{I}[\rho_t < 0.15]}\,0.75^{\mathbb{I}[b_t \geq b_* \,\wedge\, \rho_t < 0.25]}, \tag{16}$$

where $\mathbb{I}$ is the indicator function. Either trigger also switches the transition from squared loss to componentwise Huber loss. With one regime, $w_t = 1$ and the squared-loss branch remains active.

Writing $\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}} = \mathcal{A}(w_t\varphi_t(\hat{\mathbf{r}}_t^{\mathrm{w}}))$, the applied coefficient on eligible updates with positive prior loss is

$$\lambda_{\mathrm{lskp}} = \min\left(\lambda_{\mathrm{base}}, \frac{r_{\mathrm{cap}}\ \mathrm{sg}(\mathcal{L}_{\mathrm{diff}})}{\mathrm{sg}(\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}}) + \varepsilon_{\mathrm{cap}}}\right), \tag{17}$$

where sg stops gradients and $\varepsilon_{\mathrm{cap}} > 0$ stabilizes the ratio. Thus the coefficient is fixed during backpropagation, while residuals and soft assignments remain differentiable with respect to the predicted-clean window. Coupling begins after a case-specific warmup, every fourth optimization step, using at most four windows satisfying $\sigma_n = \sqrt{1 - \bar{\alpha}_n} \leq 0.60$. Otherwise, or if the prior loss is zero, the LSKP contribution is zero. Appendix B.5 gives the coordinatewise penalty and update-selection rules, including the numerical constants for each case.

3.4 Training and inference pipeline

Appendix Algorithm 1 gives the full pipeline: train SN-CNF and fit the prior on the resulting clean training latents, then freeze both while training latent diffusion with LSKP. At inference, the standard reverse diffusion sampler generates latents, which are restored to the CNF code scale before coordinate-based decoding. No PDE supervision or sampling-time projection, correction, or guidance is used. Field-level spectra and derivative-sensitive quantities remain downstream fidelity guardrails.

4 Analysis

We analyze three links between the training objective and temporal diagnostics: predicted-clean accuracy controls the scoring proxy, tube discrepancy bounds local transition residuals, and finite-time operator products propagate those residuals. Under the stated coupling and regularity conditions, trajectory discrepancies yield bounds on diagnostic distributions. Appendix A gives the assumptions, formal statements, proofs, and their empirical interpretation for the evaluated priors.

The analysis separates the CNF decoder $\mathcal{G}$, snapshot codes $\mathbf{x}_t$, and coarse transition coordinates $(\mathbf{c}_t, \mathbf{q}_t)$. SN controls weight scales in the decoder's latent-conditioning branch; LSKP scores transitions after the fixed coarse map $\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t)$. Decoder regularity and transition compatibility therefore play distinct roles. Field-space bounds require local chart assumptions in Proposition A.8.

4.1 Local stochastic transition closure

Let $\mathbf{x}_t \in \mathbb{R}^{d_x}$ denote the snapshot code in Eq. (2), and let

$$\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t), \qquad \mathbf{q}_t = \mathcal{Q}(\mathbf{c}_t, \mathbf{c}_{t+1}, \mathbf{c}_{t-1}, \ldots) \tag{18}$$

be the coarse observable and transition descriptor. For a descriptor $\mathbf{q}$, LSKP assigns soft regime weights $\{\alpha_r(\mathbf{q})\}_{r=1}^R$ and interpolates local transition operators and residual scales:

$$\bar{\mathbf{K}}(\mathbf{q}) = \sum_{r=1}^{R} \alpha_r(\mathbf{q})\mathbf{K}_r, \qquad \bar{\mathbf{S}}(\mathbf{q}) = \sum_{r=1}^{R} \alpha_r(\mathbf{q})\mathbf{S}_r. \tag{19}$$

The training quantities $\bar{\mathbf{K}}_t$ and $\bar{\mathbf{S}}_t$ evaluate these maps at $\hat{\mathbf{q}}_t$; clean transitions use $\mathbf{q}_t$. We use this interpolated residual scale as the covariance of a scoring kernel, not as the exact covariance of a Gaussian mixture. These quantities define a descriptor-conditioned stochastic transition kernel

$$\mathcal{T}_{\mathbf{q}}(\cdot \mid \mathbf{c}) = \mathcal{N}\big(\bar{\mathbf{K}}(\mathbf{q})\mathbf{c}, \bar{\mathbf{S}}(\mathbf{q})\big)\,. \tag{20}$$

This kernel has a local Koopman-type interpretation on coarse coordinate observables, because its conditional mean is linear in $\mathbf{c}$ for fixed $\mathbf{q}$. Here the kernel scores short-lag transition compatibility during training; latent diffusion generates the complete latent rollout.

The corresponding whitened transition energy is

$$d_t := \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \left(\mathbf{c}_{t+1} - \bar{\mathbf{K}}(\mathbf{q}_t)\mathbf{c}_t\right) \right\|_2. \tag{21}$$

The scalar d_t measures departure from the fitted mean transition in DNS-calibrated residual units. The training objective in Eq. (15) applies squared or Huber penalties to its whitened residual coordinates. The same analysis covers a single regime, using one fitted operator and residual scale.

4.2 Predicted-clean tube energy as a training-time proxy

During diffusion training, the prior is evaluated on the denoiser's predicted-clean latent window rather than on the noisy latent state. Let $\hat{d}_t$ denote the whitened transition energy computed from $\hat{\mathbf{x}}_t$ and the induced $\hat{\mathbf{c}}_t, \hat{\mathbf{q}}_t$. Under Assumptions A.1 and A.2, Lemma A.3 gives

$$|\hat{d}_t - d_t| \le c_{\mathrm{proxy}} \left(\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2 + \|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_2 + \delta_{\mathrm{hist},t}\right), \tag{22}$$

where $\delta_{\mathrm{hist},t}$ collects prediction errors in any history terms used by the descriptor. The diffusion parameterization gives a concrete reason for the noise gate. Subtracting the clean normalized window from Eq. (10) yields

$$\hat{\mathbf{Z}}_0^{(n)} - \mathbf{Z}_0 = \sqrt{\frac{1-\bar{\alpha}_n}{\bar{\alpha}_n}}\left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{Z}_n, n)\right). \tag{23}$$

The factor increases with the noise level; fixed affine de-normalization preserves this dependence in the CNF coordinates. The noise gate therefore limits amplification of prediction error. Reliability weights separately address ambiguous regime membership in multi-regime scoring.

The same tube energy controls one-step coarse transition deviation. Define

$$\boldsymbol{\zeta}_t = \mathbf{c}_{t+1} - \bar{\mathbf{K}}(\mathbf{q}_t)\mathbf{c}_t. \tag{24}$$

If $d_t \le \epsilon$ and

$$\left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{1/2} \right\|_2 \le \sigma_+, \tag{25}$$

then Lemma A.4 gives

$$\|\boldsymbol{\zeta}_t\|_2 \le \sigma_+ \epsilon. \tag{26}$$

Thus tube discrepancy bounds the coarse transition residual at its DNS-calibrated scale.

4.3 Finite-time amplification and diagnostic consequences

Along an evaluated coarse trajectory, fix its descriptor sequence and let

$$\mathbf{A}_s = \bar{\mathbf{K}}(\mathbf{q}_s), \qquad g_{a:b} = \|\mathbf{A}_{a-1}\mathbf{A}_{a-2}\cdots\mathbf{A}_b\|_2 \tag{27}$$

denote the finite-time amplification, with $g_{b:b} = 1$ for an empty product. Compare this trajectory with the corresponding mean path $\bar{\mathbf{c}}_{s+1} = \mathbf{A}_s\bar{\mathbf{c}}_s$ and define $\mathbf{e}_s = \mathbf{c}_s - \bar{\mathbf{c}}_s$. If tube discrepancy is bounded by ϵ and residual scale by σ_+ over the window, Theorem A.5 gives

$$\|\mathbf{e}_{t+h}\|_2 \le g_{t+h:t}\|\mathbf{e}_t\|_2 + \sigma_+\epsilon \sum_{j=0}^{h-1} g_{t+h:t+j+1}. \tag{28}$$

Ordered products govern accumulation even when individual operators are expansive. Appendix B.4 therefore examines finite-product norms, residual scales, and regime support.

A separate coupling connects reference and generated trajectory laws. Suppose their coarse endpoints have expected discrepancies bounded by δ_0 and δ_ℓ under that coupling. For an $\ell_{\mathcal{H},\ell}$-Lipschitz lag-ℓ feature map $\mathcal{H}_\ell$ with feature laws μ_ℓ and ν_ℓ, Proposition A.6 gives

$$\mathcal{W}_1(\mu_\ell, \nu_\ell) \le \ell_{\mathcal{H},\ell}(\delta_0 + \delta_\ell), \tag{29}$$

Under these coupling and regularity conditions, endpoint bounds control feature-law discrepancy. Lemma A.7 gives the analogous result for Lipschitz finite-window diagnostics. Decoder regularity links these bounds to field-space discrepancies and the spectral and derivative guardrails.

Finally, event-conditioned diagnostics require adequate event coverage. Let μ and $\tilde{\mu}$ be reference and generated feature laws, with event probabilities p and $\tilde{p}$ for $\mathcal{E}$ and conditional laws denoted by subscripts. If $b_{\max}$ bounds the feature diameter, Proposition A.9 gives

$$\mathcal{W}_1(\mu, \tilde{\mu}) \le p\,\mathcal{W}_1(\mu_{\mathcal{E}}, \tilde{\mu}_{\mathcal{E}}) + (1-p)\,\mathcal{W}_1(\mu_{\mathcal{E}^c}, \tilde{\mu}_{\mathcal{E}^c}) + b_{\max}|p - \tilde{p}|. \tag{30}$$

This decomposition separates within-event transition mismatch from event-mass error, motivating joint reporting of conditional diagnostic values, event counts, and occupancy.

5 Experiments

The experiments ask three questions: whether the original backbone exhibits DNS-referenced long-horizon transition drift; whether spectral normalization alone removes this drift; and whether LSKP provides additional transition-side benefit without sampling-time correction or decoded-field collapse.

We compare three groups. *Origin-CoNFiLD* is the original PDE-free CNF-based latent diffusion backbone. *SN-CNF* applies spectral normalization to the decoder's latent-conditioning branch, without a transition prior. *LSKP-final* adds the stochastic transition closure on the SN-stabilized chart. The comparisons distinguish chart regularization from the transition-prior objective.

5.1 Experimental setup

We use the three CoNFiLD turbulence datasets [Du et al., 2024] identified in Section 3 for held-out generation within each physical setting. Diffusion training and prior fitting use training data; validation screens priors and selects checkpoints; held-out fields provide the DNS reference. Appendix B.5 specifies the source arrays, preprocessing, and temporal partitions. For Case1/2/3, training windows contain 128/256/256 frames, and complete rollouts contain 3840/768/1536 frames. The main comparison uses rollout seed 42, giving 1/3/1 generated trajectories and 8/24/19 evaluation windows of length 1473/280/288. Appendix intervals pool rollout seeds 42/43/44, giving 3/9/3 trajectories and 24/72/57 windows. These are generation seeds, distinct from diffusion-training seeds. Appendix B.11 gives window strides and pooling rules, and distinguishes aggregate estimates from window summaries.

Our primary protocol, Protocol-C+, measures temporal transport and memory alongside decoded-field guardrails. Let $\boldsymbol{\psi}_t$ contain kinetic energy, 16 spectral bins, four derivative statistics, and two scalar summaries, each standardized over time within its evaluation window. At lag ℓ, define

$$D_\ell(t) = \|\boldsymbol{\psi}_{t+\ell} - \boldsymbol{\psi}_t\|_2, \qquad \mathcal{W}_1^{\mathrm{time}}(\ell) = \mathcal{W}_1(P^\ell_{\mathrm{DNS}}, P^\ell_{\mathrm{gen}}), \tag{31}$$

where the two empirical laws pool these scalar distances within DNS and generated windows, respectively. We report $\ell = 1, 16$, with a nominal budget of 4096 random temporal pairs divided across windows in each pool. Appendix B.1 specifies the features and integer allocation.

For memory, we use kinetic energy $E_t = \frac{1}{2}\langle\|\mathbf{u}_t\|_2^2\rangle_{\mathrm{space}}$, where $\mathbf{u}_t$ is the velocity field. We average the normalized energy autocorrelation curves over windows, then sum from lag zero through lag 128 or up to, but excluding, the first nonpositive lag. The resulting integral time, in snapshot intervals, gives $\Delta T_{\mathrm{int}} = |T^{\mathrm{gen}}_{\mathrm{int}} - T^{\mathrm{DNS}}_{\mathrm{int}}|/(|T^{\mathrm{DNS}}_{\mathrm{int}}| + 10^{-8})$. These two transport diagnostics and the memory gap are the primary readouts. Log-spectrum and derivative gaps check decoded-field fidelity. Protocol-E separates calm, typical, and burst windows using DNS event-score quantiles; Protocol-M collects fixed-checkpoint and horizon sensitivities. Late-tail uses each window's final quarter; Protocol-E reports event counts and occupancy. Appendix B provides the complete definitions and settings.

Table 1: Main comparison under the frozen validation policy, using aggregate estimates from rollout seed 42. We report two temporal-transport diagnostics, $\mathcal{W}_1^{\text{time}}(\ell{=}1)$ and $\mathcal{W}_1^{\text{time}}(\ell{=}16)$, one temporal-memory diagnostic, ΔT_{int}, and one decoded-field guardrail, $\text{Err}_{\log E, L_2}$. Lower is better; bold marks the best entry within each case and diagnostic. Annotations show relative change from Origin-CoNFiLD, $(m_{\text{Origin}} - m_{\text{Model}})/m_{\text{Origin}} \times 100\%$, with red/green denoting reduction/increase.

Case	Model	$\mathcal{W}_1^{\text{time}}(\ell{=}1)$	$\mathcal{W}_1^{\text{time}}(\ell{=}16)$	ΔT_{int}	$\text{Err}_{\log E, L_2}$
Case1	Origin-CoNFiLD	**0.0314**	0.356	0.0532	0.0971
	SN-CNF	0.0381 -21.3%	**0.306** +14.0%	0.0938 -76.3%	0.120 -23.6%
	LSKP-final	0.0898 -186.0%	0.402 -12.9%	**0.0478** +10.2%	**0.0956** +1.5%
Case2	Origin-CoNFiLD	0.551	0.664	0.883	0.482
	SN-CNF	0.622 -12.9%	0.519 +21.8%	**0.843** +4.5%	**0.424** +12.0%
	LSKP-final	**0.524** +4.9%	**0.490** +26.2%	0.859 +2.7%	0.429 +11.0%
Case3	Origin-CoNFiLD	5.80	3.34	0.978	0.536
	SN-CNF	5.65 +2.6%	3.17 +5.1%	0.979 -0.1%	0.478 +10.8%
	LSKP-final	**0.0426** +99.3%	**0.494** +85.2%	**0.383** +60.8%	**0.177** +67.0%

For the main comparison, checkpoints are selected within each training run by held-out diffusion loss, independently of test diagnostics. We retain the validation signal from the CoNFiLD training setup and apply a fixed smoothing rule. Validation cadences follow the model recipes; Appendix B.5 specifies the rule, selected steps, and complete training trajectories.

Before evaluating LSKP, Appendix Figure 2 diagnoses the original backbone failure classes exposed by the long-horizon protocol, including temporal-consistency mismatch, tail-window transport mismatch, short-lag temporal transport error, and event-conditioned spectral degradation. Appendix Table 3 shows that SN-CNF improves several chart-sensitive diagnostics. SN-CNF is the direct comparator on the shared regularized chart; LSKP-final adds the DNS-calibrated transition penalty.

5.2 Main results: transition closure improves targeted drift diagnostics

Table 1 summarizes the frozen-checkpoint comparison. The largest transition reductions occur in the evaluated cases with the largest backbone errors; gains elsewhere depend on the diagnostic.

Case3 shows the clearest reduction in backbone transition drift. Relative to SN-CNF in Table 1, LSKP-final reduces $\mathcal{W}_1^{\text{time}}(\ell{=}1)$ by more than two orders of magnitude, $\mathcal{W}_1^{\text{time}}(\ell{=}16)$ by 84.4%, and ΔT_{int} by 60.9%. The log-spectrum guardrail improves by 63.0%. These diagnostics show a strong transition and memory benefit alongside improved spectral agreement.

Across three diffusion-training seeds with the CNF chart and fitted prior fixed, LSKP-final is lower than SN-CNF on all four Case3 diagnostics (Appendix B.7.1). At horizon 1536, lag-16 transport is 2.5604 ± 1.2772 for SN-CNF and 0.4989 ± 0.0248 for LSKP-final. The corresponding memory gaps are 0.8378 ± 0.2416 and 0.2762 ± 0.0196. These are means and sample standard deviations of run-level window means, using rollout seeds 42/43/44 for each trained model.

A separate common-step comparison at 50k, 100k, and 155k shows that the ordering varies with training stage: LSKP-final improves all four diagnostics over SN-CNF at 100k and both fixed-lag diagnostics at 155k (Appendix B.7.2). The seeds support repeatability under the validation rule; the grid separately measures how the diagnostic ordering changes across diffusion training.

Case2 shows a smaller transition-side benefit. LSKP-final improves lag-1 and lag-16 transport over SN-CNF by 15.8% and 5.6% in the main comparison; memory and log-spectrum errors are slightly higher. This separates the additional transition signal from the effects of chart stabilization.

Case1 combines improved temporal-memory consistency with higher errors at both fixed lags relative to SN-CNF. Its log-spectrum aggregate also improves in Table 1. Appendix B.14 pools different rollouts and uses window summaries, so its spectral ranking is interpreted separately.

Appendix Figure 6 traces representative short-lag transport, longer-lag transport, memory, and spectral diagnostics across rollout horizons. Case3 retains its transition and memory separation across the reported horizons, while Case1/2 show diagnostic-dependent ordering. Appendix Table 35 gives the full horizon grid, and Figure 7 gives window means and intervals for the representative diagnostics.

Appendix B.12 examines similarity to training windows and longer horizons with fixed models. Qualitative Case2/3 field views in Appendix B.15 complement the quantitative guardrails.

Appendix Figure 9 separates Case1 within-event log-spectrum error from event occupancy. Appendix B.13 specifies the subwindows, DNS-derived thresholds, and coverage counts.

5.3 Mechanism checks, ablations, and cost

Appendix B.4 relates generated tube discrepancy to primary temporal diagnostics and examines finite-time amplification and event coverage. Generated LSKP-final descriptors exceed the training 99th-percentile nearest-center distance in fewer than 1.5% of transitions in each case (Appendix Table 6). Tube and diagnostic changes agree in direction in 37 of 42 comparisons across cases, horizons, and diagnostics, including 24 joint improvements and 13 joint worsenings; within-case trends are mixed. These explanatory checks do not select checkpoints.

The component analysis in Appendix Table 25 tests the local, history, and selective extensions in Case1. Removing them individually increases the temporal-memory gap from 0.0478 to 0.0603, 0.141, and 0.0595, respectively; spectral ordering varies across ablations. Case2/3 test the shared single-tube core, with additional independently trained runs in Appendix Table 26.

The 2-by-2 chart/prior diagnostic separates the transition prior from SN (Appendix B.8). Adding LSKP reduces Case3 lag-16 transport from 3.334 to 0.420 on the original chart, and from 3.168 to 0.424 on the SN chart. These are pooled aggregate estimates over rollout seeds 42/43/44. The prior therefore contributes on both charts in this comparison, with mixed Case1/2 outcomes.

Matched-budget Case1 velocity and acceleration controls give memory gaps of 0.160 and 0.210, compared with 0.136 for LSKP-final under the same pooled window-summary protocol. Acceleration improves lag-1 transport, so the distinction concerns the joint diagnostic pattern (Appendix B.9.2). In Case2/3, input gradients of the fitted single-tube objective differ from those of velocity and acceleration penalties on the same validation windows (Appendix B.9.1).

Table 32 records SN-CNF/LSKP-final costs of 13.38/13.27, 26.01/26.21, and 24.96/30.50 GPU-hours per 100k steps for Case1/2/3. Profiled peak reserved memory is unchanged within each case. LSKP adds no inference-time prior evaluation to the standard sampler. Appendix B.10 gives profiling and accounting procedures and the device configurations for the additional Case3 diffusion runs.

6 Discussion

The results support DNS-calibrated transition scoring in the evaluated CoNFiLD settings. The factorial links Case3 transition reductions to the prior on either chart; SN supplies the regularized decoder chart used by the final method. The single-tube results support the shared stochastic core, while Case1 tests the local, history, and selective extensions. Gains vary with the case and diagnostic: Case3 shows the clearest reductions under severe transition drift, Case2 improves fixed-lag transport, and Case1 trades short-lag accuracy for improved memory. Spectral and derivative statistics monitor decoded-field fidelity; event-conditioned diagnostics require adequate event coverage.

Case1's regime-count sweep favors different values for different diagnostics, motivating adaptive prior selection. Further directions include uncertainty from refitting the CNF chart and prior, and combining transition closure with sampling-time correction while retaining PDE-free generation. Other latent generators and independent turbulence benchmarks provide further tests of the closure.

Acknowledgments

The work is supported by the Natural Science Foundation of Fujian Province of China (Grant No.2024J01158), the National Natural Science Foundation of China (62476268), and the Shenzhen Science and Technology Program (Grant No.JCYJ20240813155840052).

References

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M. Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tpYeermigp.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, Cambridge, UK, 2019.

Xiaoyuan Cheng, Wenxuan Yuan, Yiming Yang, Yuanzhao Zhang, Sibo Cheng, Yi He, and Zhuo Sun. Information shapes koopman representation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Szh0ELyQxL. Oral presentation.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Matthew J. Colbrook, Zlatko Drmač, and Andrew Horning. An introductory guide to koopman learning, 2025. URL https://arxiv.org/abs/2510.22002.

Zijun Deng and Yuxin Peng. NS-Diff: Fluid Navier–Stokes guided video diffusion via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://cvpr.thecvf.com/virtual/2026/poster/36346.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15(1):10416, 2024. doi: 10.1038/s41467-024-54712-1. URL https://www.nature.com/articles/s41467-024-54712-1.

Uriel Frisch. *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press, Cambridge, UK, 1995. ISBN 9780521457132.

Han Gao, Xu Han, Xiantao Fan, Luning Sun, Li-Ping Liu, Lian Duan, and Jian-Xun Wang. Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation. *Computer Methods in Applied Mechanics and Engineering*, 427:117023, 2024. doi: 10.1016/j.cma.2024.117023. URL https://www.sciencedirect.com/science/article/pii/S0045782524002792.

Junyi Guo, Pan Du, Xiantao Fan, Yahui Li, and Jian-Xun Wang. Conditional neural field for spatial dimension reduction of turbulence data: A comparison study, 2025. URL https://arxiv.org/abs/2510.25135.

Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. Cocogen: Physically consistent and conditioned score-based generative models for forward and inverse problems. *SIAM Journal on Scientific Computing*, 47(2):C399–C425, 2025. doi: 10.1137/24M1636071. URL https://epubs.siam.org/doi/10.1137/24M1636071.

A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *C. R. Acad. Sci. USSR*, 30:301–305, 1941.

T. Li, L. Biferale, F. Bonaccorso, M. A. Scarpolini, and M. Buzzicotti. Synthetic lagrangian turbulence by generative diffusion models. *Nature Machine Intelligence*, 6(4):393–403, 2024. doi: 10.1038/s42256-024-00810-0. URL https://www.nature.com/articles/s42256-024-00810-0.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3d flow simulation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZhlwoC1XaN.

Xin-Yang Liu, Meet Hemant Parikh, Xiantao Fan, Pan Du, Qing Wang, Yi-Fan Chen, and Jian-Xun Wang. CoNFiLD-inlet: Synthetic turbulence inflow using generative latent diffusion models with neural fields. *Physical Review Fluids*, 10:054901, 2025. doi: 10.1103/PhysRevFluids.10.054901. URL https://journals.aps.org/prfluids/abstract/10.1103/PhysRevFluids.10.054901.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018. doi: 10.1038/s41467-018-07210-0. URL https://www.nature.com/articles/s41467-018-07210-0.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Roberto Molinaro, Samuel Lanthaler, Bogdan Raonić, Tobias Rohner, Victor Armegioiu, Stephan Simonis, Dana Grund, Yannick Ramic, Zhong Yi Wan, Fei Sha, Siddhartha Mishra, and Leonardo Zepeda-Núñez. Generative ai for fast and accurate statistical computation of fluids, 2024. URL https://arxiv.org/abs/2409.18359.

Juan Nathaniel and Pierre Gentine. Generative emulation of chaotic dynamics with coherent prior. *Computer Methods in Applied Mechanics and Engineering*, 448:118410, 2026. doi: 10.1016/j.cma.2025.118410. URL https://www.sciencedirect.com/science/article/pii/S0045782525006826.

Indranil Nayak, Ananda Chakrabarti, Mrinal Kumar, Fernando L. Teixeira, and Debdipta Goswami. Temporally consistent koopman autoencoders for forecasting dynamical systems. *Scientific Reports*, 15:22127, 2025. doi: 10.1038/s41598-025-05222-7. URL https://www.nature.com/articles/s41598-025-05222-7.

Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, UK, 2000. ISBN 9780521598866.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Abdullah Saydemir, Marten Lienen, and Stephan Günnemann. Unfolding time: Generative modeling for turbulent flows in 4d, 2024. URL https://openreview.net/forum?id=rqFLF2wwpg. ICML 2024 AI4Science Workshop poster.

Z. S. She and E. Leveque. Universal scaling laws in fully developed turbulence. *Physical Review Letters*, 72(3): 336–339, 1994.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3a835d3215755c435ef4fe9965a3f2a0-Abstract.html.

Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Learning hamiltonian neural koopman operator and simultaneously sustaining and discovering conservation laws. *Physical Review Research*, 6(1):L012031, 2024. doi: 10.1103/PhysRevResearch.6.L012031. URL https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.6.L012031.

Appendix

A Additional theoretical details **14**

A.1 Geometric interpretation 14

A.2 Operator-theoretic interpretation of the local kernel 14

A.3 Formal assumptions and statements 14

A.4 Proofs of theoretical statements 18

A.5 Assumption and diagnostic mapping 20

A.6 Scope of the current theory 21

B Additional experimental diagnostics **21**

B.1 Metric definitions and reporting roles 21

B.2 Motivation and backbone diagnostics 22

B.3 Theory-supporting prior diagnostics 23

B.4 Mechanism diagnostics: tube score, amplification, and event coverage 24

B.5 Implementation and checkpoint details 26

B.6 Observable and partition diagnostics 34

B.7 Training variability and checkpoint sensitivity 36

B.8 Factorial diagnostic: latent chart and transition prior 38

B.9 Component ablations and regularization controls 38

B.10 Training resources 42

B.11 Statistical reporting and horizon diagnostics 43

B.12 Fixed-checkpoint sensitivity checks 44

B.13 Event-conditioned decomposition 45

B.14 Physical-fidelity guardrails 46

B.15 Qualitative decoded-field sanity check 48

A Additional theoretical details

This appendix supplements Section 4 with formal proofs, scope clarifications, and the connection between theory assumptions and the empirical prior diagnostics. The notation follows the main text: lower-case bold symbols denote vectors, upper-case bold symbols denote matrices, and calligraphic symbols denote maps or operators.

A.1 Geometric interpretation

The formal statements below use the same three-level geometry as the main analysis. Physical fields live in $\mathcal{Y}$ and are accessed through the CNF decoder $\mathcal{G}$. The stable latent chart $\mathbf{x}_t$ is the space where the decoder remains expressive and where spectral normalization supports local regularity. The LSKP prior is fit in the lower-dimensional transition geometry $(\mathbf{c}_t, \mathbf{q}_t)$, where $\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t)$ and $\mathbf{q}_t = \mathcal{Q}(\cdot)$ summarize local short-lag dynamics. LSKP regularizes predicted-clean transitions against calibrated stochastic tubes in coarse geometry, while decoded-field consequences are monitored through the decoder-side guardrail arguments in Proposition A.8.

A.2 Operator-theoretic interpretation of the local kernel

For fixed $\mathbf{q}$, the stochastic transition kernel $\mathcal{T}_{\mathbf{q}}$ defines the operator

$$(\mathcal{K}_{\mathbf{q}} f)(\mathbf{c}) = \int f(\mathbf{c}')\, \mathcal{T}_{\mathbf{q}}(d\mathbf{c}' \mid \mathbf{c}) \tag{32}$$

on scalar observables. Applying this operator to each coordinate observable $\mathcal{I}_i(\mathbf{c}) = c_i$ returns the corresponding coordinate of the conditional mean under $\mathcal{T}_{\mathbf{q}}$. Since this conditional mean is $\bar{\mathbf{K}}(\mathbf{q})\mathbf{c}$, the vector of coordinate-observable images is $\bar{\mathbf{K}}(\mathbf{q})\mathbf{c}$. This is the precise sense in which the local fitted matrices represent a Koopman-type action: they close the operator on coarse coordinate observables locally and stochastically, rather than defining a global Koopman rollout model.

A.3 Formal assumptions and statements

The following assumptions and formal statements provide the complete version of the theory summarized in Section 4.

A.3.1 Stable CNF latent chart and coarse observables

Let $\mathbf{y}_t(\cdot) \in \mathcal{Y}$ denote the physical field at time t, and let

$$\mathbf{x}_t \in \mathbb{R}^{d_x} \tag{33}$$

be the optimized snapshot code introduced in Eq. (2). The decoder is denoted by

$$\mathbf{y}_t(\cdot) \approx \mathcal{G}(\mathbf{x}_t; \cdot). \tag{34}$$

The LSKP prior uses a coarse observable

$$\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t), \qquad \mathbf{c}_t \in \mathbb{R}^{d_c}, \tag{35}$$

and a transition descriptor

$$\mathbf{q}_t = \mathcal{Q}(\mathbf{c}_t, \mathbf{c}_{t+1}, \mathbf{c}_{t-1}, \ldots) \in \mathbb{R}^{d_q}. \tag{36}$$

The descriptor may include the current coarse state, finite differences, short history, or other transition-side summaries. The theory only requires local regularity on the region visited by clean and predicted-clean training windows.

Assumption A.1 (Stable latent chart)**.** *There exists a compact region $\mathcal{X} \subset \mathbb{R}^{d_x}$ containing the clean and predicted-clean latents considered during training. On $\mathcal{X}$, the decoder $\mathcal{G}$ and coarse observable map $\mathcal{P}$ are Lipschitz:*

$$\|\mathcal{G}(\mathbf{x}) - \mathcal{G}(\mathbf{x}')\|_{\mathcal{Y}} \le \ell_{\mathcal{G}} \|\mathbf{x} - \mathbf{x}'\|_2, \tag{37}$$

$$\|\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{x}')\|_2 \le \ell_{\mathcal{P}} \|\mathbf{x} - \mathbf{x}'\|_2, \tag{38}$$

for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Moreover, for each regime-restricted chart patch $\mathcal{X}_r \subset \mathcal{X}$, the coarse observable is locally identifying:

$$m_r \|\mathbf{x} - \mathbf{x}'\|_2 \leq \|\mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{x}')\|_2 \leq \ell_{\mathcal{P}} \|\mathbf{x} - \mathbf{x}'\|_2, \qquad \mathbf{x}, \mathbf{x}' \in \mathcal{X}_r, \tag{39}$$

with $m_r > 0$.

Assumption A.1 separates upper regularity from local identification. SN controls the weight scales of the decoder's latent-conditioning maps, while the fixed linear coarse map has a finite upper Lipschitz constant. The lower bound $m_r > 0$ is an additional requirement on the evaluated chart patches; neither SN nor PCA alone establishes it.

A.3.2 Local stochastic transition scoring kernel

A conventional global Koopman rollout formulation would seek one lifted coordinate system in which all evolution is linear. LSKP uses a family of descriptor-conditioned local stochastic transition-scoring kernels in the coarse latent observable space.

The descriptor $\mathbf{q}_t$ is assigned softly to R local regimes:

$$\boldsymbol{\alpha}(\mathbf{q}_t) = (\alpha_1(\mathbf{q}_t), \ldots, \alpha_R(\mathbf{q}_t)), \qquad \alpha_r(\mathbf{q}_t) \geq 0, \qquad \sum_{r=1}^{R} \alpha_r(\mathbf{q}_t) = 1. \tag{40}$$

Each regime has a local matrix $\mathbf{K}_r \in \mathbb{R}^{d_c \times d_c}$ and a diagonal residual-scale matrix $\mathbf{S}_r \in \mathbb{R}^{d_c \times d_c}$ used for whitening. The softly interpolated local mean and residual scale are

$$\bar{\mathbf{K}}(\mathbf{q}) = \sum_{r=1}^{R} \alpha_r(\mathbf{q}) \mathbf{K}_r, \qquad \bar{\mathbf{S}}(\mathbf{q}) = \sum_{r=1}^{R} \alpha_r(\mathbf{q}) \mathbf{S}_r. \tag{41}$$

Because the descriptor may be computed from the transition window itself, the following object is used as a descriptor-conditioned scoring kernel rather than as an autonomous Markov rollout kernel:

$$\mathcal{T}_{\mathbf{q}}(\cdot \mid \mathbf{c}) = \mathcal{N}\big(\bar{\mathbf{K}}(\mathbf{q})\mathbf{c}, \bar{\mathbf{S}}(\mathbf{q})\big). \tag{42}$$

For a scalar observable $f : \mathbb{R}^{d_c} \to \mathbb{R}$, this kernel induces a local Koopman-type operator

$$(\mathcal{K}_{\mathbf{q}} f)(\mathbf{c}) := \int f(\mathbf{c}')\, \mathcal{T}_{\mathbf{q}}(d\mathbf{c}' \mid \mathbf{c}). \tag{43}$$

For the coordinate observable $\mathcal{I}(\mathbf{c}) = \mathbf{c}$, the operator acts componentwise as

$$(\mathcal{K}_{\mathbf{q}} \mathcal{I})(\mathbf{c}) = \bar{\mathbf{K}}(\mathbf{q})\mathbf{c}. \tag{44}$$

Thus the fitted matrices $\mathbf{K}_r$ should be read as a local stochastic transition closure of the Koopman action on coarse coordinate observables, not as a global rollout operator.

Assumption A.2 (Regular finite-memory stochastic closure)**.** *On the compact region visited by the training windows, the maps $\mathcal{Q}$, $\boldsymbol{\alpha}$, $\bar{\mathbf{K}}$, $\bar{\mathbf{S}}^{1/2}$, and $\bar{\mathbf{S}}^{-1/2}$ are locally Lipschitz and uniformly bounded. Moreover, the clean descriptor-conditioned transition residual admits the decomposition*

$$\mathbf{c}_{t+1} = \bar{\mathbf{K}}(\mathbf{q}_t)\mathbf{c}_t + \mathbf{n}_t + \mathbf{r}_t, \tag{45}$$

where

$$\mathbb{E}[\mathbf{n}_t \mid \mathbf{q}_t] = \mathbf{0}, \qquad \mathbb{E}[\mathbf{n}_t \mathbf{n}_t^\top \mid \mathbf{q}_t] \preceq \bar{\mathbf{S}}(\mathbf{q}_t), \tag{46}$$

and the local closure error obeys

$$\left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \mathbf{r}_t \right\|_2 \leq \eta. \tag{47}$$

The associated whitened transition energy is

$$d_t := \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \left(\mathbf{c}_{t+1} - \bar{\mathbf{K}}(\mathbf{q}_t)\mathbf{c}_t \right) \right\|_2. \tag{48}$$

For fixed $\mathbf{q}_t$, $d_t^2/2$ is the Mahalanobis transition-surprise term under the local Gaussian kernel $\mathcal{T}_{\mathbf{q}_t}$, up to the log-determinant and normalization constants. The penalty therefore measures departures from the DNS-fitted mean in units of the local residual scale.

A.3.3 Predicted-clean proxy consistency

During diffusion training, the prior is evaluated on the denoiser's predicted-clean latent window. Let $\hat{\mathbf{x}}_t$ be the predicted-clean latent estimate and define

$$\hat{\mathbf{c}}_t = \mathcal{P}(\hat{\mathbf{x}}_t), \qquad \hat{\mathbf{q}}_t = \mathcal{Q}(\hat{\mathbf{c}}_t, \hat{\mathbf{c}}_{t+1}, \hat{\mathbf{c}}_{t-1}, \ldots). \tag{49}$$

The predicted-clean tube energy is

$$\hat{d}_t := \left\| \bar{\mathbf{S}}(\hat{\mathbf{q}}_t)^{-1/2} \left(\hat{\mathbf{c}}_{t+1} - \bar{\mathbf{K}}(\hat{\mathbf{q}}_t)\hat{\mathbf{c}}_t \right) \right\|_2 . \tag{50}$$

Lemma A.3 (Predicted-clean proxy consistency)**.** *Under the stable-chart and closure assumptions, there exists a constant $c_{\mathrm{proxy}} > 0$ such that*

$$|\hat{d}_t - d_t| \le c_{\mathrm{proxy}} \left(\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2 + \|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}\|_2 + \delta_{\mathrm{hist},t} \right), \tag{51}$$

where $\delta_{\mathrm{hist},t}$ denotes the predicted-clean error in any additional history terms used by $\mathcal{Q}$.

Lemma A.3 controls the scoring error caused by imperfect predicted-clean latents. Equation (23) identifies the noise-dependent amplification that motivates restricting prior queries to low-noise windows. The assignment-based weights separately reduce the influence of ambiguous transitions. Together these mechanisms address prediction accuracy and regime uncertainty at their respective stages.

A.3.4 Tube control and finite-time accumulation

Let

$$\boldsymbol{\zeta}_t := \mathbf{c}_{t+1} - \bar{\mathbf{K}}(\mathbf{q}_t)\mathbf{c}_t \tag{52}$$

be the one-step deviation around the local mean transition.

Lemma A.4 (Tube membership controls one-step deviation)**.** *If $d_t \le \epsilon$, then*

$$\|\boldsymbol{\zeta}_t\|_2 \le \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{1/2} \right\|_2 \epsilon. \tag{53}$$

If additionally

$$\sup_{\mathbf{q}} \left\| \bar{\mathbf{S}}(\mathbf{q})^{1/2} \right\|_2 \le \sigma_+, \tag{54}$$

then

$$\|\boldsymbol{\zeta}_t\|_2 \le \sigma_+ \epsilon. \tag{55}$$

To propagate local tube control beyond one step along an evaluated window, fix the corresponding descriptor sequence and define

$$\mathbf{A}_s := \bar{\mathbf{K}}(\mathbf{q}_s), \tag{56}$$

and for integers $a > b$ define the ordered product

$$\mathbf{F}_{a:b} := \mathbf{A}_{a-1}\mathbf{A}_{a-2}\cdots\mathbf{A}_b, \qquad \mathbf{F}_{b:b} := \mathbf{I}. \tag{57}$$

The finite-time amplification coefficient is

$$g_{a:b} := \|\mathbf{F}_{a:b}\|_2 . \tag{58}$$

When $g_{a:b} > 0$, define the associated finite-time amplification rate

$$\lambda_{a:b} := \frac{1}{a-b} \log g_{a:b}. \tag{59}$$

This is the finite-horizon analogue of a Lyapunov-type growth rate for the local regime-switching closure. The theory therefore does not require every one-step operator to be contractive; it requires the finite-time amplification factors relevant to the evaluated windows to remain controlled.

Theorem A.5 (Finite-time accumulation under local stochastic transition closure)**.** *Let a coarse trajectory satisfy*

$$\mathbf{c}_{s+1} = \mathbf{A}_s \mathbf{c}_s + \boldsymbol{\zeta}_s. \tag{60}$$

Let $\bar{\mathbf{c}}_s$ follow the corresponding local mean dynamics

$$\bar{\mathbf{c}}_{s+1} = \mathbf{A}_s \bar{\mathbf{c}}_s, \tag{61}$$

and define $\mathbf{e}_s = \mathbf{c}_s - \bar{\mathbf{c}}_s$. Then for any horizon $h \geq 1$,

$$\mathbf{e}_{t+h} = \mathbf{F}_{t+h:t}\mathbf{e}_t + \sum_{j=0}^{h-1} \mathbf{F}_{t+h:t+j+1}\boldsymbol{\zeta}_{t+j}, \tag{62}$$

and therefore

$$\|\mathbf{e}_{t+h}\|_2 \leq g_{t+h:t} \|\mathbf{e}_t\|_2 + \sum_{j=0}^{h-1} g_{t+h:t+j+1} \left\|\boldsymbol{\zeta}_{t+j}\right\|_2. \tag{63}$$

If $d_{t+j} \leq \epsilon$ and $\left\|\bar{\mathbf{S}}(\mathbf{q}_{t+j})^{1/2}\right\|_2 \leq \sigma_+$ for $j = 0, \ldots, h-1$, then

$$\|\mathbf{e}_{t+h}\|_2 \leq g_{t+h:t} \|\mathbf{e}_t\|_2 + \sigma_+ \epsilon \sum_{j=0}^{h-1} g_{t+h:t+j+1}. \tag{64}$$

This finite-time statement accommodates heterogeneous turbulent latent dynamics: local operator norms may exceed one while the evaluated finite-time products remain controlled. Equivalently, the relevant quantity is the finite-time rate $\lambda_{a:b}$ rather than a universal one-step contraction constant. The familiar contraction condition $\|\mathbf{A}_s\|_2 \leq \kappa < 1$ is only a sufficient special case, not the central claim. The corresponding empirical checks are regime support, operator scale, and residual scale, which are reported as theory-supporting prior diagnostics in Appendix Table 4; horizon-dependent behavior is examined separately in the long-horizon diagnostics.

A.3.5 Short-lag transport and field-level consequences

To relate trajectory control to temporal diagnostics, consider a coupling of reference and generated coarse trajectories. The mean path in Theorem A.5 uses the evaluated trajectory's fixed descriptor sequence; it is not itself the DNS reference trajectory. Let a lag-ℓ transition feature be

$$\mathbf{f}_{\ell,t} = \mathcal{H}_\ell(\mathbf{c}_t, \mathbf{c}_{t+\ell}), \qquad \tilde{\mathbf{f}}_{\ell,t} = \mathcal{H}_\ell(\tilde{\mathbf{c}}_t, \tilde{\mathbf{c}}_{t+\ell}), \tag{65}$$

where $\mathcal{H}_\ell$ is $\ell_{\mathcal{H},\ell}$-Lipschitz. Let μ_ℓ and ν_ℓ be the laws of $\mathbf{f}_{\ell,t}$ and $\tilde{\mathbf{f}}_{\ell,t}$.

Proposition A.6 (Short-lag temporal transport bound)**.** *Suppose there exists a coupling of reference and generated coarse trajectories such that*

$$\mathbb{E}\|\mathbf{c}_t - \tilde{\mathbf{c}}_t\|_2 \leq \delta_0, \qquad \mathbb{E}\|\mathbf{c}_{t+\ell} - \tilde{\mathbf{c}}_{t+\ell}\|_2 \leq \delta_\ell. \tag{66}$$

Then

$$\mathcal{W}_1(\mu_\ell, \nu_\ell) \leq \ell_{\mathcal{H},\ell}(\delta_0 + \delta_\ell). \tag{67}$$

When the endpoint discrepancy δ_ℓ is bounded by finite-time accumulation of local tube errors, the same amplification factors govern the lag-ℓ temporal transport gap.

Lemma A.7 (Diagnostic pushforward bound)**.** *More generally, let ρ_ℓ and $\tilde{\rho}_\ell$ denote the reference and generated laws of a coarse trajectory window $(\mathbf{c}_t, \ldots, \mathbf{c}_{t+\ell})$. Let*

$$\mathcal{M}_\ell : (\mathbf{c}_t, \ldots, \mathbf{c}_{t+\ell}) \mapsto \mathbf{z}_\ell \tag{68}$$

be an $\ell_{\mathcal{M}}$-Lipschitz diagnostic map, such as a lagged transition feature or finite-window memory statistic. Field-level guardrails are handled below through decoder regularity. Then

$$\mathcal{W}_1\big((\mathcal{M}_\ell)_\# \rho_\ell, (\mathcal{M}_\ell)_\# \tilde{\rho}_\ell\big) \leq \ell_{\mathcal{M}}\, \mathcal{W}_1(\rho_\ell, \tilde{\rho}_\ell). \tag{69}$$

This pushforward view transfers trajectory-law control to any diagnostic satisfying the stated Lipschitz condition. For normalized temporal features, applying the result requires control of normalization scales; integral-time diagnostics additionally require a stable first-nonpositive-lag cutoff. These are diagnostic regularity conditions, distinct from the transition closure assumptions. Event-conditioned metrics further depend on occupancy and coverage.

The decoded field consequences follow through the stable chart.

Proposition A.8 (Field-space propagation through the CNF chart)**.** *Assume Assumption A.1. If* $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}_r$*, then*

$$\|\mathcal{G}(\mathbf{x}) - \mathcal{G}(\hat{\mathbf{x}})\|_{\mathcal{Y}} \leq \frac{\ell_{\mathcal{G}}}{m_r} \|\mathcal{P}(\mathbf{x}) - \mathcal{P}(\hat{\mathbf{x}})\|_2 . \tag{70}$$

If $m_r \geq m_* > 0$ *for all evaluated local chart patches, then*

$$\|\mathcal{G}(\mathbf{x}) - \mathcal{G}(\hat{\mathbf{x}})\|_{\mathcal{Y}} \leq c_{\text{field}} \|\mathcal{P}(\mathbf{x}) - \mathcal{P}(\hat{\mathbf{x}})\|_2 , \qquad c_{\text{field}} := \frac{\ell_{\mathcal{G}}}{m_*}. \tag{71}$$

Proposition A.8 motivates decoded-field fidelity checks as guardrails. Since LSKP acts on transition geometry without adding a reconstruction branch, we monitor spectra, spectral error, and derivative-sensitive statistics alongside the transition diagnostics.

A.3.6 Coverage of event-conditioned statistics

Event-conditioned transport and intermittency analyses compare within-event feature statistics and event frequencies, including burst occupancy.

Let $\mathcal{E}$ be an event set associated with a diagnostic, such as a burst defined by physical-field activity. Let $p = \mathbb{P}_{\text{ref}}(\mathcal{E})$ and $\tilde{p} = \mathbb{P}_{\text{gen}}(\mathcal{E})$. Let $\mu_{\mathcal{E}}$ and $\tilde{\mu}_{\mathcal{E}}$ be the conditional feature distributions inside the event, and let $b_{\max}$ bound the feature diameter.

Proposition A.9 (Coverage decomposition)**.** *For any bounded feature distribution decomposed into event and non-event parts,*

$$\mathcal{W}_1(\mu, \tilde{\mu}) \leq p\, \mathcal{W}_1(\mu_{\mathcal{E}}, \tilde{\mu}_{\mathcal{E}}) + (1-p)\, \mathcal{W}_1(\mu_{\mathcal{E}^c}, \tilde{\mu}_{\mathcal{E}^c}) + b_{\max}|p - \tilde{p}|, \tag{72}$$

up to the standard convention that conditional terms are omitted when the corresponding mass is zero.

Thus, the decomposition separates within-event feature mismatch from event-frequency error. The conditional temporal bounds concern fixed-lag diagnostics such as lag-1 and $\mathcal{W}_1^{\text{time}}(\ell{=}16)$. Event-conditioned transport, burst occupancy, and high-order intermittency are assessed empirically together with event counts and occupancy gaps.

A.4 Proofs of theoretical statements

Proof of Lemma A.3.

Proof. Define the residual map

$$\mathcal{R}(\mathbf{c}, \mathbf{c}', \mathbf{q}) = \bar{\mathbf{S}}(\mathbf{q})^{-1/2} \left(\mathbf{c}' - \bar{\mathbf{K}}(\mathbf{q})\mathbf{c}\right), \tag{73}$$

where $\mathbf{q}$ is the descriptor produced by $\mathcal{Q}$ on the relevant local transition window. By Assumption A.2, this map is locally Lipschitz on the compact region of interest. Hence there exists $\ell_{\mathcal{R}} > 0$ such that

$$\|\mathcal{R}(\hat{\mathbf{c}}_t, \hat{\mathbf{c}}_{t+1}, \hat{\mathbf{q}}_t) - \mathcal{R}(\mathbf{c}_t, \mathbf{c}_{t+1}, \mathbf{q}_t)\|_2 \tag{74}$$

$$\leq \ell_{\mathcal{R}} \left(\|\hat{\mathbf{c}}_t - \mathbf{c}_t\|_2 + \|\hat{\mathbf{c}}_{t+1} - \mathbf{c}_{t+1}\|_2 + \delta_{\text{hist},t}\right). \tag{75}$$

The reverse triangle inequality gives

$$|\hat{d}_t - d_t| \leq \|\mathcal{R}(\hat{\mathbf{c}}_t, \hat{\mathbf{c}}_{t+1}, \hat{\mathbf{q}}_t) - \mathcal{R}(\mathbf{c}_t, \mathbf{c}_{t+1}, \mathbf{q}_t)\|_2 . \tag{76}$$

Finally, the Lipschitz property of $\mathcal{P}$ from Assumption A.1 converts latent errors into coarse errors:

$$\|\hat{\mathbf{c}}_s - \mathbf{c}_s\|_2 \leq \ell_{\mathcal{P}} \|\hat{\mathbf{x}}_s - \mathbf{x}_s\|_2 , \qquad s \in \{t, t+1\}. \tag{77}$$

Absorbing all local Lipschitz and compactness constants into c_{proxy} proves the stated bound. □

Proof of Lemma A.4.

Proof. By definition,

$$d_t = \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \boldsymbol{\zeta}_t \right\|_2 . \tag{78}$$

Since $\bar{\mathbf{S}}(\mathbf{q}_t)$ is positive definite,

$$\boldsymbol{\zeta}_t = \bar{\mathbf{S}}(\mathbf{q}_t)^{1/2} \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \boldsymbol{\zeta}_t. \tag{79}$$

Therefore,

$$\|\boldsymbol{\zeta}_t\|_2 \le \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{1/2} \right\|_2 \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{-1/2} \boldsymbol{\zeta}_t \right\|_2 = \left\| \bar{\mathbf{S}}(\mathbf{q}_t)^{1/2} \right\|_2 d_t. \tag{80}$$

If $d_t \le \epsilon$, this gives the first claim. The uniform bound follows from the assumption that

$$\sup_{\mathbf{q}} \left\| \bar{\mathbf{S}}(\mathbf{q})^{1/2} \right\|_2 \le \sigma_+. \tag{81}$$

□

Proof of Theorem A.5.

Proof. From

$$\mathbf{c}_{s+1} = \mathbf{A}_s \mathbf{c}_s + \boldsymbol{\zeta}_s, \qquad \bar{\mathbf{c}}_{s+1} = \mathbf{A}_s \bar{\mathbf{c}}_s, \tag{82}$$

we obtain

$$\mathbf{e}_{s+1} = \mathbf{A}_s \mathbf{e}_s + \boldsymbol{\zeta}_s. \tag{83}$$

Unrolling this recursion for h steps gives

$$\mathbf{e}_{t+h} = \mathbf{F}_{t+h:t} \mathbf{e}_t + \sum_{j=0}^{h-1} \mathbf{F}_{t+h:t+j+1} \boldsymbol{\zeta}_{t+j}. \tag{84}$$

Taking Euclidean norms and applying the triangle inequality gives

$$\|\mathbf{e}_{t+h}\|_2 \le g_{t+h:t} \|\mathbf{e}_t\|_2 + \sum_{j=0}^{h-1} g_{t+h:t+j+1} \left\|\boldsymbol{\zeta}_{t+j}\right\|_2 . \tag{85}$$

The final inequality follows by applying Lemma A.4 to each $\boldsymbol{\zeta}_{t+j}$. □

Contraction as a sufficient special case. If $\|\mathbf{A}_s\|_2 \le \kappa < 1$ for all s, then

$$g_{t+h:t+j+1} \le \kappa^{h-1-j}. \tag{86}$$

Theorem A.5 then reduces to

$$\|\mathbf{e}_{t+h}\|_2 \le \kappa^h \|\mathbf{e}_t\|_2 + \sigma_+ \epsilon \sum_{j=0}^{h-1} \kappa^{h-1-j}. \tag{87}$$

This contraction statement is a useful sufficient condition, but the main theory uses the more general finite-time amplification bound.

The same product norm can be expressed through the finite-time rate

$$\lambda_{a:b} = \frac{1}{a-b} \log g_{a:b}. \tag{88}$$

Positive values indicate local finite-window amplification, negative values indicate contraction over that window, and near-zero values indicate neutral finite-window transport. Operator-scale and horizon-sensitivity diagnostics assess whether finite-time amplification permits local tube control to propagate beyond one step.

Proof of Proposition A.6.

Proof. For any chosen coupling, the primal definition of $\mathcal{W}_1$ gives

$$\mathcal{W}_1(\mu_\ell, \nu_\ell) \le \mathbb{E}\,\|\mathcal{H}_\ell(\mathbf{c}_t, \mathbf{c}_{t+\ell}) - \mathcal{H}_\ell(\tilde{\mathbf{c}}_t, \tilde{\mathbf{c}}_{t+\ell})\|_2 \tag{89}$$

$$\le \ell_{\mathcal{H},\ell}\mathbb{E}\,(\|\mathbf{c}_t - \tilde{\mathbf{c}}_t\|_2 + \|\mathbf{c}_{t+\ell} - \tilde{\mathbf{c}}_{t+\ell}\|_2) \tag{90}$$

$$\le \ell_{\mathcal{H},\ell}(\delta_0 + \delta_\ell). \tag{91}$$

□

Proof of Lemma A.7.

Proof. Let π be any coupling of ρ_ℓ and $\tilde{\rho}_\ell$. Pushing coupled samples through $\mathcal{M}_\ell$ gives a valid coupling of $(\mathcal{M}_\ell)_\#\rho_\ell$ and $(\mathcal{M}_\ell)_\#\tilde{\rho}_\ell$. Therefore,

$$\mathcal{W}_1\big((\mathcal{M}_\ell)_\#\rho_\ell, (\mathcal{M}_\ell)_\#\tilde{\rho}_\ell\big) \le \int \|\mathcal{M}_\ell(\mathbf{w}) - \mathcal{M}_\ell(\tilde{\mathbf{w}})\|_2 \, d\pi(\mathbf{w}, \tilde{\mathbf{w}}) \tag{92}$$

$$\le \ell_{\mathcal{M}} \int \|\mathbf{w} - \tilde{\mathbf{w}}\|_2 \, d\pi(\mathbf{w}, \tilde{\mathbf{w}}). \tag{93}$$

Taking the infimum over all couplings π yields the stated bound. □

Proof of Proposition A.8.

Proof. Fix a local chart patch $\mathcal{X}_r$. Decoder Lipschitzness gives

$$\|\mathcal{G}(\mathbf{x}) - \mathcal{G}(\hat{\mathbf{x}})\|_{\mathcal{Y}} \le \ell_{\mathcal{G}} \|\mathbf{x} - \hat{\mathbf{x}}\|_2 . \tag{94}$$

The local lower Lipschitz bound for $\mathcal{P}$ gives

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \le m_r^{-1} \|\mathcal{P}(\mathbf{x}) - \mathcal{P}(\hat{\mathbf{x}})\|_2 . \tag{95}$$

Combining the two inequalities proves the first statement. The uniform version follows by replacing m_r with m_*. □

Proof of Proposition A.9.

Proof. Write

$$\mu = p\mu_{\mathcal{E}} + (1-p)\mu_{\mathcal{E}^c}, \qquad \tilde{\mu} = \tilde{p}\tilde{\mu}_{\mathcal{E}} + (1-\tilde{p})\tilde{\mu}_{\mathcal{E}^c}. \tag{96}$$

Introduce the intermediate distribution

$$\mu' = p\tilde{\mu}_{\mathcal{E}} + (1-p)\tilde{\mu}_{\mathcal{E}^c}. \tag{97}$$

By convexity of $\mathcal{W}_1$ over mixtures with the same weights,

$$\mathcal{W}_1(\mu, \mu') \le p\,\mathcal{W}_1(\mu_{\mathcal{E}}, \tilde{\mu}_{\mathcal{E}}) + (1-p)\,\mathcal{W}_1(\mu_{\mathcal{E}^c}, \tilde{\mu}_{\mathcal{E}^c}). \tag{98}$$

The remaining difference between μ' and $\tilde{\mu}$ is only the event-mass mismatch. Since the feature diameter is bounded by $b_{\max}$,

$$\mathcal{W}_1(\mu', \tilde{\mu}) \le b_{\max}|p - \tilde{p}|. \tag{99}$$

The result follows by the triangle inequality. □

A.5 Assumption and diagnostic mapping

The diagnostics examine the ingredients of the finite-time argument on the fitted priors and evaluated windows. Table 4 reports support, operator scale, and residual scale; Appendix B.4 examines generated tube scores, operator products, and event coverage. These measurements assess the assumptions where the method is used.

The SN-CNF comparison and decoded-field guardrails assess behavior under the regularized chart, without certifying the lower bound in Assumption A.1. Regime counts and residual scales check fitting support and numerical conditioning. Finite-product norms probe amplification directly: Case1 uses sampled descriptor-path products, while Case2/3 use exact powers of the fitted single-regime operator. The noise gate limits predicted-clean error amplification; reliability weights address assignment ambiguity, not denoising-error probability. Event occupancy and event-window counts assess the additional coverage requirement.

A.6 Scope of the current theory

The theory describes local stochastic transition closure. Its implications concern:

- the kernel-induced local Koopman interpretation,
- predicted-clean proxy validity and one-step tube control,
- fixed-lag temporal transport and diagnostic pushforward bounds,
- finite-time accumulation under bounded local amplification, and
- propagation through a stable CNF chart.

These statements hold on the specified chart patches and evaluated finite windows, including the single-regime case. A monotone implication from decreasing the average training penalty to improving sampled rollouts would additionally require control of denoiser generalization and reverse-diffusion stability. The rollout comparisons in Section 5 evaluate that empirical link. Event-conditioned diagnostics further require adequate coverage, as isolated by Proposition A.9.

The evidence has corresponding roles: prior diagnostics check support and residual geometry, finite-product diagnostics assess accumulation, and the seed and common-step comparisons examine the trained generators. Their interpretation follows the same separation between fitting a transition score, applying it during training, and evaluating sampled trajectories.

B Additional experimental diagnostics

The submitted backbone, fitted-prior, tube-score, amplification, horizon, event, factorial, and Case1 sensitivity diagnostics are retained. Analyses completed afterward are identified where introduced; the data-partition and evaluation-unit tables clarify the original protocol.

B.1 Metric definitions and reporting roles

Temporal transport features. The vector $\boldsymbol{\psi}_t$ in Eq. (31) has 23 coordinates: kinetic energy, the first 16 of 24 spectral-energy bins, four derivative summaries (enstrophy, squared strain rate, squared velocity-gradient magnitude, and divergence ratio), and the spatial mean and standard deviation of a scalar field. The scalar is pressure in Case1/2 and velocity magnitude in the two-channel Case3 fields. All available velocity components enter kinetic energy: two in Case1/3 and three in Case2. Spectral and derivative summaries use the case-specific spatial operators described in Appendix B.14.

Standardization is performed independently for every DNS or generated evaluation window, after the windows have been extracted. Each raw feature is centered by its temporal mean and divided by its population standard deviation plus 10^{-8}. Thus this diagnostic measures temporal changes relative to each window's own feature scale; it is complemented by the separate energy-memory and decoded-field diagnostics. This feature normalization is separate from the fixed training-split latent alignment used by the prior.

Pair allocation and pooling. For a pool with N_{win} windows, we draw $\max(1, \lfloor 4096/N_{\mathrm{win}} \rfloor)$ start indices independently in each window, uniformly with replacement among starts admitting the requested lag. Each pair lies within one window. DNS and generated pairs are sampled separately, then their scalar $D_\ell(t)$ values are pooled to form the two empirical laws. The realized pair count is $N_{\mathrm{win}} \max(1, \lfloor 4096/N_{\mathrm{win}} \rfloor)$, which can be slightly below 4096 because of integer allocation. The evaluator random seed is 42. Matching window lengths does not require a generated trajectory to track an individual DNS trajectory or use the same pair indices.

The main columns use lags 1 and 16. The supplementary short-lag summary $\overline{\mathcal{W}}^{\mathrm{time}}_{1,\ell=1:16}$ averages the five measured lags $\{1, 2, 4, 8, 16\}$. The separate tail-transport diagnostic samples lags uniformly from 1 to 64, capped by the number of frames in the selected segment minus one; it is not the lag-16 diagnostic.

Table 2: Diagnostic groups and reporting roles. Primary diagnostics most directly reflect the LSKP transition-prior objective; event-conditioned diagnostics additionally depend on event coverage; guardrails check decoded-field fidelity.

Diagnostic group	Metric	Measured quantity	Claim role
Fixed-lag transition	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1), \mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	$\mathcal{W}_1$ mismatch between pooled DNS and generated scalar feature-distance distributions at fixed lags (Eq. (31)).	Primary evidence for short-lag transition consistency
Short-lag summary	$\overline{\mathcal{W}}_{1,\ell=1:16}^{\mathrm{time}}$	Average fixed-lag temporal $\mathcal{W}_1$ over the five measured lags $\{1, 2, 4, 8, 16\}$.	Robustness summary for appendix sensitivity and factorial diagnostics
Temporal memory	ΔT_{int}, $\Delta T_{\mathrm{int}}^{\text{late-tail}}$	Gap in integral temporal-correlation or memory-time statistics; the late-tail variant uses the final quarter of each evaluation window.	Primary memory-sensitive evidence; late-tail estimates use shorter temporal segments
Tail/event transport	$\mathcal{W}_{1,\mathrm{time}}^{\mathrm{tail}}$, Protocol-E events	Temporal transport in the final quarter of each evaluation window; separately, field diagnostics within DNS-score event groups such as calm, typical, and burst.	Supplementary diagnostics; event-conditioned results require event counts and event-occupancy gaps
Decoded-field guardrails	$\mathrm{Err}_{\log E,L_2}$, ΔG_2, ΔS_2, $\Delta\Omega_2$	Log energy-spectrum discrepancy and normalized gaps in gradient, strain-rate, and enstrophy statistics.	Physical-fidelity guardrails checking whether decoded fields remain comparable

Integral time and memory gap. For a window a with ℓ_{ev} frames, write $\bar{E}_a$ for its temporal mean energy and $j_{\max} = \min(128, \ell_{\mathrm{ev}} - 1)$. Its normalized autocorrelation is

$$\mathrm{ACF}_a(0) = 1, \qquad \mathrm{ACF}_a(j) = \frac{\sum_{t=0}^{\ell_{\mathrm{ev}}-j-1}(E_{a,t} - \bar{E}_a)(E_{a,t+j} - \bar{E}_a)}{\sum_{t=0}^{\ell_{\mathrm{ev}}-1}(E_{a,t} - \bar{E}_a)^2 + 10^{-8}}, \quad 1 \le j \le j_{\max}. \tag{100}$$

We average these curves over windows separately for DNS and generated data. For either averaged curve $\overline{\mathrm{ACF}}$, let J be its first nonpositive lag, or $j_{\max} + 1$ if all available lags are positive. Then

$$T_{\mathrm{int}} = \sum_{j=0}^{J-1} \overline{\mathrm{ACF}}(j), \qquad \Delta T_{\mathrm{int}} = \frac{|T_{\mathrm{int}}^{\mathrm{gen}} - T_{\mathrm{int}}^{\mathrm{DNS}}|}{|T_{\mathrm{int}}^{\mathrm{DNS}}| + 10^{-8}}. \tag{101}$$

The sum includes lag zero and excludes the first nonpositive value. It has units of snapshot intervals; no conversion to physical time is applied. For late-tail memory, the same construction is applied to the final quarter of each evaluation window, with the maximum lag reduced to the smaller of 128 and the DNS/generated tail lengths minus one.

Table 2 groups the diagnostics by their role in supporting our claims. Primary diagnostics most directly reflect the LSKP transition-prior objective. Event-conditioned diagnostics additionally depend on event coverage. Explanatory diagnostics localize where errors occur, and guardrails check whether decoded-field fidelity remains comparable. In the main comparison (Table 1), the two fixed-lag temporal-transport columns and ΔT_{int} serve as the claim-aligned transition and memory readouts, while $\mathrm{Err}_{\log E,L_2}$ is reported as a decoded-field guardrail rather than a primary LSKP target.

B.2 Motivation and backbone diagnostics

Figure 2 reports the representative motivation diagnostics referenced in Section 5.1. The diagnostics identify statistical discrepancies beyond short-range visual plausibility; the metrics need not worsen monotonically with horizon. Table 3 then compares Origin-CoNFiLD with SN-CNF under a fixed downstream latent-diffusion setup and evaluation protocol, isolating the effect of latent-chart stabilization. SN-CNF is therefore a backbone intervention, not a substitute for a transition-side dynamical prior.

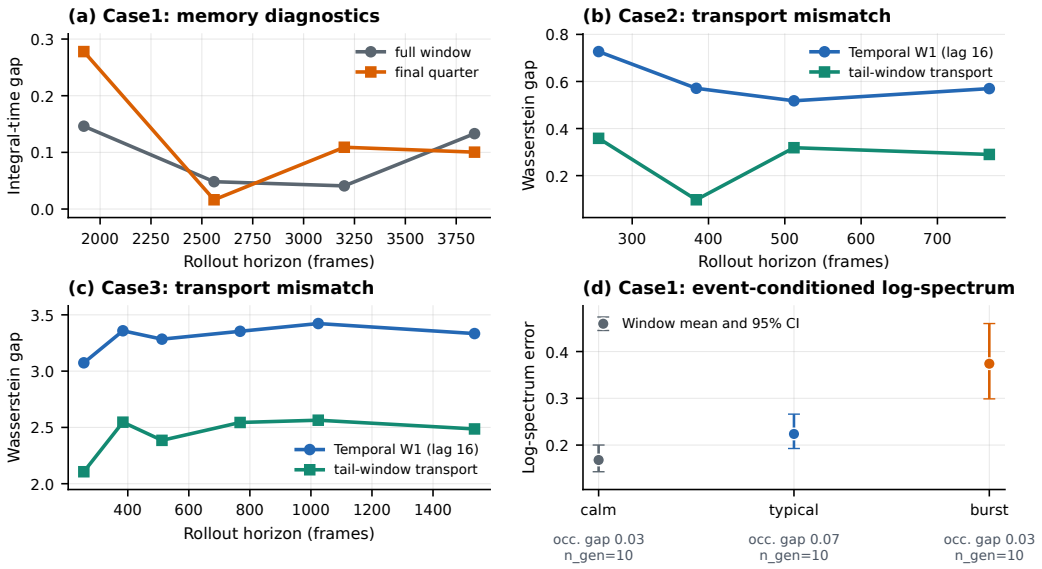

Figure 2: Motivation diagnostics for Origin-CoNFiLD against DNS under Protocol-C+ (a–c) and Protocol-E (d). Horizons count generated frames. (a) Relative integral-time gaps for complete evaluation windows and their final quarters. (b,c) Lag-16 temporal Wasserstein distance and the separate tail-window transport diagnostic, which samples random lags within each window's final quarter. These curves show pooled aggregate diagnostics. (d) Case1 event-conditioned log-spectrum error on the 1473-frame worst-window comparison, using overlapping 128-frame subwindows at stride 48. Points show generated-subwindow means against a fixed DNS spectrum for each event group; bars show 95% percentile bootstrap intervals over those subwindows (200 resamples). DNS event-score quantiles 0.3/0.7 define calm, typical, and burst groups, distinct from LSKP descriptor regimes. Here `occ. gap` is the absolute difference in group occupancy fractions, and `n_gen` counts generated subwindows.

Table 3: Backbone role of the SN-stabilized latent chart. Values are compact aggregate point estimates at the selected Protocol-C+ horizon; lower is better for all reported diagnostics. The table reports selected metrics that improve under SN-CNF alone, before introducing the LSKP transition prior.

Case	Metric	Origin	SN-CNF	Relative change
Case1	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.356	0.306	-14.0%
Case1	ΔG_2	0.0302	0.000884	-97.1%
Case2	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.664	0.519	-21.9%
Case2	ΔG_2	0.0951	0.0633	-33.5%
Case3	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	3.34	3.17	-5.0%
Case3	$\mathrm{Err}_{\log E, L_2}$	0.536	0.478	-10.8%

B.3 Theory-supporting prior diagnostics

The local finite-time analysis in Section 4 requires a well-supported and numerically stable fitted prior. We therefore report theory-supporting diagnostics for the final model of each case, covering regime support, operator scale, residual scale, and direct transition-tube calibration.

These empirical checks probe regime support, operator amplification, and non-degenerate residual scales, the ingredients required for the local stochastic closure interpretation. Weak regime coverage or degenerate residual scales would limit how reliably the fitted transition score can be used over longer windows.

Table 5 evaluates the stochastic transition tubes directly. For each transition, we compute the same softly assigned, diagonal-whitened residual used in the LSKP training loss and report the RMS whitened distance. This diagnostic is a scale and coverage check, not a claim that residuals are exactly

Table 4: Theory-supporting prior diagnostics for LSKP-final. Regime support is measured by nearest-center hard assignment in the normalized descriptor space. Columns report regime-support counts, the mean and maximum spectral norm of local transition cores, and the mean and range of regime-wise residual standard deviation.

Case	R	Support counts	$\|\mathbf{K}_r\|_2$ mean/max	residual std. mean/range
Case1	8	train min/med/max: 162k/196k/245k; valid: 8/8 active, min 17.9k	$1.10/1.16$	$0.276\,[0.150, 1.42]$
Case2	1	train: 98.2k; valid: 1/1 active, 6.38k	$1.02/1.02$	$0.474\,[0.280, 0.661]$
Case3	1	train: 522k; valid: 1/1 active, 8.42k	$1.02/1.02$	$0.166\,[0.072, 0.340]$

Table 5: Whitened transition-tube calibration for LSKP-final. Coverage columns report the fraction of transitions with RMS whitened distance at most 1 or 2. The generated split uses LSKP-final long rollouts; RMS values on the same order as the training split, together with high two-tube coverage, indicate that generated transitions remain within the calibrated stochastic tubes.

Case	Split	Transitions	RMS mean	RMS q95	Cov. ≤ 1	Cov. ≤ 2	Active regimes
Case1	train	1.61M	0.990	1.239	55.5%	99.99%	8/8
Case1	valid	170.9k	1.013	1.256	49.0%	99.93%	8/8
Case1	test	170.9k	1.009	1.261	49.0%	100.0%	8/8
Case1	generated	11.5k	0.925	1.152	72.5%	100.0%	8/8
Case2	train	98.2k	0.989	1.234	52.1%	100.0%	1/1
Case2	valid	6.38k	0.734	0.928	99.2%	100.0%	1/1
Case2	test	6.38k	0.793	0.997	95.7%	100.0%	1/1
Case2	generated	6.90k	0.952	1.189	62.1%	100.0%	1/1
Case3	train	522.5k	0.726	0.961	96.7%	99.90%	1/1
Case3	valid	8.42k	0.650	0.857	100.0%	100.0%	1/1
Case3	test	8.42k	0.715	0.991	95.2%	100.0%	1/1
Case3	generated	4.61k	0.744	1.024	94.3%	99.35%	1/1

Gaussian. Generated rows use LSKP-final rollouts at the final Protocol-C+ horizon with matched rollout seeds.

Generated transitions sometimes show a larger within-one-tube fraction than train/validation/test transitions, most clearly in Case1. We read this as a side effect of training-time LSKP regularization, which pulls predicted-clean transitions toward the fitted tube core, rather than as evidence that generated trajectories are more faithful than DNS.

We also check descriptor-space support directly. For each generated transition, we compute the normalized transition descriptor used by the frozen prior and measure its nearest-center distance in the standardized descriptor space. Table 6 indicates that generated transitions rarely fall outside the train-split descriptor support: the fraction above the train 99th percentile is below 1.5% in all three cases. This supports evaluating the fitted score on sampled latent transitions. Case2 and Case3 remain single-regime tube checks rather than multi-regime expressivity tests.

Finally, we examine the confidence score used for selective weighting. It measures regime-assignment ambiguity, not a calibrated probability of transition error. In Case1, where the prior has eight regimes, Pearson correlations between confidence and whitened transition-tube RMS are 0.0246 on train (1.61M transitions), 0.00028 on validation (170.9k transitions), 0.0214 on test (170.9k transitions), and 0.0197 on generated transitions (11.5k transitions). These marginal correlations do not characterize the joint activity/confidence rule. Tables 27 and 28 separate its fixed subgroups and quantify the supported joint contrasts. The single-regime priors in Case2 and Case3 make confidence constant; selective weighting is inactive while the stochastic-tube penalty remains active.

B.4 Mechanism diagnostics: tube score, amplification, and event coverage

The preceding tables check whether the fitted LSKP prior is well supported and whether LSKP-final generated transitions remain inside calibrated stochastic tubes. We next ask whether these mechanism quantities track the primary temporal diagnostics used for the main claim. The audit compares

Table 6: Descriptor-support check for generated LSKP-final transitions. Distances are dimension-normalized nearest-center distances in the standardized descriptor space. The last two columns report the fraction of generated transitions whose descriptor distance exceeds the train-split 95th and 99th percentile thresholds.

Case	Generated median	Train q95	Gen. $>$ train q95	Gen. $>$ train q99
Case1	0.923	1.113	1.9%	0.3%
Case2	0.987	1.228	3.1%	0.4%
Case3	0.852	1.044	3.0%	1.3%

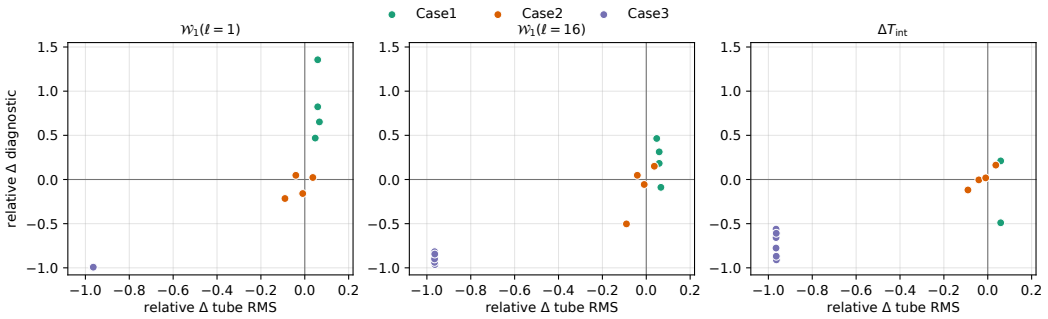

Figure 3: Mechanism diagnostic relating fitted-tube discrepancy to primary temporal diagnostics. Each panel shows 14 case–horizon pairs: four in Case1, four in Case2, and six in Case3. Relative deltas are the LSKP-final minus SN-CNF difference divided by the absolute SN-CNF value; tube RMS is the coordinate-normalized whitened residual diagnostic. The lower-left quadrant marks aligned gains: LSKP-final has both lower tube RMS and lower diagnostic value. The figure supports the transition-tube interpretation as an explanatory diagnostic; it is not used for checkpoint selection and does not imply that every decoded-field or event-conditioned metric is determined by the tube score.

SN-CNF with LSKP-final in the same SN-stabilized latent chart, using the frozen final LSKP prior for each case. Origin-CoNFiLD is excluded from the tube-score comparison because it uses a different latent chart, leaving the final-prior tube score not chart-matched.

For each generated transition, we compute the RMS of the whitened residual underlying the LSKP score, then average over the stored tube-score pool for each case and horizon. This RMS diagnostic is distinct from the weighted, coordinatewise training loss. We join these summaries to the horizon manifest in Table 35 and subtract SN-CNF from LSKP-final. Across the 42 comparisons over three primary diagnostics, 24 show both lower tube RMS and lower diagnostic error, 13 show both higher, and 5 disagree. Figure 3 plots relative changes, divided by the corresponding SN-CNF magnitude; the association statistics below use absolute changes.

Association audit added after submission. Table 7 lists all 14 case–horizon pairs, with one tube-RMS change shared by the three diagnostic comparisons at each point. The grid has four horizons in Case1, four in Case2, and six in Case3. Tube summaries concatenate transitions from the archived generation files, while temporal diagnostics retain the original horizon-manifest estimator and pool. These pools need not coincide; pairing is by case and horizon, not by identical individual evaluation windows.

For each primary diagnostic, Table 8 uses 1000 paired resamples of the 14 aggregate points, with replacement and seed 3407. Each draw keeps the tube and diagnostic change together. Intervals are the 2.5th and 97.5th percentiles of finite correlation estimates; the same draws also summarize sign agreement. The points reuse fitted models and priors across horizons, so these intervals describe sensitivity to resampling the fixed grid, not independent training or trajectory variability.

The pooled correlations are positive, while the within-case correlations in Table 9 are mixed in sign. Omitting Case3 lowers Pearson correlations to 0.7432, 0.7765, and 0.6427 for lag 1, lag 16, and memory. Thus between-case scale contrasts contribute substantially to the pooled pattern. Together

Table 7: Complete aggregate inputs for the tube-score association. All four change columns are absolute LSKP-final minus SN-CNF differences; negative values indicate lower discrepancy or error. Files and transitions give tube-score input counts per model, which agree between SN-CNF and LSKP-final within each row. A file can contain several trajectories. Counts describe the tube pool, not the metric-window pool or independent training repetitions.

Case	Horizon	Files	Transitions	Tube RMS	Lag 1	Lag 16	Memory
Case1	1920	1	1,919	0.0424892	0.0250746	0.125108	0.0865135
Case1	2560	1	2,559	0.0518849	0.0393707	0.0687624	0.0220205
Case1	3200	1	3,199	0.0582017	0.0363543	-0.0378395	0.0551094
Case1	3840	3	11,517	0.0514439	0.0516589	0.0957791	-0.0459369
Case2	256	1	765	-0.0403924	0.039762	0.0335683	-0.00419474
Case2	384	1	1,149	-0.0887374	-0.133577	-0.311327	-0.107949
Case2	512	1	1,533	0.03406	0.0125881	0.0799622	0.124185
Case2	768	3	6,903	-0.00905633	-0.0988282	-0.0292777	0.0154029
Case3	256	1	255	-19.6023	-5.59331	-2.80059	-0.889151
Case3	384	1	383	-19.9052	-5.61041	-2.93433	-0.85137
Case3	512	1	511	-20.0858	-5.65595	-2.96676	-0.760594
Case3	768	1	767	-20.3336	-5.53693	-2.54056	-0.550532
Case3	1024	1	1,023	-20.3413	-5.59328	-2.78796	-0.641983
Case3	1536	4	6,140	-20.2175	-5.61204	-2.67548	-0.596377

Table 8: Pooled associations between absolute tube and diagnostic changes over 14 case–horizon pairs per diagnostic. Brackets are 95% percentile intervals from paired aggregate-point resampling. Sign agreement includes both joint improvement and joint worsening: 13/14 points for lag 1 and 12/14 for lag 16 and memory.

Diagnostic	Pearson	Spearman	Sign-agreement fraction
$\mathcal{W}_1^{\mathrm{time}}(1)$	0.9997 [0.9994, 0.9999]	0.7978 [0.4433, 0.9598]	0.9286 [0.7857, 1.0000]
$\mathcal{W}_1^{\mathrm{time}}(16)$	0.9946 [0.9901, 0.9988]	0.7495 [0.3995, 0.9233]	0.8571 [0.6429, 1.0000]
ΔT_{int}	0.9619 [0.9401, 0.9872]	0.7582 [0.3568, 0.9152]	0.8571 [0.6429, 1.0000]

with the retained joint worsenings and disagreements, these results support an exploratory mechanism interpretation at the aggregate scale; they do not establish a uniform relation within cases or a causal training-loss-to-rollout guarantee.

We next inspect finite-time amplification of the fitted local transition closure. Case1 uses sampled regime-switching products along generated descriptor sequences, while Case2 and Case3 use exact powers of their selected single-regime stochastic tubes. For Case1, each mean aggregates 6144 sampled products with at most 2048 start positions per rollout seed; Case2 and Case3 are single-regime checks, so we report exact powers rather than sampled uncertainty. Figure 4 (top) shows near-marginal or transiently expansive products rather than uniform contraction. At product length 16, the Case1 mean/q95 amplification is $1.45/1.71$ for SN-CNF generated descriptor paths and $1.44/1.68$ for LSKP-final, while Case2 gives 0.939 and Case3 gives 1.43 under the fitted single-regime operators. These values are consistent with the theoretical emphasis on finite-time amplification and short-lag control, rather than with a global contractivity claim.

We also audit event coverage for the Protocol-E event-conditioned analysis. Figure 4 (bottom) reports Case1 generated/DNS event-window counts and occupancy gaps for calm, typical, and burst windows. The groups follow the Protocol-E definition in Appendix B.13: the event score is a DNS-referenced 95th-percentile enstrophy proxy, with calm and burst thresholds set by the 0.3 and 0.7 DNS-score quantiles. A cell is marked coverage-limited when the generated or DNS event-window count is below 10, or when the absolute event-occupancy gap is at least 0.10. Seven of nine event-group rows have low generated or DNS event counts, and six of nine carry occupancy-gap warnings. This is why we use event-conditioned results to localize failures and coverage effects, rather than to claim unconditional regime-wise dominance.

B.5 Implementation and checkpoint details

The train/validation/test partition is temporal: Case1 leaves 127 retained snapshots between adjacent segments, while Case2/3 use disjoint contiguous segments with no additional gap. Windows are formed separately within each segment. Final metrics are computed on held-out raw physical fields rather than latent-space validation or test losses. Protocol-C+ uses fixed pairing rules, and Protocol-E

Table 9: Case-specific and leave-one-case-out associations for the same absolute changes. Each cell is a point correlation; the sample count is the number of case–horizon pairs per diagnostic. "Omit Case3", for example, retains the eight Case1/2 pairs. Within-case and leave-one-case-out results separate horizon ordering from the pooled association.

		Lag 1		Lag 16		Memory	
Scope	Pairs	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
Case1 only	4	0.4977	0.2000	-0.8970	-1.0000	-0.3100	-0.2000
Case2 only	4	0.5268	0.4000	0.8567	0.8000	0.9853	1.0000
Case3 only	6	-0.2604	-0.4286	-0.4759	-0.6000	-0.9195	-0.8286
Omit Case1	10	0.9996	0.6606	0.9930	0.6485	0.9488	0.6121
Omit Case2	10	0.9997	0.6485	0.9954	0.5394	0.9569	0.5394
Omit Case3	8	0.7432	0.5000	0.7765	0.3095	0.6427	0.4524

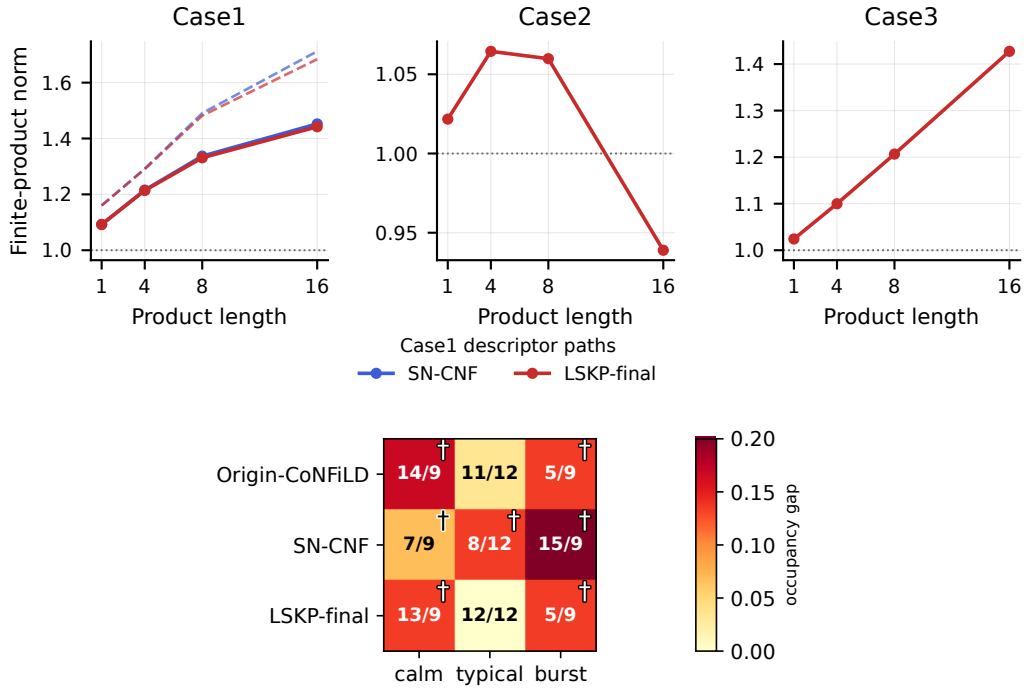

Figure 4: Additional mechanism diagnostics. Top: finite-product norms of the fitted transition closure. In Case1, solid and dashed curves show the mean and 95th percentile along generated descriptor paths; Case2 and Case3 show exact powers of their fitted single-regime operators. The dotted horizontal line marks unit amplification. Bottom: Case1 Protocol-E event coverage for the tail-matched comparison. Colors show absolute event-occupancy gaps, and cell text gives generated/DNS window counts; † marks either count below 10 or an absolute occupancy gap of at least 0.10. Together these diagnostics separate finite-time amplification from event-coverage effects.

event groups are defined by a DNS-referenced event score. Case2 follows the same clean contiguous latent organization as the other cases; only case-level prior hyperparameters are selected using validation-side prior diagnostics before test evaluation. The main and pooled comparisons use the rollout seeds specified in Appendix B.11.

Data source and preprocessing. The physical-field arrays and coordinates come from the original CoNFiLD data archive [Du et al., 2024].[1] The Case1 loader discards the first raw snapshot, leaving 16000 frames; Case2/3 retain all 1200/2880 frames. Case1/3 use the supplied point coordinates, while Case2 constructs coordinates on its regular 400×100 grid. CNF coordinates and field channels use

[1] Original data archive: DOI 10.5281/zenodo.14037782.

Table 10: Temporal partitions of the retained latent sequence for prior/diffusion fitting and evaluation. Gap counts omitted snapshots at each segment boundary. Window counts are training/validation/test counts after forming windows within each segment, not counts of independent trajectories.

Case	Frames	Training range	Validation range	Test range	Gap	Windows
Case1	16000	$[0, 12800)$	$[12927, 14400)$	$[14527, 16000)$	127	12673/1346/1346
Case2	1200	$[0, 640)$	$[640, 920)$	$[920, 1200)$	0	385/25/25
Case3	2880	$[0, 2304)$	$[2304, 2592)$	$[2592, 2880)$	0	2049/33/33

Table 11: Common diffusion-training settings for the reported model groups. Case1 uses latent length 128, while Case2 and Case3 use latent length 256.

Category	Setting	Value
Architecture	U-Net width	128 channels, 2 residual blocks, 4 attention heads, 64 head channels
Architecture	Attention resolutions	32, 16, and 8
Diffusion	Noise process	1000 cosine steps
Optimization	Optimizer	AdamW, learning rate 5×10^{-5}, zero weight decay
Optimization	Batch size	16 for training and validation; full-batch microbatching within each mini-batch
Checkpointing	Periodic saves	Every 5000 optimization steps
Validation	Validation cadence	Every 1000 steps for Case1 and LSKP-final runs; every 100 steps for Case2/Case3 Origin-CoNFiLD and SN-CNF runs
EMA	Model averaging	EMA rate 0.9999; generation uses the EMA weights from the selected best checkpoint

min–max normalization to $[-1, 1]$. CNF reconstruction uses the supplied snapshot array; the temporal partitions below apply after snapshot-code extraction, for prior/diffusion fitting and the corresponding physical-field evaluation. All reported lengths, lags, and strides count retained snapshots, with no additional temporal decimation in these loaders.

Table 10 gives the exact partitions, shared by the Origin and SN latent datasets. Indices are zero-based and ranges are half-open. For Case1, the raw-file index is one greater than the corresponding retained-sequence index; Case2/3 have zero offset. Training, validation, and test windows have lengths 128/256/256 for Case1/2/3 and stride one within each segment. These dataset window counts differ from the rollout evaluation windows in Table 34; the final DNS reference segments contain 1473/280/288 frames.

Following the CoNFiLD latent-diffusion training setup, all diffusion models use the same latent-diffusion backbone within each case unless explicitly noted. The optimizer is AdamW with learning rate 5×10^{-5}, zero weight decay, full-batch microbatching (`microbatch=-1`), and EMA rate 0.9999. Our diffusion implementation uses epsilon prediction with the MSE objective in Eq. (9), fixed reverse-process variance, and 1000 cosine-noise steps. We log every 500 optimization steps and save periodic checkpoints every 5000 steps. Validation uses the held-out latent split, with diffusion timesteps fixed by seed 42 and Gaussian noise resampled under the original training recipes. Latents are min-max normalized using the training split only, and the same statistics are stored for inference-time denormalization.

For SN-CNF and LSKP-final, spectral normalization is applied to the latent-conditioning branch of the CNF decoder, namely the maps that inject optimized latent codes into the coordinate-based field decoder. The coordinate branch, diffusion sampler, and inference procedure are otherwise unchanged.

The full procedural pipeline is summarized in Algorithm 1.

Validation checkpoint selection. The CoNFiLD training setup provides separate training and validation latent data and a held-out diffusion-loss signal. We retain that signal and use our own fixed smoothing and improvement rule to select checkpoints. Selection ranks checkpoints only within a training run, independently of the prior loss and test diagnostics; absolute validation losses are not compared across models or latent charts. This keeps model selection separate from the temporal and field statistics used to assess the generated rollouts.

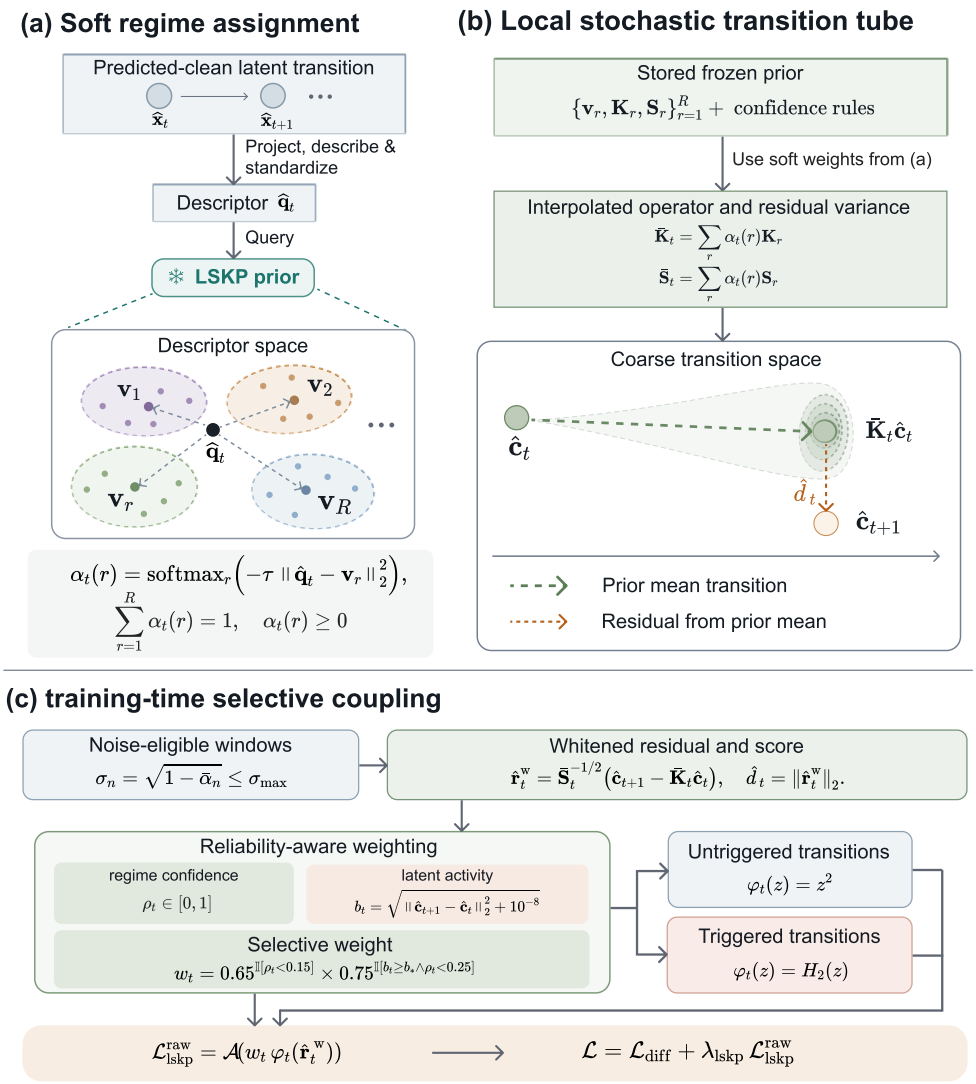

Figure 5: Training-time mechanism of the frozen local stochastic Koopman prior. (a) Predicted-clean transitions yield descriptor queries standardized with fixed training statistics. The soft weights $\alpha_t(r)$ are shared by the operator and residual variance interpolations in (b). (b) The prior mean transition runs from $\hat{\mathbf{c}}_t$ to $\bar{\mathbf{K}}_t\hat{\mathbf{c}}_t$; the residual arrow connects this mean to the predicted-clean endpoint $\hat{\mathbf{c}}_{t+1}$. The faint envelope depicts residual scale associated with $\bar{\mathbf{S}}_t$, not a zero-penalty region. The score $\hat{d}_t$ is the residual norm after whitening by $\bar{\mathbf{S}}_t^{-1/2}$. (c) The penalty acts coordinatewise: z is a scalar coordinate of $\hat{\mathbf{r}}_t^{\mathrm{w}}$, and H_2 is the Huber function with threshold 2. The operator $\mathcal{A}$ averages weighted coordinate penalties across selected windows, adjacent transitions, and residual coordinates, dividing by the number of selected coordinate entries rather than the sum of weights. The coefficient λ_{lskp} follows Eq. (17); window selection follows Appendix B.5. Prior parameters remain frozen while gradients through residuals and soft assignments update the denoiser. For $R = 1$, $\alpha_t(1) = \rho_t = w_t = 1$; the fitted operator, residual scaling, and squared penalty remain active under the prescribed coupling schedule. Inference uses the unchanged sampler and decoder.

The first finite validation loss initializes the smoothed value. Each subsequent finite call updates it as 0.90 times the preceding smoothed value plus 0.10 times the current loss. The first 10 calls participate in smoothing but cannot update the best checkpoint. Beginning with call 11, a checkpoint is accepted when its smoothed loss improves the stored best by more than 10^{-4}. Non-finite losses are

Algorithm 1 DNS-calibrated stochastic transition closure for latent diffusion

Require: Snapshot arrays $\{\mathbf{y}_{0:T_i}^{(i)}\}$, CNF decoder $\mathcal{G}$, latent diffusion model

Fit the shared SN-CNF decoder and snapshot codes to the supplied snapshots by alternating reconstruction updates
Freeze $\mathcal{G}$ and collect latent trajectories $\{\mathbf{x}_{0:T_i}^{(i)}\}$
Apply temporal partitions to the extracted codes for prior/diffusion fitting and evaluation (Appendix B.5)
Fit training latent standardization and PCA, forming $\mathbf{c}_t = \mathcal{P}(\mathbf{x}_t)$
Build raw transition descriptors and fit their training-coordinate means and standard deviations
Standardize descriptors to obtain $\mathbf{q}_t$ and cluster them into regimes $\{\mathbf{v}_r\}_{r=1}^{R}$
for each regime r **do**
    Fit $\mathbf{K}_r$ by ridge regression from $\mathbf{c}_t$ to $\mathbf{c}_{t+1}$
    Estimate residual variances and form the stabilized diagonal scoring matrix $\mathbf{S}_r$
end for
Freeze $\{\mathbf{v}_r, \mathbf{K}_r, \mathbf{S}_r\}_{r=1}^{R}$, $\mathcal{P}$, $\mathcal{Q}$, τ, descriptor normalization, and burst threshold
for diffusion training step **do**
    Sample a minibatch of normalized windows, noise steps, and Gaussian noise $(\mathbf{Z}_0, n, \boldsymbol{\epsilon})$
    Form $\mathbf{Z}_n$ by Eq. (8) and recover $\hat{\mathbf{Z}}_0^{(n)}$ by Eq. (10)
    Undo diffusion normalization to obtain $\hat{\mathbf{X}}_{0:\ell_w}^{(n)}$
    Initialize $\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}} = 0$ and $\lambda_{\mathrm{lskp}} = 0$
    if warmup is complete and this is a scheduled coupling update **then**
        Uniformly select at most four windows with $\sigma_n \le 0.60$, without replacement
        if at least one window is selected **then**
            Build $\hat{\mathbf{q}}_t$ and compute assignments and confidence by Eqs. (11)–(12)
            Form the current stochastic tube by $\bar{\mathbf{K}}_t = \sum_r \alpha_t(r)\mathbf{K}_r$ and $\bar{\mathbf{S}}_t = \sum_r \alpha_t(r)\mathbf{S}_r$
            Compute $\hat{\mathbf{r}}_t^{\mathrm{w}}$ and $\hat{d}_t$ by Eq. (14)
            Apply weights from Eq. (16) and coordinatewise penalties to obtain $\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}}$
            For positive prior loss, compute the coefficient by Eq. (17)
        end if
    end if
    Optimize $\mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{lskp}}\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}}$
end for
At inference, sample with the standard reverse diffusion sampler and decode with $\mathcal{G}$

skipped. Generation uses parameter-EMA weights with decay 0.9999 saved at the accepted step; this parameter averaging is distinct from loss smoothing with decay 0.90.

Validation runs every 1000 optimization steps for Case1 and all LSKP-final runs, and every 100 steps for Case2/3 Origin-CoNFiLD and SN-CNF. The warmup and smoothing operate in validation calls, so their spans in optimization steps differ across these cadences. Table 12 records selected and latest saved steps. In Case3, the best steps 7.9k/10.4k/166k were selected from trajectories trained through 500k/155k/270k for Origin-CoNFiLD/SN-CNF/LSKP-final. Early selection therefore does not mean early termination. The common-step comparison in Appendix B.7.2 examines training-stage sensitivity separately; it does not reconstruct selection under a shared validation cadence.

The frozen checkpoint set in Table 12 is used consistently for the main three-model comparisons and long-horizon/event-conditioned figures. Later saved checkpoints provide training-budget context and the fixed common-step grid; they do not replace the validation-selected main comparison.

The offline LSKP prior is fitted only from clean SN-CNF latent trajectories in the training split; validation latents are used for prior-hyperparameter selection and diagnostic checks, not for fitting the final prior. We use contiguous latent windows, PCA coarse observables, normalized transition descriptors, k-means descriptor clustering with 25 iterations, ridge-regularized regime-wise linear transition fits, diagonal residual variances with a 10^{-6} floor, and validation-side prior diagnostics. Table 13 lists the case-wise settings, using $m = d_c$ for the coarse-observable dimension. These settings are fixed before test evaluation.

Table 12: Frozen validation-side checkpoints used for reported evaluation. Best step is selected by validation-side smoothed diffusion loss; latest saved step records the total available training trajectory for that run.

Case	Model	Best step	Latest saved step
Case1	Origin-CoNFiLD	792000	1000000
Case1	SN-CNF	182000	360000
Case1	LSKP-final	62000	255000
Case2	Origin-CoNFiLD	57300	355000
Case2	SN-CNF	55100	185000
Case2	LSKP-final	103000	315000
Case3	Origin-CoNFiLD	7900	500000
Case3	SN-CNF	10400	155000
Case3	LSKP-final	166000	270000

Table 13: Offline LSKP prior hyperparameters for the final model. Here m is the PCA coarse-observable dimension, R is the number of fitted local stochastic regimes, and history indicates whether previous-increment and second-difference descriptor terms are included. Burst-clustering weight is used only during offline descriptor clustering and is separate from the training-time burst relaxation in Table 17. Case2 and Case3 use the stability-selected single-regime stochastic-tube setting.

Case	m	R	History	Ridge γ	Burst-clustering weight
Case1	48	8	yes	10^{-3}	2.0
Case2	48	1	no	10^{-1}	1.0
Case3	16	1	no	10^{-1}	1.0

Prior selection and validation checks. Prior selection uses feasibility checks on candidates fitted to training transitions. Validation data assess whether every regime is represented and whether residual scales remain above the numerical floor. Operator and whitening scales are inspected for numerical stability, together with the uncapped training diagnostic, applied coefficient, and diffusion-training behavior. Configurations are fixed before test generation. This separates prior selection from the validation diffusion-loss rule used to select a checkpoint within a diffusion run.

Table 14 gives a post-submission audit of stored offline fitting probes and the selected single-regime priors. Case2 probes with two or three regimes leave one or two regimes unvisited in validation, with residual variances at the floor. Case3 probes with three or four regimes retain a 510-transition training regime with no validation examples and floor-scale variance. Ridge 0.1 reduces the largest operator norms relative to the weaker settings but leaves these support failures unresolved. The selected single-regime fits retain validation support and non-floor residual scales.

Table 15 separately reports stored training-log diagnostics for the archived multi-regime and final single-regime diffusion runs. The archived runs have much larger unweighted tube losses; the Case2 run also has a markedly reduced applied coefficient. The logged diagnostic is distinct from the weighted, possibly Huber-penalized loss entering the cap in Eq. (17).

Table 16 gives a separate post-submission evaluation of the archived Case2 eight-regime and Case3 four-regime validation-selected checkpoints. These are different candidates from the full offline probe grid, and their training budgets differ from those of the final runs. Case2 has higher means on three diagnostics and a lower lag-16 mean; Case3 has lower temporal-diagnostic means and a higher spectral error. These mixed outcomes distinguish the support and numerical-health rationale for selecting a prior from downstream metric rankings.

During diffusion training, the LSKP prior is frozen and queried only on predicted-clean latent windows. The final coupling settings are shown in Table 17. All final runs use only the transition-tube regularizer; no additional reconstruction-side LSKP loss is active.

All final LSKP runs use diffusion-noise gate $\sigma_{\max} = 0.60$, MSE transition loss with Huber fallback parameter 2.0, confidence threshold 0.15, confidence weight scale 0.65, burst-confidence threshold

Table 14: Offline candidate-prior health, audited after submission from stored fitting probes and the selected single-regime priors. All probed regime counts and ridge values are shown. Train min. is the smallest regime's training-transition count. Unvisited and Floor give the number of regimes with zero validation support and with mean stored variance at the 10^{-6} floor, respectively, out of R. The norm column is the largest operator Frobenius norm; validation MSE is the unweighted mean squared whitened residual. These probes are distinct from the archived diffusion runs in Tables 15 and 16.

Case	R	Ridge	Train min.	Unvisited	Floor	$\max_r \|\mathbf{K}_r\|_F$	Val. MSE
Case2	1	0.1	98 175	0/1	0/1	6.613	0.5493
Case2	2	0.01	5 320	1/2	1/2	143.0	0.5727
Case2	2	0.03	5 320	1/2	1/2	22.40	0.5727
Case2	2	0.1	5 320	1/2	1/2	6.613	0.5727
Case2	3	0.01	1 530	2/3	2/3	35.80	0.5678
Case2	3	0.03	1 530	2/3	2/3	25.98	0.5678
Case2	3	0.1	1 530	2/3	2/3	6.610	0.5678
Case3	1	0.1	522 495	0/1	0/1	3.986	0.4350
Case3	3	0.01	510	1/3	1/3	75.30	1.392
Case3	3	0.03	510	1/3	1/3	86.85	1.392
Case3	3	0.1	510	1/3	1/3	15.08	1.392
Case3	4	0.01	510	1/4	1/4	75.30	2.047
Case3	4	0.03	510	1/4	1/4	86.85	2.047
Case3	4	0.1	510	1/4	1/4	15.08	2.047

Table 15: Archived training-coupling diagnostics, audited after submission. Medians use the last fifth of positive post-warmup log records, rounded down and limited to 30; the retained record count is shown. The tube loss is the logged unweighted diagnostic before the coefficient, not the weighted, possibly Huber-penalized loss in Eq. (17). The two medians are reported separately and are not multiplied to estimate the applied contribution.

Case	Run	R	Records	Logged tube loss	Applied coefficient
Case2	Final	1	30	2.435×10^{-1}	6.425×10^{-6}
Case2	Archived	8	16	4.125×10^{5}	9.740×10^{-11}
Case3	Final	1	30	2.400×10^{-1}	8.625×10^{-6}
Case3	Archived	4	17	4.490×10^{6}	4.240×10^{-6}

0.25, and burst weight scale 0.75. Compute accounting and peak-memory profiling are reported in Table 32; peak memory uses matched short runs rather than SwanLab whole-machine counters.

Coordinatewise transition penalty. For an ordinary transition, $\varphi_t(z) = z^2$. If $\rho_t < 0.15$ or jointly $b_t \geq b_*$ and $\rho_t < 0.25$, that transition uses $\varphi_t(z) = H_2(z)$, where

$$H_2(z) = \begin{cases} \frac{1}{2}z^2, & |z| \leq 2, \\ 2(|z| - 1), & |z| > 2. \end{cases} \tag{102}$$

The function acts on each coordinate of $\hat{\mathbf{r}}_t^{\mathrm{w}}$ before averaging. Each transition's weight w_t multiplies all its coordinate penalties; $\mathcal{A}$ then divides their sum by the number of selected coordinate entries, not by the sum of weights. Without an eligible window the LSKP contribution is zero. For an unweighted squared-loss transition, the coordinate mean is $\hat{d}_t^2/d_c$, and the corresponding RMS distance is $\hat{d}_t/\sqrt{d_c}$. Applying Huber to either scalar distance would give a different objective.

Latent alignment and PCA. All final priors use contiguous windows with unit temporal stride. Alignment uses fixed training statistics without subtracting each window's first code. Offline fitting first standardizes each latent coordinate using its population mean and standard deviation over training windows and times. Write the mean as $\boldsymbol{\mu}_x$ and the stored scale as $\mathbf{s}_x$, which equals the population standard deviation plus 10^{-8}. Then

$$\widetilde{\mathbf{x}}_t = (\mathbf{x}_t - \boldsymbol{\mu}_x) \oslash \mathbf{s}_x, \qquad \mathbf{c}_t = \mathbf{V}_{\mathcal{P}}^{\top}(\widetilde{\mathbf{x}}_t - \boldsymbol{\mu}_{\mathcal{P}}), \tag{103}$$

where $\oslash$ denotes coordinatewise division, and $\boldsymbol{\mu}_{\mathcal{P}}$ and the orthonormal basis $\mathbf{V}_{\mathcal{P}}$ are fitted to aligned training latents. PCA retains the fixed dimensions 48/48/16 in Case1/2/3, without rescaling

Table 16: Post-submission rollout evaluation of archived multi-regime candidates: Case2 uses $R = 8$ and Case3 $R = 4$, compared with the final $R = 1$ models. Archived checkpoints were selected by validation diffusion loss at 49k/47k; final checkpoints are 103k/166k. Archived runs ended near 50k, whereas final training budgets were 315k/270k. Both columns use rollout seeds 42/43/44, horizons 768/1536, the same evaluator and window pooling, and 72/57 evaluation windows. Entries are window means with 95% percentile bootstrap intervals (200 resamples), reported separately from the aggregate estimates in Table 1. All four diagnostics are shown; lower is better. This is an archived-candidate sensitivity comparison with unequal training budgets.

Case	Diagnostic	Archived multi-regime	Final $R = 1$
Case2	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1)$	0.7174 [0.6823, 0.7557]	0.4993 [0.4563, 0.5498]
Case2	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.6354 [0.5756, 0.6888]	0.6523 [0.6104, 0.7022]
Case2	ΔT_{int}	0.9011 [0.8796, 0.9208]	0.8658 [0.8411, 0.8927]
Case2	$\mathrm{Err}_{\log E, L_2}$	0.4353 [0.4313, 0.4390]	0.4284 [0.4259, 0.4311]
Case3	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1)$	0.04186 [0.03485, 0.04969]	0.04751 [0.03726, 0.05770]
Case3	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.3835 [0.3355, 0.4360]	0.5204 [0.4261, 0.5982]
Case3	ΔT_{int}	0.2726 [0.2204, 0.3298]	0.2985 [0.2465, 0.3510]
Case3	$\mathrm{Err}_{\log E, L_2}$	0.2670 [0.2398, 0.2906]	0.2478 [0.2185, 0.2768]

Table 17: Training-time LSKP coupling hyperparameters for final runs. Interval is measured in optimization steps; subbatch is the maximum number of noise-eligible windows per coupling update.

Case	Window	Warmup	Interval	Subbatch	λ_{base}	Ratio cap
Case1	128	5000	4	4	0.020	0.03
Case2	256	10000	4	4	0.015	0.02
Case3	256	5000	4	4	0.020	0.03

components by singular values. During diffusion training, the predicted-clean window is first restored from min–max normalization. The stored scale $\mathbf{s}_x$ receives an additional 10^{-8} denominator adjustment before the same PCA projection. Validation, test, and generated queries reuse the stored training statistics.

Descriptor construction. The base raw blocks are given in Eq. (6), with $\varepsilon_q = 10^{-6}$. Case1 inserts $\mathbf{h}_{t-1}$ and $\mathbf{h}_t - \mathbf{h}_{t-1}$ between $\mathbf{h}_t$ and $\delta_{e,t}$. At each window's first transition, $\mathbf{h}_{-1} = \mathbf{0}$, so the first second-difference block equals $\mathbf{h}_0$; history never crosses window boundaries. Case2/3 use the base form. The resulting descriptor dimensions are 194/98/34. Each raw coordinate is centered by its training mean and divided by its training population standard deviation, floored at 10^{-6}. Both clustering and subsequent soft assignment use these standardized descriptors. These latent-activity summaries are separate from the physical-field event labels in Protocol-E.

Local fitting and numerical scales. Lloyd k-means uses seed 3407 and at most 25 iterations, stopping when successive centers agree with absolute/relative tolerances $10^{-8}/10^{-5}$. Centers are initialized from a seeded random subset of the clustering descriptors; an empty cluster retains its preceding center. Soft assignment uses inverse temperature $\tau = 5.0$ in Eq. (11). For Case1, descriptors at or above the training 75th-percentile increment-norm threshold are duplicated once for center fitting, giving burst boost 2.0; Case2/3 use boost 1.0. This duplication affects center fitting only. Original training transitions are then assigned to their nearest center for ridge fitting, using the case-specific parameters in Table 13. If a regime has fewer than $\max(4, d_c + 1)$ transitions, its operator and residual scale use the $\min(N_{\mathrm{fit}}, \max(16, d_c + 1))$ nearest training descriptors to that center, where N_{fit} is the total number of training transitions. Per-coordinate residual variances are estimated from squared residuals, floored at 10^{-6}, and stored as a vector. Whitening adds a further 10^{-6} to each variance; $\mathbf{S}_r$ denotes this stabilized diagonal scoring matrix throughout the formulas. Since regime weights sum to one, this is equivalent to adding $10^{-6}\mathbf{I}$ after interpolating the stored variances.

Reliability, coefficient, and update selection. During diffusion training, the frozen prior projects the predicted-clean latent window through the same PCA map and descriptor builder. Soft assignment

uses the inverse-temperature convention in Eq. (11). For the confidence score in Eq. (12), logarithm arguments are floored at 10^{-8} and the final score is clipped to $[0, 1]$. Threshold comparisons define discrete masks. A single regime gives $\rho_t = 1$, so both selective triggers are inactive. The online activity proxy uses $\varepsilon_b = 10^{-8}$. Its stored threshold b_* is the training 75th percentile of the offline coarse-increment score $\sqrt{\|\mathbf{c}_{t+1} - \mathbf{c}_t\|_2^2 + 10^{-6}}$. These latent-activity quantities are distinct from physical-field event labels used in evaluation.

The cap in Eq. (17) uses $\varepsilon_{\mathrm{cap}} = 10^{-8}$ and the weighted, possibly Huber-penalized $\mathcal{L}_{\mathrm{lskp}}^{\mathrm{raw}}$ before multiplication by λ_{lskp}. This is distinct from the unweighted squared-residual diagnostic. Both loss values entering the ratio are detached; gradients still pass through the prior residual and soft assignments to the denoiser. If the raw prior loss is zero, the coefficient is set to zero. Reported runs use no amplitude ramp after warmup. Values of λ_{base} and r_{cap} are listed in Table 17.

Optimization steps are counted from zero. Coupling is enabled once the warmup count is reached and the global step is divisible by four. From windows passing $\sigma_n \leq 0.60$, at most four are sampled uniformly without replacement; if four or fewer qualify, all are retained. The mean in $\mathcal{A}$ is over these selected windows. Before warmup, on other optimization steps, or with no eligible window, only the diffusion loss is optimized. Long rollouts use the selected EMA checkpoint, the standard diffusion sampler, training-split latent min–max denormalization, and the inherited CNF decoder.

B.6 Observable and partition diagnostics

B.6.1 PCA coverage and numerical conditioning

The fixed observable map gives a reproducible coordinate system for fitting and querying the transition prior. The post-submission audit in Table 18 examines dimensions 8, 16, 32, 48, 64, and 96 using the original training and validation latents with frozen training alignment. The dimension audit constructs nested PCA bases from the training covariance; validation data do not determine the directions. The explicit final dimensions 48/48/16 take precedence over the explained-variance target in fitting.

Training covariance retention is the fraction of the training eigenvalue sum captured by the retained basis. Validation covariance retention projects validation deviations from their own mean onto that training basis. Validation energy instead uses deviations from the training PCA mean; its unretained fraction is the projection residual. The mean-shift column gives the fraction of this train-centered validation energy attributable to displacement between the split means. The audit uses 12673/385/2049 training windows and 1346/25/33 validation windows for Case1/2/3, with 128/256/256 frames per window. These are stored windows, not independent trajectory counts.

At the final dimensions, train/validation covariance retention is 72.354/67.063%, 54.222/5.236%, and 58.535/45.387% for Case1/2/3. Thus the observable captures a selected subspace, with particularly different held-out covariance coverage in Case2. The separate energy and mean-shift readouts distinguish covariance coverage from a change in centering. The regularized diagnostic systems have finite condition numbers throughout the grid. Because changing dimension also changes the target coordinates, the per-coordinate MSE values are diagnostic readouts, not a common-target dimension-selection score. No downstream rollout metric was used to select entries from this audit.

B.6.2 Descriptor neighborhoods and transition-fit coherence

The Case1 audit, added after submission, tests how descriptor neighborhoods organize transition fits on the fixed 48-dimensional observable. We retain the frozen eight-regime soft prior and refit hard nearest-center models with ridge 10^{-3}. Comparators are a global $R = 1$ fit, five random-label controls, and five circular-shift controls. Random training labels preserve the exact training regime histogram; validation and test labels are sampled from those training frequencies. Circular shifts use a nonzero offset separately within each stored transition window, preserving its label histogram while disrupting alignment with the transition being scored. Base seeds are 3407–3411, with offsets of 10000 and 20000 for validation and test controls. All operators and residual scales are fitted only on training transitions.

Tables 19 and 20 retain every comparator on 1609471/170942/170942 train/validation/test transition entries. MSE averages squared one-step residuals across entries and coordinates. The diagnostic NLL averages $\frac{1}{2}[\log s_j + (\epsilon_{t,j}^{\mathrm{fit}})^2/s_j]$, where $\epsilon_{t,j}^{\mathrm{fit}}$ is coordinate j of the fitted transition residual and s_j is a training-fitted diagonal variance scale. Hard fits estimate this scale by the training mean

Table 18: PCA dimension audit under frozen latent alignment. Asterisks mark the dimensions used in the final models. Covariance retention, validation energy, projection residual, and mean-shift fractions are percentages with the definitions in the text. MSE is the per-coordinate validation residual of a global one-step ridge fit with coefficient 10^{-3} at every dimension. Cond. is the condition number of its regularized training normal matrix. These controlled fits use a common coefficient; final prior coefficients remain those in Table 13.

Case	d_c	Train cov. (%)	Val. cov. (%)	Val. energy (%)	Proj. residual (%)	Mean shift (%)	Val. MSE	Cond.
Case1	8	34.489	31.085	32.215	67.785	3.562	0.206368	7.07
	16	47.241	43.197	44.341	55.659	3.562	0.145225	10.92
	32	62.380	56.755	57.868	42.132	3.562	0.112676	18.64
	48*	72.354	67.063	67.991	32.009	3.562	0.0951104	25.22
	64	80.357	75.674	76.345	23.655	3.562	0.0841694	30.61
	96	92.470	90.018	90.300	9.700	3.562	0.0699076	45.23
Case2	8	12.643	0.701	0.710	99.290	4.751	0.0797078	1.29
	16	23.085	1.572	1.560	98.440	4.751	0.0960369	1.47
	32	40.456	3.401	3.376	96.624	4.751	0.108109	1.90
	48*	54.222	5.236	5.153	94.847	4.751	0.117152	2.37
	64	65.306	7.397	7.259	92.741	4.751	0.130499	2.91
	96	81.495	12.787	12.519	87.481	4.751	0.168293	4.56
Case3	8	48.627	35.445	45.536	54.464	17.570	0.00959487	6.83
	16*	58.535	45.387	53.889	46.111	17.570	0.00843248	18.68
	32	69.445	55.150	62.110	37.890	17.570	0.0103638	28.84
	48	77.387	64.504	69.864	30.136	17.570	0.0109708	37.81
	64	83.584	72.478	76.455	23.545	17.570	0.0107537	48.72
	96	92.012	83.625	85.660	14.340	17.570	0.0095399	81.38

Table 19: Case1 descriptor-partition residuals and diagonal-Gaussian scores. The frozen soft prior uses its existing operators; the hard Lloyd, global, random-label, and circular-shift rows refit operators on training data only. Each random/shift row identifies its base label-control seed, not a diffusion-training replicate. All splits and all five controls per family are shown. Lower MSE/NLL is better; scores are descriptive point estimates.

	Train		Validation		Test	
Variant	MSE	NLL	MSE	NLL	MSE	NLL
Frozen soft prior	0.0891635	-0.8332616	0.0957547	-0.8085273	0.0938457	-0.8130337
Lloyd hard fit	0.0891967	-0.8331969	0.0958075	-0.8082181	0.0938943	-0.8128183
Global $R = 1$	0.0934800	-0.8124520	0.0951103	-0.8119836	0.0935793	-0.8169935
Random 3407	0.0934603	-0.8125615	0.0951265	-0.8118892	0.0935967	-0.8168944
Time shift 3407	0.0934460	-0.8126431	0.0951451	-0.8118097	0.0936151	-0.8168098
Random 3408	0.0934608	-0.8125575	0.0951295	-0.8118898	0.0935976	-0.8168914
Time shift 3408	0.0934462	-0.8126411	0.0951388	-0.8118137	0.0936121	-0.8168097
Random 3409	0.0934603	-0.8125581	0.0951269	-0.8118929	0.0935961	-0.8168969
Time shift 3409	0.0934458	-0.8126413	0.0951419	-0.8117918	0.0936115	-0.8168146
Random 3410	0.0934598	-0.8125607	0.0951264	-0.8119019	0.0935970	-0.8168921
Time shift 3410	0.0934479	-0.8126373	0.0951415	-0.8118029	0.0936099	-0.8168242
Random 3411	0.0934607	-0.8125591	0.0951266	-0.8118884	0.0935986	-0.8168886
Time shift 3411	0.0934479	-0.8126382	0.0951436	-0.8117978	0.0936084	-0.8168206

squared residual, floored at 10^{-6}; the soft prior interpolates its operators and stored variances before scoring. The diagnostic omits the common Gaussian constant and uses the variance floor without the additional whitening stabilizer used by the training loss. Since descriptors include the transition increment, this is an audit of transition-fit consistency, not a causal forecast from the current state alone.

The descriptor partition yields more distinct operators and smaller training residuals than the global and label-control fits. Its held-out one-step scores do not improve on the global fit: soft-prior validation/test MSE is 0.09575/0.09385, compared with 0.09511/0.09358 for $R = 1$. None of the five frequency-matched random controls improves validation MSE over the global model. We therefore use Lloyd regions as operational neighborhoods in descriptor space; the audit does not identify dynamically optimal or physically unique regime boundaries. These frozen-latent fits characterize the prior construction, whereas the generated-rollout comparisons and component ablations measure the effect of using it during diffusion training.

Table 20: Support and operator separation for the same Case1 partition audit. Counts summarize training entries assigned to each regime; active regimes are reported for validation/test. Every row has 100% coverage on all three splits. Separation averages pairwise operator Frobenius differences divided by the mean Frobenius norm of the two operators; it is zero for one regime. The soft row reports the lower central support count, while refitted rows average the two central counts for the median.

Variant	R	Min. count	Median count	Val./test active	Separation
Frozen soft prior	8	162,087	195,814.0	8/8	0.0845119
Lloyd hard fit	8	162,087	196,962.5	8/8	0.0845119
Global $R = 1$	1	1,609,471	1,609,471.0	1/1	0.0000000
Random 3407	8	162,087	196,962.5	8/8	0.0063142
Time shift 3407	8	162,087	196,962.5	8/8	0.0084054
Random 3408	8	162,087	196,962.5	8/8	0.0062825
Time shift 3408	8	162,087	196,962.5	8/8	0.0083408
Random 3409	8	162,087	196,962.5	8/8	0.0063303
Time shift 3409	8	162,087	196,962.5	8/8	0.0084332
Random 3410	8	162,087	196,962.5	8/8	0.0064377
Time shift 3410	8	162,087	196,962.5	8/8	0.0080861
Random 3411	8	162,087	196,962.5	8/8	0.0062236
Time shift 3411	8	162,087	196,962.5	8/8	0.0081531

Table 21: Additional Case3 diffusion-training runs. Each entry is a window mean with a 95% percentile bootstrap interval (200 resamples), using rollout seeds 42/43/44 at horizon 1536 and 57 overlapping evaluation windows per trained model. Selected steps use each run's validation-loss rule; SN-CNF/LSKP-final retain budgets 155k/270k and validation cadences 100/1000. The CNF chart and fitted prior are fixed across training seeds. Lower is better.

Model / training seed	Selected step	$\mathcal{W}_1^{\mathrm{time}}(1)$	$\mathcal{W}_1^{\mathrm{time}}(16)$	ΔT_{int}	$\mathrm{Err}_{\log E, L_2}$
SN-CNF / 42	10400	5.6362 [5.5898, 5.6843]	3.1795 [3.1355, 3.2321]	0.9759 [0.9744, 0.9773]	0.4934 [0.4808, 0.5053]
SN-CNF / 1042	17900	2.7546 [2.6980, 2.8110]	1.0916 [1.0335, 1.1438]	0.5588 [0.5109, 0.6049]	0.3246 [0.3049, 0.3457]
SN-CNF / 2042	5600	5.8937 [5.8478, 5.9424]	3.4101 [3.3581, 3.4507]	0.9786 [0.9781, 0.9790]	0.5361 [0.5332, 0.5388]
LSKP-final / 42	166000	0.0475 [0.0373, 0.0577]	0.5204 [0.4261, 0.5982]	0.2985 [0.2465, 0.3510]	0.2478 [0.2185, 0.2768]
LSKP-final / 1042	77000	0.0454 [0.0334, 0.0566]	0.5045 [0.4203, 0.5952]	0.2685 [0.2220, 0.3266]	0.2345 [0.2040, 0.2650]
LSKP-final / 2042	79000	0.0500 [0.0371, 0.0666]	0.4718 [0.4202, 0.5455]	0.2617 [0.2163, 0.2996]	0.2151 [0.1878, 0.2391]

B.7 Training variability and checkpoint sensitivity

B.7.1 Case3 replication across diffusion training seeds

After submission, we compared diffusion-training seeds 42, 1042, and 2042 for SN-CNF and LSKP-final; seed 42 identifies the training runs used in Table 1. Both methods share the fixed SN-CNF chart, data split, diffusion architecture, and optimizer; all LSKP runs use the same fitted prior and coupling settings. The budgets and validation cadences match the reference runs in Table 12: 155k steps with validation every 100 steps for SN-CNF, and 270k steps with validation every 1000 steps for LSKP-final. These settings are held fixed across seeds within each method. Each run is evaluated from its own validation-selected parameter-EMA checkpoint using rollout seeds 42/43/44 at horizon 1536.

Table 21 gives all six runs, selected steps, and four diagnostics. Table 22 summarizes the mean and sample standard deviation of the three run-level window means, together with differences between matching training seeds. LSKP-final is lower on every diagnostic in all three seed pairs, including seed 1042, for which SN-CNF has its lowest errors. This supports repeatability of the Case3 gains

Table 22: Case3 variation across three diffusion-training runs. Values are mean ± sample standard deviation of the run-level window means in Table 21. The last column subtracts SN-CNF from LSKP-final for each matching training seed, then summarizes the three differences. LSKP-final is lower in all three seed pairs on every diagnostic. These summaries describe training-run variation, separately from the window intervals above.

Diagnostic	SN-CNF	LSKP-final	Paired difference (LSKP-final − SN-CNF)
$\mathcal{W}_1^{\mathrm{time}}(1)$	4.7615 ± 1.7428	0.0476 ± 0.0023	-4.7139 ± 1.7408
$\mathcal{W}_1^{\mathrm{time}}(16)$	2.5604 ± 1.2772	0.4989 ± 0.0248	-2.0615 ± 1.2845
ΔT_{int}	0.8378 ± 0.2416	0.2762 ± 0.0196	-0.5615 ± 0.2357
$\mathrm{Err}_{\log E, L_2}$	0.4514 ± 0.1118	0.2325 ± 0.0164	-0.2189 ± 0.1177

Table 23: Complete Case3 common-step comparison on the training runs used in Table 1. All three models are evaluated at all three fixed steps using EMA weights (decay 0.9999), rollout seeds 42/43/44, horizon 1536, and 57 overlapping windows per cell. Entries are window means with 95% percentile bootstrap intervals (1000 resamples). These intervals describe window variability at a fixed checkpoint. All four diagnostics are shown; lower is better.

Step	Model	$\mathcal{W}_1^{\mathrm{time}}(1)$	$\mathcal{W}_1^{\mathrm{time}}(16)$	ΔT_{int}	$\mathrm{Err}_{\log E, L_2}$
50k	Origin-CoNFiLD	0.0462 [0.0366, 0.0585]	0.3664 [0.3144, 0.4244]	0.2577 [0.2118, 0.3071]	0.2706 [0.2443, 0.2970]
50k	SN-CNF	0.0454 [0.0354, 0.0558]	0.4075 [0.3517, 0.4661]	0.2407 [0.1948, 0.2899]	0.2418 [0.2135, 0.2682]
50k	LSKP-final	0.0593 [0.0443, 0.0790]	0.4899 [0.4146, 0.5653]	0.2857 [0.2333, 0.3327]	0.2310 [0.2031, 0.2600]
100k	Origin-CoNFiLD	0.0407 [0.0328, 0.0505]	0.4030 [0.3409, 0.4715]	0.2679 [0.2184, 0.3234]	0.2504 [0.2239, 0.2734]
100k	SN-CNF	0.0505 [0.0368, 0.0678]	0.5246 [0.4562, 0.5949]	0.2814 [0.2349, 0.3263]	0.2495 [0.2183, 0.2811]
100k	LSKP-final	0.0398 [0.0273, 0.0562]	0.4225 [0.3629, 0.4788]	0.2514 [0.2059, 0.2958]	0.2288 [0.2026, 0.2539]
155k	Origin-CoNFiLD	0.0359 [0.0261, 0.0472]	0.3572 [0.3019, 0.4155]	0.2401 [0.1880, 0.2938]	0.2662 [0.2387, 0.2928]
155k	SN-CNF	0.0478 [0.0359, 0.0628]	0.5047 [0.4227, 0.6013]	0.2378 [0.1944, 0.2856]	0.2318 [0.2038, 0.2628]
155k	LSKP-final	0.0427 [0.0340, 0.0526]	0.4811 [0.4054, 0.5585]	0.2872 [0.2404, 0.3362]	0.2399 [0.2116, 0.2683]

across diffusion initializations under the specified training and validation protocols. The window intervals and across-run standard deviations describe different levels of variability.

The four additional runs resumed after machine interruptions using periodic model, optimizer, and parameter-EMA states and saved best-checkpoint metadata. Recovery did not preserve full random-number-generator states or the current validation-loss smoothing accumulator; these runs are not bitwise continuations of uninterrupted training.

B.7.2 Case3 common-step comparisons

After submission, we separately evaluated saved checkpoints from the Origin-CoNFiLD, SN-CNF, and LSKP-final training runs used in Table 1, at 50k, 100k, and 155k steps. The steps were fixed from checkpoint availability and the terminal SN-CNF budget; existing rollouts were reused after their checkpoint and evaluation metadata were verified. All nine model-step cells use saved parameter-EMA weights with decay 0.9999, rollout seeds 42/43/44, horizon 1536, and the same four diagnostics. Table 23 retains the complete grid with 1000-resample window intervals.

LSKP-final is lower than SN-CNF on 1/4, 4/4, and 2/4 diagnostics at 50k, 100k, and 155k, respectively. At 100k, the reductions relative to SN-CNF range from 8.3% to 21.3%; at 155k, both fixed-lag transport diagnostics remain lower. Requiring an improvement over both Origin-CoNFiLD and SN-CNF instead gives 1/4, 3/4, and 0/4. Window intervals overlap across these comparisons. The complete grid therefore shows training-stage-dependent ordering and magnitude, complementing

Table 24: Appendix factorial diagnostic for chart stabilization and LSKP transition regularization. Origin+LSKP applies LSKP during diffusion training on the original CNF latent chart. Priors are fitted separately on each chart using the same case-level construction and hyperparameters. Lower is better for all diagnostics. The table is used for mechanism attribution only and does not replace the main three-model comparison; numerical minima are not treated as significance claims.

Case	Diagnostic	Origin	Origin+LSKP	SN-CNF	LSKP-final
Case1	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.390	0.434	0.389	0.443
Case1	$\overline{\mathcal{W}}_{1,\ell=1:16}^{\mathrm{time}}$	0.156	0.232	0.170	0.218
Case1	ΔT_{int}	0.133	0.177	0.177	0.0948
Case1	$\mathrm{Err}_{\log E,L_2}$	0.0906	0.0901	0.111	0.109
Case2	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.570	0.638	0.509	0.493
Case2	$\overline{\mathcal{W}}_{1,\ell=1:16}^{\mathrm{time}}$	0.470	0.447	0.458	0.384
Case2	ΔT_{int}	0.875	0.880	0.882	0.882
Case2	$\mathrm{Err}_{\log E,L_2}$	0.485	0.483	0.42690	0.42687
Case3	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	3.334	0.420	3.168	0.424
Case3	$\overline{\mathcal{W}}_{1,\ell=1:16}^{\mathrm{time}}$	4.891	0.190	4.674	0.197
Case3	ΔT_{int}	0.979	0.271	0.975	0.347
Case3	$\mathrm{Err}_{\log E,L_2}$	0.540	0.191	0.483	0.178

the seed replication under each model's validation rule. The reported main checkpoints remain unchanged; the favorable 100k row is not used as a new selection criterion.

B.8 Factorial diagnostic: latent chart and transition prior

Table 24 crosses original and SN-stabilized CNF charts with diffusion training without and with LSKP. Origin-CoNFiLD and SN-CNF use plain diffusion; Origin-CNF+LSKP and LSKP-final add transition scoring on their respective charts. The prior is fitted separately from training transitions in each chart, using the same case-level construction and hyperparameters, rather than reusing a fitted operator across unaligned coordinates.

The four groups retain the diffusion architecture and sampling procedure within each case. Values are pooled aggregate point estimates over rollout seeds 42/43/44 at the final Protocol-C+ horizon, distinct from the seed-42 main table and the appendix window means. This diagnostic separates chart and prior contributions under the reported checkpoint policies; it is not a matched-step comparison or an interaction significance test.

Generic temporal regularization is tested separately through the trained Case1 controls in Appendix B.9.2 and the Case2/3 objective/gradient comparison in Appendix B.9.1. The factorial addresses which ingredient contributes to the observed gains; these controls address whether the LSKP score is equivalent to the tested difference penalties.

The clearest prior contribution is in Case3: adding LSKP on the original chart reduces lag-16 transport from 3.334 to 0.420, alongside reductions in lag-averaged transport and memory gap. Comparable transition reductions appear when adding LSKP to the SN chart. Thus SN alone does not explain the Case3 transition effect. The four-way comparison also retains Case1 trade-offs and Case2 near-ties. It supports a contribution from the transition prior on both charts, without establishing uniformly better outcomes for the SN chart. SN provides the weight-controlled chart used by the final recipe; the table does not independently rank the numerical stability of prior fits across charts.

B.9 Component ablations and regularization controls

Table 25 reports the Case1 component analysis referenced in Section 5. Its full local-regime, history-aware configuration exercises the local and selective extensions. Case2/3 use single-regime, no-history priors, so their additional runs in Table 26 measure training sensitivity. Cross-case chart/prior attribution is addressed by the factorial diagnostic above. Table 32 reports the training-cost profile for the three reported model groups.

The Case1 component table should be read with the metric hierarchy. The regular temporal-memory row provides the cleanest component signal. The log-spectrum row is a decoded-field guardrail, and

Table 25: Component analysis of the LSKP transition prior on Case1. All ablations retain the same SN-CNF backbone; SN-CNF is shown only as a reference anchor, not as an LSKP ablation. Values are compact point estimates from the reported component runs. Lower is better, but the table is read as a component diagnostic rather than a significance test.

Variant	ΔT_{int}	$\mathrm{Err}_{\log E, L_2}$
SN-CNF	0.0938	0.12
LSKP-final	0.0478	0.0956
w/o reliability weighting	0.0595	0.11
w/o local regimes	0.0603	0.0943
w/o descriptor history	0.141	0.109

Table 26: Independently trained single-regime sensitivity runs for Case2 and Case3. Values are means over pooled evaluation windows with 95% percentile bootstrap intervals (200 resamples), using rollout seeds 42/43/44 at horizons 768/1536 and 72/57 windows. Both columns use the same fitted one-regime prior. The independent runs disable the selective settings, but $\rho_t = w_t = 1$ already makes those branches inactive; their differences therefore do not isolate a gate-removal effect. Independent-run checkpoints were selected at 101k/104k by validation diffusion loss. Intervals describe evaluation-window variability, not retraining uncertainty; lower is better.

Case	Diagnostic	LSKP-final	Independent run
Case2	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1)$	0.499 [0.456, 0.550]	0.530 [0.488, 0.570]
	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.652 [0.610, 0.702]	0.588 [0.531, 0.651]
	ΔT_{int}	0.866 [0.841, 0.893]	0.893 [0.875, 0.907]
Case3	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1)$	0.0475 [0.0373, 0.0577]	0.0516 [0.0378, 0.0687]
	$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	0.520 [0.426, 0.598]	0.524 [0.468, 0.613]
	ΔT_{int}	0.298 [0.247, 0.351]	0.248 [0.203, 0.297]

the slightly lower value for the no-local-regime variant shows that local regimes are not expected to improve every guardrail monotonically. Late-tail component point estimates are not used as component evidence here. This variant uses the final quarter of each evaluation window, rather than selecting physical-field events; event-conditioned behavior and coverage are analyzed separately through Protocol-E and the diagnostics above.

Case1 fixed-gate diagnostics. The selective rule serves two related purposes: low assignment confidence reduces the influence of ambiguous regime fits, and the joint activity/confidence trigger robustifies active transitions with uncertain assignments. We added this diagnostic after submission using the frozen Case1 prior and existing latent sequences, without refitting or varying thresholds. For this offline diagnostic, activity is $b_t^{\mathrm{off}} = \sqrt{\|\mathbf{c}_{t+1} - \mathbf{c}_t\|_2^2 + 10^{-6}}$, with the stored training 75th-percentile threshold $b_* = 2.39634037$. The online proxy in Eq. (16) uses the 10^{-8} stabilizer specified above. Here "burst" means $b_t^{\mathrm{off}} \geq b_*$, distinct from physical-field event labels. The two triggers are $\rho_t < 0.15$ and burst with $\rho_t < 0.25$; their union defines affected transitions, and its complement defines unaffected transitions.

Table 27 shows that low confidence alone does not consistently imply larger residuals. The joint burst/low-confidence subgroup has higher mean RMS than unaffected transitions by 11.1%, 19.2%, and 20.9% on train, validation, and test, respectively. This conditional pattern coexists with the near-zero marginal correlations above. The higher mean is specific to the joint subgroup; the full affected union need not show the same pattern, as the generated split illustrates.

Table 28 retains both source-window and contiguous-block intervals. Resampling preserves transitions within a stored window; blocks additionally preserve adjacency across 32 windows. These intervals describe dependence at the specified resampling scale, not independent training runs or arbitrary longer-range dependence. Only 16 generated transitions enter the joint subgroup, so its generated statistics remain descriptive. This frozen-prior diagnostic concerns stored latent transitions, not the denoiser's noisy predicted-clean training updates. The separate single-run component comparison in Table 25 changes ΔT_{int} from 0.0478 to 0.0595 when weighting is removed; it does not identify the joint subgroup as the cause of that change.

Table 27: Case1 fixed-gate subgroup statistics. Scores are whitened residual RMS, $d_t/\sqrt{d_c}$, from the frozen prior on stored DNS and generated latents. Count n is the number of transition entries, not independent replicates; fractions use each split's total. Groups overlap, so only complementary pairs sum to the total. DNS splits use stored 128-frame windows; generated scores use three complete 3840-frame trajectories. Quantiles and means are descriptive.

Split	Group	n	Fraction (%)	Mean	Median	q90	q95	q99
Train	All	1,609,471	100.00	0.9895	0.9809	1.1785	1.2393	1.3668
	$\rho_t < 0.15$	7,600	0.47	0.9758	0.9756	1.1579	1.2454	1.4281
	$\rho_t \geq 0.15$	1,601,871	99.53	0.9896	0.9810	1.1787	1.2393	1.3668
	Burst	402,420	25.00	1.1137	1.1088	1.2935	1.3426	1.4796
	Non-burst	1,207,051	75.00	0.9481	0.9465	1.1016	1.1452	1.2261
	Burst, $\rho_t < 0.25$	3,190	0.20	1.0996	1.0861	1.2934	1.3584	1.4281
	Gate-affected union	8,877	0.55	0.9928	0.9914	1.1594	1.2454	1.4281
	Unaffected	1,600,594	99.45	0.9895	0.9809	1.1786	1.2393	1.3668
Validation	All	170,942	100.00	1.0130	1.0040	1.1994	1.2557	1.4086
	$\rho_t < 0.15$	1,561	0.91	1.0233	0.9998	1.2270	1.3818	1.3818
	$\rho_t \geq 0.15$	169,381	99.09	1.0129	1.0041	1.1989	1.2546	1.4086
	Burst	42,297	24.74	1.1355	1.1307	1.3277	1.3833	1.5496
	Non-burst	128,645	75.26	0.9727	0.9689	1.1347	1.1786	1.2506
	Burst, $\rho_t < 0.25$	1,076	0.63	1.2068	1.1457	1.3818	1.4322	1.4322
	Gate-affected union	2,004	1.17	1.0685	1.1060	1.3277	1.3818	1.4322
	Unaffected	168,938	98.83	1.0123	1.0031	1.1973	1.2543	1.4051
Test	All	170,942	100.00	1.0093	1.0023	1.2037	1.2610	1.3821
	$\rho_t < 0.15$	1,288	0.75	0.9574	0.9340	1.1251	1.4159	1.4159
	$\rho_t \geq 0.15$	169,654	99.25	1.0097	1.0024	1.2037	1.2610	1.3695
	Burst	38,648	22.61	1.1464	1.1351	1.3185	1.3667	1.5869
	Non-burst	132,294	77.39	0.9693	0.9722	1.1312	1.1777	1.2447
	Burst, $\rho_t < 0.25$	730	0.43	1.2201	1.1132	1.4781	1.4781	1.4781
	Gate-affected union	1,637	0.96	1.0233	0.9860	1.4159	1.4781	1.4781
	Unaffected	169,305	99.04	1.0092	1.0023	1.2031	1.2564	1.3657
Generated	All	11,517	100.00	0.9248	0.9195	1.0970	1.1524	1.2566
	$\rho_t < 0.15$	180	1.56	0.9022	0.8890	1.0742	1.1409	1.2460
	$\rho_t \geq 0.15$	11,337	98.44	0.9251	0.9199	1.0972	1.1524	1.2566
	Burst	815	7.08	1.0915	1.0848	1.2484	1.2918	1.4174
	Non-burst	10,702	92.92	0.9121	0.9102	1.0725	1.1228	1.2098
	Burst, $\rho_t < 0.25$	16	0.14	1.1064	1.1252	1.2304	1.2585	1.3148
	Gate-affected union	185	1.61	0.9069	0.8901	1.0993	1.1503	1.2434
	Unaffected	11,332	98.39	0.9251	0.9198	1.0970	1.1524	1.2566

Table 28: Case1 joint subgroup versus unaffected transitions. Mean increase is relative to the unaffected mean; tail excess is the difference in exceedance probabilities, in percentage points (pp). Exceedance uses the unaffected q95 within each split, fixed across resamples. Brackets are 95% percentile intervals from 1000 cluster resamples. Clusters are complete source windows (seed 3407) or disjoint blocks of 32 adjacent stored windows (seed 3408), retaining a final short block. Supported clusters contain at least one joint transition.

Split	Resampling unit	Clusters	Supported	Mean increase (%)	Tail excess (pp)
Train	Window	12,673	2923	11.1 [10.7, 11.6]	11.1 [9.9, 12.4]
Train	32-window block	397	145	11.1 [9.0, 13.4]	11.1 [4.9, 18.0]
Validation	Window	1,346	822	19.2 [18.6, 19.9]	24.5 [22.3, 26.7]
Validation	32-window block	43	32	19.2 [15.8, 23.3]	24.5 [12.8, 37.1]
Test	Window	1,346	514	20.9 [19.6, 22.3]	42.3 [38.2, 46.6]
Test	32-window block	43	25	20.9 [13.1, 29.4]	42.3 [17.1, 67.7]

B.9.1 Single-tube objective and gradient comparison

The one-regime prior retains a fitted operator and residual geometry. In a post-submission audit, we compare its squared tube objective with velocity and acceleration penalties on 4096 deterministically selected three-frame validation windows in each of Case2 and Case3. Windows are formed within the stored aligned sequences, then selected at evenly spaced indices from the flattened list. They are clean latent windows, not noisy denoiser outputs or independent training runs. The aligned input dimension is 256 in both cases; LSKP uses the fixed PCA dimensions 48 and 16, respectively.

LSKP scores the final transition of each window after projection, averaging squared residuals divided by the stored residual variances, floored at 10^{-6}. This diagnostic uses the stored variance floor

Table 29: Single-regime objective/gradient audit, using 4096 selected validation windows per case. Score columns give Pearson/Spearman correlations with the LSKP score. Cosine columns summarize the empirical distribution of input-gradient alignment; q05/q95 are window-distribution quantiles, not confidence bounds. The final column is the median comparator/LSKP gradient-norm ratio under the stated objective normalizations.

		Score correlation		Gradient cosine				Norm ratio median
Case	Comparator	Pearson	Spearman	Mean	Median	q05	q95	
Case2	Velocity	0.6387	0.5754	0.2504	0.2489	0.1979	0.3043	0.1307
Case2	Acceleration	0.5725	0.5341	0.1457	0.1478	0.0821	0.2038	0.2328
Case3	Velocity	0.1633	0.1957	0.1084	0.1102	0.0437	0.1661	0.0067
Case3	Acceleration	0.0795	0.1196	0.0068	0.0066	-0.0223	0.0391	0.0050

Table 30: Additional generic transition-regularizer sensitivity on Case1. Values are window means with 95% intervals over pooled rollout seeds 42/43/44 at $T = 3840$. Both variants use the SN-CNF latent chart and replace the LSKP transition-tube score with a generic predicted-clean latent smoothness penalty. Lower is better; numerical minima are not treated as significance claims.

Variant	$\mathcal{W}_1^{\mathrm{time}}(1)$	$\mathcal{W}_1^{\mathrm{time}}(16)$	ΔT_{int}	$\mathrm{Err}_{\log E, L_2}$
Velocity	0.124 [0.113,0.137]	0.529 [0.471,0.589]	0.160 [0.116,0.203]	0.140 [0.133,0.146]
Acceleration	0.0538 [0.0481,0.0595]	0.465 [0.408,0.538]	0.210 [0.166,0.249]	0.114 [0.109,0.119]

without the additional training whitening stabilizer. Velocity and acceleration penalties average squared first and second differences over the aligned latent coordinates. All gradients are taken with respect to the same three-frame aligned input and flattened across frames and coordinates for cosine comparison. The norm ratio is the comparator gradient norm divided by the LSKP gradient norm. These are gradients of the uncoupled scalar objectives, before diffusion-noise gating or the training coefficient is applied.

Table 29 shows that the tested single-tube objective has different local gradient directions from the two smoothing penalties. These directions jointly reflect projection, the fitted transition operator, and residual scaling; the norm ratios additionally depend on objective normalization. The comparison establishes non-equivalence for the evaluated objectives, not a downstream performance ranking. The trained Case1 controls in Table 30 assess rollout effects separately.

B.9.2 Additional Case1 sensitivity checks

We additionally run two Case1 sensitivity checks to test whether the mechanism evidence is robust to nearby alternatives. These checks use the same Case1 final-horizon rollout length $T = 3840$, matched rollout seeds 42, 43, and 44, and pooled bootstrap summaries with $N_{\mathrm{win}} = 24$ evaluation windows. They are not used for checkpoint selection, are not part of the frozen main three-model protocol, and do not replace Table 1.

The first check asks whether LSKP can be reduced to a generic predicted-clean smoothness penalty. We train two SN-CNF-chart diffusion variants that replace the LSKP tube score with either a predicted-clean latent velocity penalty or a predicted-clean latent acceleration penalty, using the same 255k training budget and validation-side best-checkpoint rule. Table 30 shows that generic smoothing does not reproduce the full Case1 LSKP pattern. The acceleration penalty can improve pointwise fixed-lag transport on Case1, including the lag-1 row, so it is a meaningful stress test rather than a straw baseline. However, under the same pooled Case1 protocol, both generic penalties leave the temporal-memory gap larger than the $R = 8$ LSKP-final anchor in Table 31. Thus the useful signal is not simply "make predicted-clean latents smoother"; the calibrated stochastic transition tube and selective coupling remain different from a generic smoothness prior.

The second check varies the number of local regimes in the Case1 LSKP-style prior. We train otherwise matched sensitivity variants with $R \in \{1, 2, 4, 16\}$ regimes, again using the same 255k budget and validation-side best-checkpoint rule, and evaluate the same pooled three-seed rollouts. Table 31 also includes the frozen main $R = 8$ LSKP-final configuration as an anchor, evaluated under the same Case1 final-horizon pooled protocol but not retrained as part of this sweep. The table shows that the number of regimes matters: $R = 2$ is strongest on the two fixed-lag temporal-transport rows,

Table 31: Additional Case1 regime-count sensitivity for LSKP-style transition priors. Values are window means with 95% intervals over pooled rollout seeds 42/43/44 at $T = 3840$. The $R = 8$ row is the frozen main LSKP-final Case1 configuration evaluated under the same pooled protocol; it is included as an anchor and is not a newly retrained sweep member. The table reports the claim-facing transition, memory, and spectral-guardrail diagnostics; tail and derivative-guardrail rankings are discussed in the text. Lower is better; numerical minima are not treated as significance claims.

Regimes	$\mathcal{W}_1^{\mathrm{time}}(1)$	$\mathcal{W}_1^{\mathrm{time}}(16)$	ΔT_{int}	$\mathrm{Err}_{\log E, L_2}$
$R = 1$	0.0569 [0.0506,0.0637]	0.468 [0.412,0.520]	0.216 [0.172,0.263]	0.124 [0.119,0.128]
$R = 2$	0.0502 [0.0441,0.0601]	0.372 [0.322,0.426]	0.180 [0.138,0.222]	0.125 [0.119,0.132]
$R = 4$	0.0523 [0.0464,0.0593]	0.467 [0.396,0.545]	0.121 [0.0820,0.163]	0.113 [0.109,0.116]
$R = 8$ (final)	0.100 [0.0927,0.108]	0.468 [0.418,0.523]	0.136 [0.0972,0.175]	0.131 [0.121,0.142]
$R = 16$	0.0612 [0.0530,0.0686]	0.516 [0.456,0.581]	0.159 [0.118,0.202]	0.119 [0.111,0.127]

Table 32: Audited training cost for the three reported model groups. Run GPU-hours are computed from TensorBoard scalar wall time under a one-GPU-per-run assumption; filesystem modification times are not used. Because reported runs have different latest saved steps, matched-step training cost should be read from the hours-per-100k-steps and relative-step-time columns, with relative step time normalized to Origin-CoNFiLD within each case. Peak memory is an approximate profiled value, measured as PyTorch peak reserved memory in matched four-step short runs on NVIDIA RTX 5880 Ada GPUs with checkpointing and SwanLab logging disabled. All methods use the standard sampling pipeline; LSKP-final introduces no inference-time correction.

Case	Model	Run GPU-h	h/100k steps	Rel. step time	Profiled peak mem. (GiB)
Case1	Origin-CoNFiLD	87.9	8.79	1.00×	16.5
Case1	SN-CNF	48.2	13.38	1.52×	16.5
Case1	LSKP-final	33.8	13.27	1.51×	16.5
Case2	Origin-CoNFiLD	85.8	24.18	1.00×	40.6
Case2	SN-CNF	48.1	26.01	1.08×	40.6
Case2	LSKP-final	82.6	26.21	1.08×	40.6
Case3	Origin-CoNFiLD	123.3	24.67	1.00×	40.6
Case3	SN-CNF	38.7	24.96	1.01×	40.6
Case3	LSKP-final	82.3	30.50	1.24×	40.6

whereas $R = 4$ is strongest on the temporal-memory and log-spectrum rows. Very high fragmentation is not uniformly helpful: $R = 16$ has the lowest ΔG_2 and $\Delta\Omega_2$ among the sweep variants, but it worsens lagged temporal transport. These secondary guardrails use the definitions in Appendix B.14. The nearby regime counts outperforming the frozen $R = 8$ anchor on individual diagnostics is a limitation of validation-side prior selection rather than a replacement protocol. It shows that Case1 is sensitive to how the prior partitions the descriptor space and that no single fixed regime count is uniformly best across reporting roles. We therefore use this sweep as sensitivity evidence for an intermediate-regime design philosophy and as motivation for adaptive regime-count selection in future priors, not as a replacement for the frozen main protocol.

B.10 Training resources

Resources for the additional Case3 training runs. The four additional diffusion-training runs in Table 21 each use one GPU, with hardware, completed optimization budgets, and logged training time listed in Table 33. We retain the accounting of Table 32: each TensorBoard event file contributes the elapsed time between the first and last observations of its scalar series with the widest step span. This measures logged process wall time, including validation, logging, and steps repeated after restarts. Interprocess downtime, initialization, and any unlogged final steps are excluded; filesystem modification times are not used.

The four additional runs account for approximately 312.5 logged GPU-hours. The two cost tables cover diffusion training for the nine original comparison runs and these four additional runs. CNF fitting, offline prior calibration, rollout evaluation, and exploratory configurations require additional compute outside this accounting. At test time, all reported model groups sample latent trajectories

Table 33: Resources for the four additional Case3 diffusion-training runs. Each run uses one device. Memory is nominal device capacity, not measured peak allocation. Logged GPU-hours sum within-process scalar timestamp spans, including repeated steps after restarts; the optimization budget refers to the completed training trajectory.

Model	Training seed	GPU	Memory (GB)	Budget	Logged GPU-h
SN-CNF	1042	A40	48	155k	59.7
SN-CNF	2042	A40	48	155k	58.6
LSKP-final	1042	RTX A6000	48	270k	89.8
LSKP-final	2042	A40	48	270k	104.4

Table 34: Evaluation units at the final horizons. All lengths and strides count frames. The two trajectory/window columns give generated trajectories followed by evaluation windows per model. Each case uses one DNS reference window. Rollout seeds control generation; they are distinct from training initialization and the evaluator random seed.

Case	Training frames	Rollout frames	Evaluation frames	Stride	Seed 42 trajectories/windows	Seeds 42/43/44 trajectories/windows
Case1	128	3840	1473	368	1/8	3/24
Case2	256	768	280	70	3/24	9/72
Case3	256	1536	288	72	1/19	3/57

with the standard reverse diffusion sampler and decode them with the inherited CNF decoder; LSKP-final introduces no projection, correction, guidance, or sampler modification.

B.11 Statistical reporting and horizon diagnostics

Windows, horizons, and sample counts. Table 34 distinguishes diffusion-training windows from complete generated rollouts and evaluation windows. For each case, the final-horizon Protocol-C+ DNS reference is one held-out segment. Generated trajectories are divided into windows of that segment's length, at the reported stride; an additional window ending at the final frame is included if the stride does not land there exactly. All windows are retained. This yields overlapping windows with matched DNS/generated lengths, rather than one-to-one trajectory pairs. At shorter horizons, the evaluation length is the smaller of the DNS segment and the generated trajectory, and the default stride is one quarter of that length, rounded down.

Aggregate estimates and window summaries. The compact main comparison uses aggregate point estimates from rollout seed 42. Appendix Protocol-C+ interval summaries pool rollout seeds 42/43/44 for the same frozen model. For temporal transport, an aggregate estimate compares the pooled DNS and generated distance laws; a window-level value compares one generated window's distance law against the pooled DNS reference. For memory, an aggregate estimate integrates the averaged ACF, whereas a window-level gap uses that generated window's integral time against the DNS reference integral time. The reported window mean is the arithmetic mean of these individual values. Pooling before applying Wasserstein distance or ACF truncation generally differs from averaging the resulting window metrics. Differences from the main table therefore reflect both the generation pool and the aggregation order, not a bootstrap correction to the same point estimate. Protocol-E summaries use windows within each event group, with event-specific counts.

For each metric, the default bootstrap routine removes non-finite window values, resamples the remaining N_{win} values with replacement 200 times using seed 42, and recomputes their mean on each draw. The interval endpoints are the 2.5th and 97.5th percentiles of these bootstrap means; the reported center is the original arithmetic window mean. As a robustness audit, we also recomputed the reported rows with 1000 resamples; the means are unchanged, and endpoint changes are modest relative to the coverage limitations discussed below. These are descriptive intervals over evaluation windows, including overlapping windows from the same rollout; they do not estimate independent training-run or hierarchical trajectory-level uncertainty.

The additional Case3 summaries in Table 22 instead report variation across three trained diffusion models per method, using the mean and sample standard deviation of their run-level window means.

Table 35: Supplementary uniform four-diagnostic horizon manifest. The table expands the four diagnostics in Table 1 across all evaluated horizons. Columns are grouped by case, with horizons shown underneath each case heading; bold marks the best model within each case, diagnostic, and horizon. The final horizon in each case matches the main comparison in Table 1. Lower is better for all diagnostics. Event-conditioned diagnostics and physical-fidelity guardrails beyond these four columns are reported separately in Figure 9 and Table 36.

		Case1				Case2				Case3					
Diagnostic	Model	1920	2560	3200	3840	256	384	512	768	256	384	512	768	1024	1536
$\mathcal{W}_1^{\mathrm{time}}(\ell{=}1)$	Origin-CoNFiLD	**0.0323**	**0.0204**	**0.0305**	**0.0314**	**0.544**	0.626	0.643	0.551	5.88	5.81	5.87	5.86	5.89	5.80
	SN-CNF	0.0535	0.0478	0.0558	0.0381	0.839	0.621	**0.554**	0.622	5.61	5.63	5.67	5.58	5.62	5.65
	LSKP-final	0.0785	0.0872	0.0921	0.0898	0.878	**0.487**	0.567	**0.524**	**0.0211**	**0.0163**	**0.0139**	**0.0452**	**0.0289**	**0.0426**
$\mathcal{W}_1^{\mathrm{time}}(\ell{=}16)$	Origin-CoNFiLD	**0.238**	**0.370**	**0.315**	0.356	0.727	0.571	**0.518**	0.664	3.07	3.36	3.28	3.35	3.42	3.34
	SN-CNF	0.269	0.377	0.426	**0.306**	**0.702**	0.619	0.533	0.519	2.91	3.22	3.16	3.11	3.10	3.17
	LSKP-final	0.394	0.445	0.388	0.402	0.735	**0.308**	0.613	**0.490**	**0.108**	**0.282**	**0.196**	**0.568**	**0.313**	**0.494**
ΔT_{int}	Origin-CoNFiLD	0.146	**0.0483**	0.0409	0.0532	**0.890**	0.913	0.884	0.883	0.977	0.980	0.980	0.980	0.980	0.978
	SN-CNF	**0.0099**	0.105	**0.0231**	0.0938	0.954	0.910	**0.762**	**0.843**	0.977	0.980	0.980	0.980	0.977	0.979
	LSKP-final	0.0964	0.127	0.0782	**0.0478**	0.950	**0.802**	0.887	0.859	**0.0881**	**0.129**	**0.219**	**0.429**	**0.335**	**0.383**
$\mathrm{Err}_{\log E, L_2}$	Origin-CoNFiLD	0.113	0.103	**0.0861**	0.0971	0.483	0.484	0.493	0.482	0.562	0.543	0.547	0.541	0.541	0.536
	SN-CNF	0.113	0.108	0.0875	0.120	0.440	0.438	**0.426**	**0.424**	0.450	0.432	0.450	0.462	0.481	0.478
	LSKP-final	**0.0910**	**0.0907**	0.0887	**0.0956**	**0.435**	**0.414**	0.428	0.429	**0.235**	**0.336**	**0.314**	**0.249**	**0.210**	**0.177**

They keep the CNF chart and fitted prior fixed. Table 23 uses 1000 window resamples at each fixed checkpoint and measures training-stage sensitivity on the main-comparison training runs, separately from the training-seed study.

Compact aggregate tables show point estimates; appendix diagnostics provide bootstrap summaries and evaluation-window counts where available. Appendix Figure 6 reports the representative horizon curves as aggregate point estimates for readability, and Figure 9 displays bootstrap intervals for the event-conditioned comparisons. Appendix Figure 7 reports window means and bootstrap intervals for the same case–diagnostic combinations, and Appendix Table 35 summarizes the complete three-case by four-diagnostic horizon manifest. These appendix diagnostics are used as sensitivity checks rather than separate model-selection criteria. Protocol-E intervals are descriptive for event groups with few evaluation windows.

Horizon comparisons. For the large Case3 lag-16 transition separation highlighted in the main comparison, the final-horizon bootstrap summary over evaluation windows is Origin-CoNFiLD $3.37\,[3.31, 3.43]$, SN-CNF $3.18\,[3.14, 3.23]$, and LSKP-final $0.520\,[0.426, 0.598]$, with $N_{\mathrm{win}} = 57$ windows for each model. This supports the targeted transition-side readout under the evaluation protocol, while still representing evaluation-window uncertainty rather than independent retraining-seed uncertainty.

In the pooled window summaries at $T = 1536$, Case3 LSKP-final has the lowest mean on all three primary transition and memory diagnostics. Table 36 also shows its lowest values on three of four field guardrails; the strain-rate gap is lower than SN-CNF but higher than Origin-CoNFiLD.

For Case2, pooled window means favor LSKP-final over SN-CNF on lag-1 transport and log-spectrum error, while lag-16 and memory favor SN-CNF. The main-table aggregate instead favors LSKP-final at both fixed lags. These summaries use different rollout pools and aggregation orders, as defined above.

For Case1, the pooled window means retain the memory improvement and fixed-lag transport trade-off relative to SN-CNF. Its spectral ranking relative to Origin-CoNFiLD changes from the seed-42 aggregate to the pooled window mean. Appendix B.14 reports this field-level comparison separately from the temporal diagnostics.

B.12 Fixed-checkpoint sensitivity checks

With model checkpoints held fixed, we examine similarity to training windows and sensitivity to longer rollout horizons. A nearest-neighbor memorization diagnostic found no near-exact duplicates between LSKP-final generated windows and training windows in either the full SN-CNF latent space or the LSKP coarse-observable space, using an RMS duplicate threshold of 10^{-4}. Generated-to-train nearest-neighbor distances are nonzero in all cases, with zero duplicates below threshold. In Case1

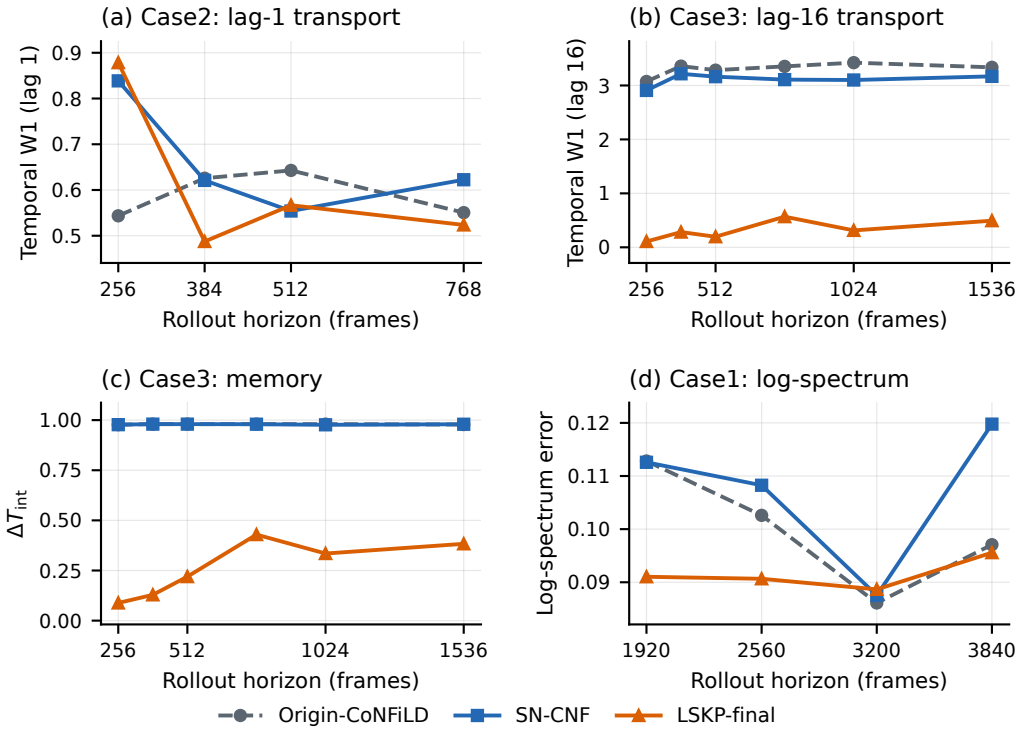

Figure 6: Horizon-wise aggregate comparison under Protocol-C+. Panels (a)–(d) show representative horizon-wise diagnostics aligned with the four reporting roles in Table 1: Case2 lag-1 transition $\mathcal{W}_1^{\text{time}}(\ell{=}1)$, Case3 $\mathcal{W}_1^{\text{time}}(\ell{=}16)$, Case3 ΔT_{int}, and Case1 $\text{Err}_{\log E, L_2}$. Curves are frozen-checkpoint aggregate point estimates, lower is better, and the complete case-by-diagnostic horizon manifest is reported in Appendix Table 35.

and Case2, generated-to-train medians are smaller than held-out test-to-train medians in both spaces; in Case3 the relation is mixed, smaller in the full latent space but larger in the coarse-observable space. We therefore interpret the diagnostic as evidence against literal copying, not as a complete mode-coverage guarantee.

For Case2, we additionally evaluate fixed-checkpoint rollouts at horizons 1024 and 1280 using the same checkpoints as the main comparison and matched rollout seeds 42, 43, and 44. Figure 8 is a horizon-sensitivity stress check rather than a checkpoint selection protocol or a replacement for the main Protocol-C+ comparison. Under this setting, LSKP-final is consistently lowest on the lag-1 transition diagnostic and on ΔT_{int} at both extended horizons, suggesting that its short-lag and memory-side effects persist beyond the reported final horizon. In contrast, $\mathcal{W}_1^{\text{time}}(\ell{=}16)$ and tail-window transport remain mixed across horizons and comparators. We therefore interpret this figure as targeted sensitivity evidence, not as evidence of uniform long-horizon transport dominance.

B.13 Event-conditioned decomposition

Protocol-E groups evaluation subwindows using an enstrophy-based activity score. Each DNS or generated subwindow is scored by the 95th percentile over time of its own frame-wise enstrophy proxy. The calm and burst thresholds are the 0.3 and 0.7 quantiles of the DNS score distribution. Scores at or below the calm threshold define *calm*; scores at or above the burst threshold define *burst*; intermediate scores define *typical*. Generated fields use the same DNS-derived thresholds. These event groups are distinct from the descriptor regimes used to fit LSKP.

For Figure 9, the tail-matched comparison retains the last 1473 frames of one generated trajectory per model and the corresponding DNS reference segment. Each segment is divided into overlapping 128-frame subwindows at stride 48, including a final window ending at the segment boundary. This gives

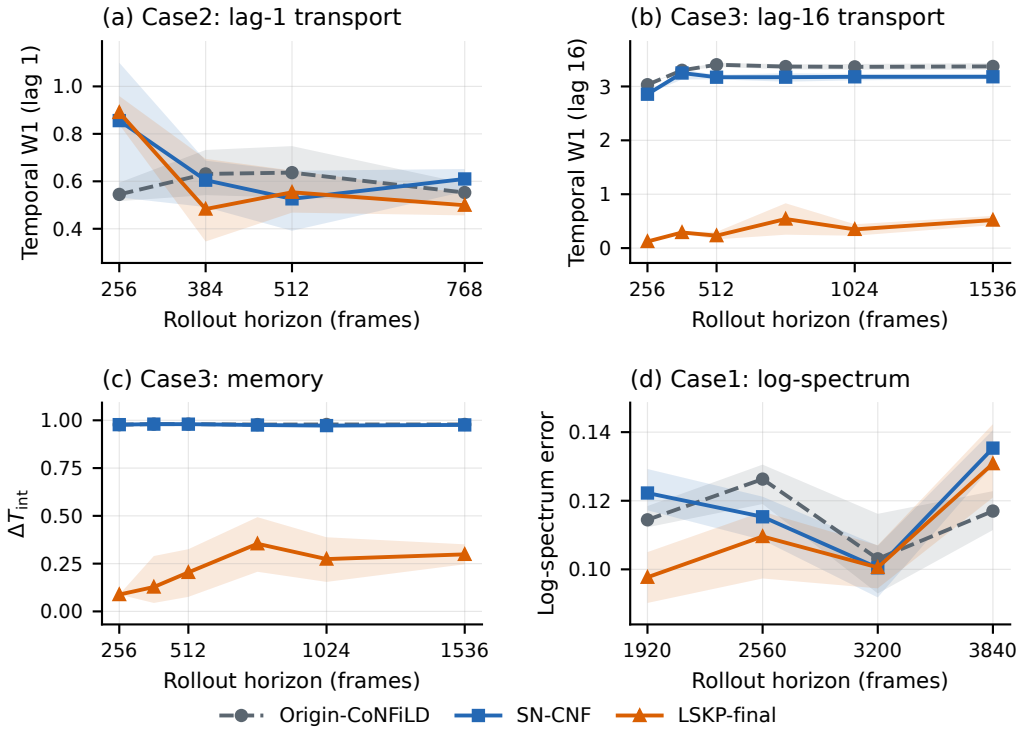

Figure 7: Window-mean summaries for the case–diagnostic combinations in Figure 6. Lines report window means and shaded regions report 95% bootstrap intervals over generated evaluation windows, with the DNS reference fixed. Aggregate point estimates are reported separately in the preceding figure. The Case3 $\mathcal{W}_1^{\text{time}}(\ell{=}16)$ and ΔT_{int} panels show the clearest transition-side and temporal separation, while the Case1 log-spectrum panel is a physical-fidelity guardrail under the same frozen-checkpoint protocol.

30 generated and 30 DNS subwindows per model comparison, rather than the pooled Protocol-C+ evaluation counts. A group is reported when both generated and DNS counts are at least five. The separate coverage audit flags counts below ten or an absolute occupancy gap of at least 0.10, so a displayed group can still carry a coverage warning. Intervals use 200 resamples of the generated subwindow values within each group, with evaluator seed 42 and the DNS group reference fixed.

The occupancy gap is the absolute difference between the generated and DNS fractions in an event group. Low within-event error and accurate occupancy measure different properties: one evaluates the fields once an event is visited, and the other evaluates its frequency. Event-conditioned evaluation therefore reports both conditional field quality and event coverage.

Figure 9 localizes Case1 log-spectrum errors across these event groups. Figure 4 reports the same groups' counts and occupancy gaps. Their descriptive intervals summarize overlapping subwindows within the fixed comparison, separately from the overall Protocol-C+ spectral and derivative guardrails below.

B.14 Physical-fidelity guardrails

The inherited CNF decoder maps generated latents to physical fields. Spectral and derivative diagnostics complement the primary temporal metrics by measuring the decoded fields' statistical fidelity.

We report one spectral and three derivative-sensitive guardrails at the final Protocol-C+ horizon. Let $E_{\text{gen}}(b)$ and $E_{\text{dns}}(b)$ denote velocity energy spectra in bin b, averaged over the frames selected by the spectrum evaluator for the scored unit. With B spectral bins and numerical floor $\varepsilon_{\text{diag}} = 10^{-8}$,

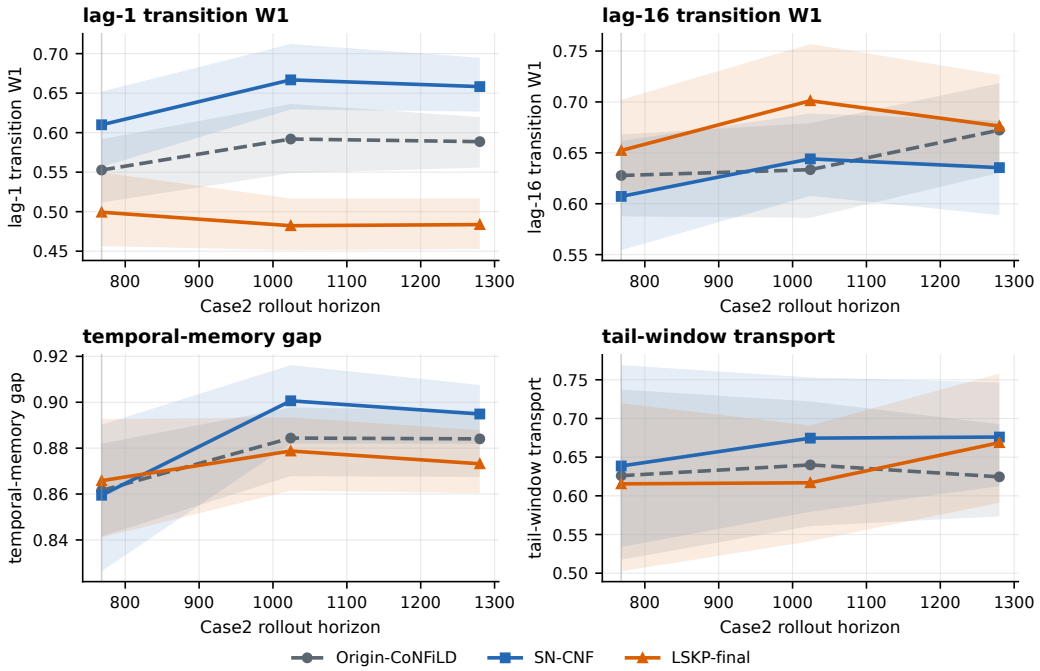

Figure 8: Case2 extended-horizon sensitivity check. Horizon 768 is the reported final Protocol-C+ horizon; horizons 1024 and 1280 use the same frozen checkpoints and matched rollout seeds 42, 43, and 44. Lines show window means and shaded regions show 95% bootstrap intervals over evaluation windows. The check probes fixed-checkpoint horizon sensitivity and is not used for checkpoint selection or as a replacement for the main Protocol-C+ comparison. The vertical lines mark the original Protocol-C+ horizon of 768. Lower is better for all panels.

the evaluator computes

$$\mathrm{Err}_{\log E,L_2} = \left(\frac{1}{B} \sum_{b=1}^{B} \left[\log(\max\{E_{\mathrm{gen}}(b), \varepsilon_{\mathrm{diag}}\}) - \log(\max\{E_{\mathrm{dns}}(b), \varepsilon_{\mathrm{diag}}\}) \right]^2 \right)^{1/2}.$$

For an aggregate, spectra are pooled before computing this discrepancy. For a window mean, each generated window is scored against the shared DNS reference spectrum, then the resulting errors are averaged. Thus the spectral summary distinguishes pooling spectra from averaging window errors.

For derivative-sensitive diagnostics, each case uses the same spatial derivative operator as its evaluator: local-neighbor gradient weights on unstructured Case1/Case3 geometries, and finite differences on the Case2 grid slice. For a generated or DNS evaluation pool, let

$$G_2 = \left\langle \|\nabla \mathbf{u}\|_F^2 \right\rangle, \qquad S_2 = \left\langle \|\mathbf{D}\|_F^2 \right\rangle, \qquad \Omega_2 = \left\langle \|\boldsymbol{\omega}\|_2^2 \right\rangle,$$

where $\mathbf{u}$ is velocity, $\mathbf{D}$ is the symmetric strain-rate tensor, and $\boldsymbol{\omega}$ is vorticity or the case-specific vorticity proxy. The brackets average over time and spatial sites in the scored unit, and over windows for aggregate quantities. The DNS-normalized absolute relative gaps are

$$\Delta G_2 = \frac{|G_{2,\mathrm{gen}} - G_{2,\mathrm{dns}}|}{|G_{2,\mathrm{dns}}| + \varepsilon_{\mathrm{diag}}},$$

$$\Delta S_2 = \frac{|S_{2,\mathrm{gen}} - S_{2,\mathrm{dns}}|}{|S_{2,\mathrm{dns}}| + \varepsilon_{\mathrm{diag}}},$$

and

$$\Delta \Omega_2 = \frac{|\Omega_{2,\mathrm{gen}} - \Omega_{2,\mathrm{dns}}|}{|\Omega_{2,\mathrm{dns}}| + \varepsilon_{\mathrm{diag}}}.$$

Lower values are better for all four guardrails. Table 36 reports window means with 95% window-bootstrap intervals for log-spectrum error, ΔG_2, and $\Delta \Omega_2$, holding the DNS reference fixed. The

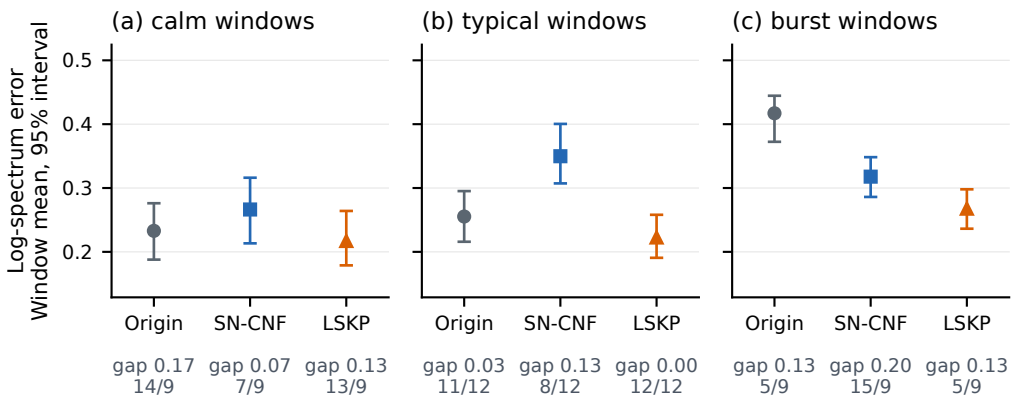

Figure 9: Case1 event-conditioned log-spectrum errors under Protocol-E. Calm, typical, and burst groups use DNS-referenced event-score quantiles. Points show generated-window means and bars show 95% bootstrap intervals, holding the DNS group reference fixed. The comparison uses the last 1473 frames of one trajectory per model, divided into overlapping 128-frame subwindows at stride 48; Appendix B.13 gives the full procedure. Lower annotations give the absolute event-occupancy gap and generated/DNS window counts. Origin and LSKP denote Origin-CoNFiLD and LSKP-final. This decomposition separates within-event field statistics from event coverage.

Table 36: CoNFiLD-style physical-fidelity guardrails at the final Protocol-C+ horizon. Log-spectrum error, ΔG_2, and $\Delta\Omega_2$ report window means with 95% window-bootstrap intervals; ΔS_2 is a pooled aggregate point estimate. Lower is better.

Case	Guardrail	Origin-CoNFiLD	SN-CNF	LSKP-final
Case1	$\mathrm{Err}_{\log E,L_2}$	**0.117 [0.112, 0.123]**	0.135 [0.130, 0.141]	0.131 [0.121, 0.142]
	ΔG_2	**0.0266 [0.0223, 0.0299]**	0.0315 [0.0236, 0.0382]	0.0329 [0.0246, 0.0430]
	ΔS_2	**0.0269**	0.0313	0.0305
	$\Delta\Omega_2$	**0.0264 [0.0227, 0.0302]**	0.0308 [0.0223, 0.0381]	0.0329 [0.0247, 0.0428]
Case2	$\mathrm{Err}_{\log E,L_2}$	0.489 [0.488, 0.490]	0.433 [0.430, 0.435]	**0.428 [0.426, 0.431]**
	ΔG_2	0.0903 [0.0883, 0.0926]	**0.0785 [0.0724, 0.0834]**	0.0797 [0.0756, 0.0838]
	ΔS_2	0.0836	**0.0671**	0.0699
	$\Delta\Omega_2$	0.0974 [0.0956, 0.0992]	0.0905 [0.0853, 0.0955]	**0.0901 [0.0866, 0.0935]**
Case3	$\mathrm{Err}_{\log E,L_2}$	0.541 [0.538, 0.544]	0.493 [0.481, 0.505]	**0.248 [0.219, 0.277]**
	ΔG_2	0.0800 [0.0786, 0.0811]	0.110 [0.107, 0.113]	**0.0310 [0.0266, 0.0362]**
	ΔS_2	**0.0204**	0.0757	0.0262
	$\Delta\Omega_2$	0.153 [0.151, 0.154]	0.152 [0.149, 0.155]	**0.0308 [0.0258, 0.0359]**

three cases contribute 24/72/57 generated windows per model, pooled over rollout seeds 42/43/44. The strain-rate gap ΔS_2 instead uses the pooled aggregate point estimate because the evaluator does not expose window-level strain entries.

LSKP-final is lowest in five of the twelve guardrail rows. In Case3, it is lowest on the spectral, gradient, and enstrophy summaries; its strain-rate gap improves over SN-CNF while Origin-CoNFiLD remains lower. In Case2, all four guardrails improve over Origin-CoNFiLD, with mixed ordering relative to SN-CNF. Origin-CoNFiLD is lowest on all four Case1 rows. In particular, the Case1 spectral ranking differs from the seed-42 aggregate in Table 1; Appendix B.11 defines the distinct pools and summary operations. Together, the guardrails distinguish changes in decoded-field statistics from the primary temporal improvements.

B.15 Qualitative decoded-field sanity check

Figures 10 and 11 provide qualitative decoded-field views for Case2 and Case3. These views complement the quantitative Protocol-C+ diagnostics and physical-fidelity guardrails with visual checks of the generated fields.

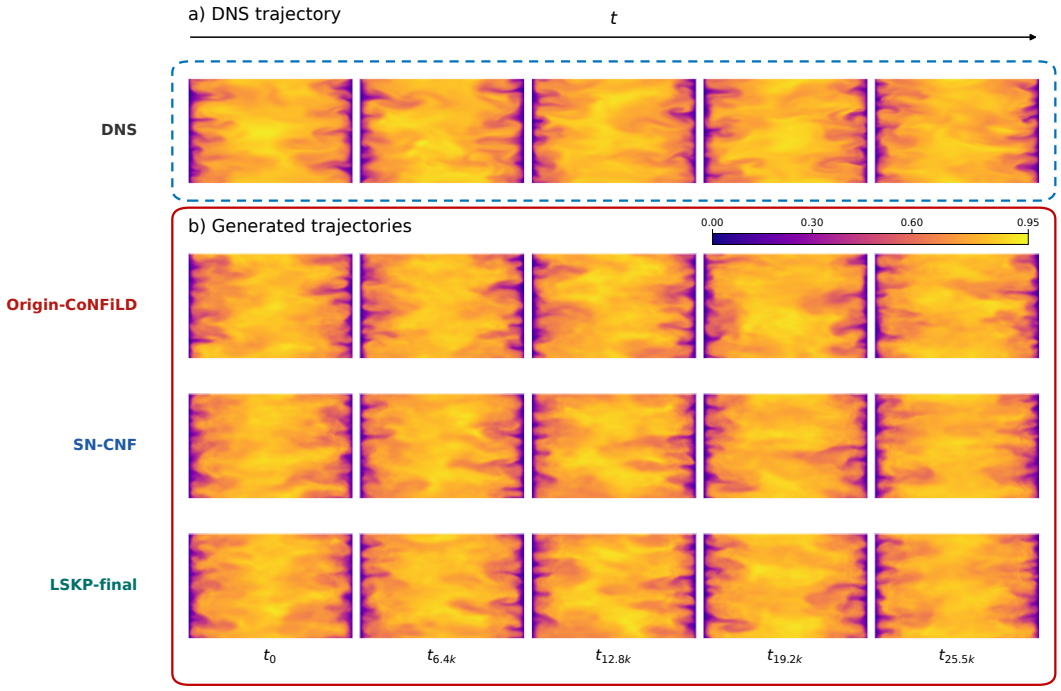

Figure 10: Representative Case2 decoded-field sanity check. Following the CoNFiLD visualization convention, the panels show the streamwise velocity component u on the Case2 grid. From left to right, the five displayed sequence labels correspond to zero-based frame indices $0, 64, 128, 192, 255$ in a 256-frame rollout. Time labels count DNS numerical steps ("k" denotes 1000), with 100 steps between adjacent frames. The DNS row uses held-out reference frames, while generated rows show the first unconditional trajectory from each model's frozen validation-side checkpoint at rollout seed 42; they are not paired to the DNS initial condition. All panels use common color limits from $u = 0$ to $u = 0.95$. The figure is qualitative only; quantitative claims are based on Protocol-C+, event-conditioned diagnostics, and physical-fidelity guardrails.

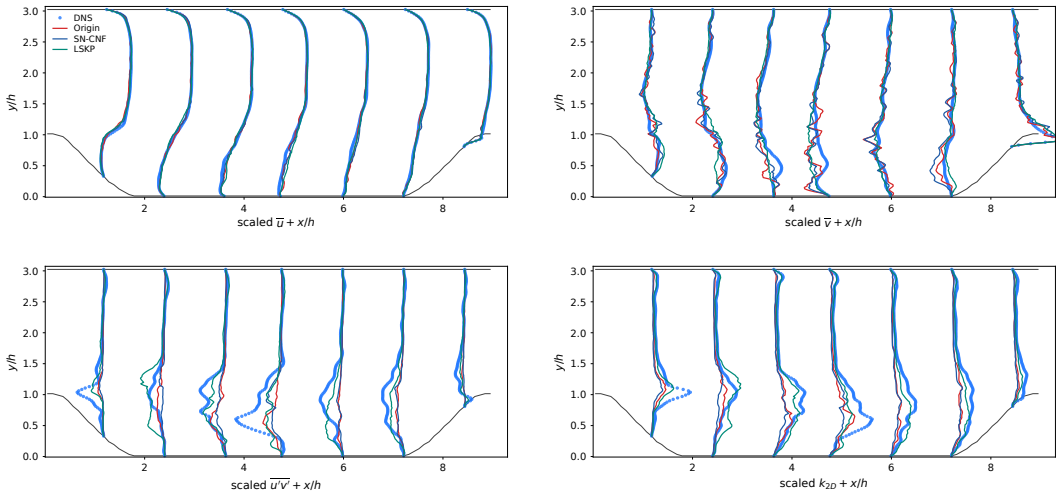

Figure 11: CoNFiLD-style Case3 long-horizon decoded-field profile sanity check. The figure reports profile statistics for DNS, Origin-CoNFiLD, SN-CNF, and LSKP-final at grid columns nearest to $x/h = 1.2, 2.4, 3.6, 4.8, 6.0, 7.2, 8.4$. For each model, the final 512 frames of a 1536-frame rollout (zero-based indices 1024–1535) are pooled over rollout seeds 42, 43, and 44, taking the first trajectory per seed. Means and fluctuations are computed after pooling; the DNS reference uses the same frame-index range. The profiles include mean velocity components, Reynolds shear $\overline{u'v'}$, and the resolved two-dimensional fluctuation-energy proxy k_{2D}. Profiles are horizontally offset by their station positions x/h. Within each panel, DNS and all models share a display scale set from the 98th percentile of absolute profile values across models and stations, including DNS. The figure is qualitative only; quantitative claims use Protocol-C+ diagnostics and physical-fidelity guardrails.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the proposed DNS-calibrated local stochastic transition closure, its training-time use, and its PDE-free scope. The claims are stated as targeted gains in transition-side and temporal diagnostics rather than uniform dominance across all metrics.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses the targeted operating regime, mixed outcomes against the SN-CNF comparator, dependence on prior coverage, and the fact that LSKP is not an inference-time correction mechanism. Appendix B further documents case-wise hyperparameter selection and aggregate-versus-bootstrap reporting limits.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4 states the assumptions and main theoretical statements for the local stochastic transition closure, and Appendix A provides formal proofs, scope clarifications, and diagnostic mappings for the theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 and Appendix B specify the backbone, objective, prior choices, training and rollout settings, primary metrics, checkpoint selector, and algorithmic pseudocode. Appendix B.5 gives the data source, preprocessing, exact temporal ranges, and window counts; the additional Case3 runs distinguish training seeds from rollout seeds. Open release is addressed separately in Item 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and usage instructions are available at `https://github.com/MAiTL-Group/CoNFiLD-LSKP`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the train/validation/test separation, model groups, evaluation protocols, metric roles, and checkpoint selection policy. Appendix B.5 reports optimizer, learning rate, batch size, EMA, validation cadence, checkpoint rule, offline prior fitting and hyperparameter selection, and LSKP coupling settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Appendix B.11 distinguishes aggregate estimates, evaluation-window bootstrap intervals, and variation across independently trained diffusion models. The additional audits specify their resampling units and counts separately; Table 22 reports sample standard deviations across three training runs per method, while the gradient audit reports empirical window quantiles rather than confidence intervals.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Table 32 reports logged GPU-hours, normalized step time, and peak memory from matched short profiling runs for the original comparisons. Appendix B.10 and Table 33 give the hardware, device memory, budgets, and logged times for the four additional Case3 training runs, and delimit the scope of this diffusion-training cost accounting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses simulated turbulence fields and does not involve human subjects, private personal data, surveillance, or safety-critical deployment. We have reviewed the NeurIPS Code of Ethics and are not aware of a violation.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is a methodological study for PDE-free turbulence generation and evaluation. It does not introduce a direct deployment system, human-facing decision process, or high-risk societal application.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk dual-use model, scraped dataset, human data, or content-generation system requiring safeguards of the type described in the question.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets are credited through citation. The CoNFiLD article is cited as Du et al. [2024]; its data are deposited on Zenodo under DOI 10.5281/zenodo.14037782 with Creative Commons Attribution 4.0 International licensing, and the archived Zenodo software record is deposited under DOI 10.5281/zenodo.14058397 with Creative Commons Attribution 4.0 International licensing. The corresponding GitHub repository is released under the Apache-2.0 license. The article itself is licensed under Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code repository includes a README, configuration files, environment requirements, and license information.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing, annotation by human workers, or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects, so IRB or equivalent human-subjects approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core method, theory, experiments, or scientific conclusions. Any language-model assistance, if used, was limited to writing, editing, or formatting support.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.