TEMPO-PDE: TRAJECTORY-AWARE POST-TRAINING FOR PDE FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Partial differential equation foundation models (PDE-FMs) are neural predictors pretrained across physical systems to learn reusable representations of PDE dynamics. However, accurate early predictions do not ensure that an adapted model remains accurate over the full forecast horizon. Reference trajectories provide ground-truth fields at multiple future times, but the corresponding loss gradients can differ in magnitude or conflict with one another. To address this problem, we propose TEMPO-PDE, a post-training framework with two complementary components. (i) Trajectory-aware supervision scores future predictions using field-value errors, spectral discrepancies, and local spatial differences, and organizes these measures into primary and auxiliary objectives. (ii) Primary-guided gradient composition removes opposing auxiliary components and bounds the remaining auxiliary norm relative to the primary gradient. Both components update the existing predictor without adding an inference module. Within the registered main comparison, TEMPO achieves the lowest observed time-averaged and final-step errors on the five main PDE benchmarks. Anonymous code: https://anonymous.4open.science/r/682426/.

1 INTRODUCTION

Partial differential equation foundation models (PDE-FMs) learn reusable predictive knowledge from trajectories of different physical systems (Herde et al., 2024; Hao et al., 2024; McCabe et al., 2024; 2025). They build on neural operators, which learn mappings from problem inputs to solution fields (Lu et al., 2021; Li et al., 2021a; Kovachki et al., 2023). Given observed fields such as velocity or pressure, an adapted predictor forecasts how these spatial distributions evolve. Learning this evolution supports the use of neural models in fluid simulation, weather forecasting, and climate modeling, where predictions must follow changing physical fields over time (Kochkov et al., 2021; Lam et al., 2023; Kochkov et al., 2024). For a new target system, post-training uses a limited set of reference trajectories. Each is a time-ordered sequence of ground-truth spatial fields from one simulation. A trajectory therefore supplies several future states against which predictions can be trained and evaluated. Our question is how to organize this supervision so that adapting the pretrained predictor improves the requested forecast as a whole, not only its earliest predictions.

Research on reliable PDE prediction has advanced along two complementary directions. The first develops predictors that reuse information across physical systems. Among pretrained models, Poseidon learns time-conditioned predictions from all-to-all snapshot pairs, while DPOT combines Fourier attention with autoregressive denoising (Herde et al., 2024; Hao et al., 2024). MPP, PDE-Transformer, and Walrus extend this direction to more heterogeneous systems, variables, and discretizations (McCabe et al., 2024; Holzschuh et al., 2025; McCabe et al., 2025). The second direction improves continued prediction. Message Passing Neural PDE Solvers use pushforward training to expose a model to predicted inputs, while PDE-Refiner uses iterative denoising to recover less dominant frequency components (Brandstetter et al., 2022; Lippe et al., 2023). PITA further studies physics-informed temporal alignment for pretrained predictors (Zhu et al., 2025). These approaches provide both reusable models and ways to learn from future states. However, these approaches do not explicitly couple temporal and spatial forecast supervision with joint control of gradient conflicts and relative magnitudes during target-task post-training. This motivates jointly designing forecast objectives and their gradient contributions to improve the full forecast. Figure 1 examines these choices using fixed Poseidon-B adaptation configurations on NS-PwC. Both panels report the per-

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

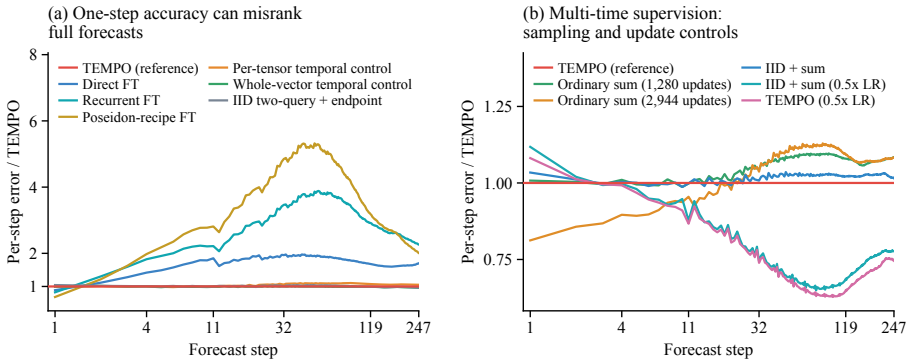


Figure 1: **Early accuracy and full-forecast quality can disagree.** Poseidon-B on NS-PwC, evaluated on 96 development trajectories. Ratios use TEMPO at the original learning rate. **(a)** Fine-tuning (FT) and temporal-supervision controls retain their own training recipes. **(b)** Ordinary sum keeps TEMPO's targets, with 1,280 or 2,944 updates. Independent and Identically Distributed (IID) and half-learning-rate configurations provide separate sampling and optimization controls.

step prediction error relative to TEMPO: a ratio below one favors the comparator, whereas a ratio above one favors TEMPO. Two observations follow:

❶ **Early accuracy is an incomplete guide to the full forecast.** The three fine-tuning controls in Figure 1a have lower error at the first prediction step, but their errors subsequently exceed TEMPO's. Figure 1b provides a longer-lasting example: the ordinary-sum control trained for 2,944 updates has 9.4% lower Mean error over the first 11 steps, yet 8.0% higher Mean error over all 247 steps. Thus, evaluating only the next state or a short prefix can favor a configuration that is less accurate over the requested window.

❷ **Supervising the same forecasts does not fully specify how they influence learning.** The ordinary-sum control in Figure 1b retains TEMPO's forecast targets, scores, and original learning rate, but sums the losses before applying global gradient clipping once. With 1,280 updates, its Mean error differs from TEMPO's by less than 0.1% over the first 11 steps, yet is 7.5% higher over all 247 steps. Extending the training for the sum control to 2,944 updates improves short-window accuracy without closing this full-window gap. The matched comparison concerns composition and clipping, while the additional IID and half-learning-rate curves examine separate sampling and optimization choices.

These observations suggest that adapting a predictor requires specifying both the forecast targets and their influence on learning. Choosing future times determines where temporal supervision is applied, but each future state is itself a spatial field. The objective must therefore also specify how errors within that field are measured. PDE-Refiner motivates attention to less dominant frequency components, while Sobolev training illustrates how derivative information can supplement target values (Lippe et al., 2023; Czarnecki et al., 2017). Adding these scores or forecast targets can produce competing parameter updates (Yu et al., 2020; Liu et al., 2021). The design of supervision should therefore be connected to how it influences learning. We ask: *How can target trajectories guide post-training to improve a pretrained PDE model's forecasts across time and space?*

Our approach. To answer this question, we propose TEMPO-PDE, a post-training framework that learns from forecasts produced through the model's existing interfaces. TEMPO compares each prediction with its reference field using field-value, spectral, and spatial-difference scores. It then organizes these scores into a primary prediction objective and complementary auxiliary supervision. For direct queries, the auxiliary target is the training-window endpoint; for recurrent prediction, it consists of direct forecasts from the same observed starting state. The primary gradient provides the reference direction and scale for combining the two objectives. TEMPO removes opposing auxiliary components and bounds the remaining auxiliary contribution before updating the predictor. Scoring and gradient processing occur only during training. Inference uses the adapted predictor.

Our contributions are threefold:

- **Trajectory-aware supervision for model adaptation.** We formulate post-training supervision across future times and spatial structures. Using the predictor's existing direct-query or recurrent interface, TEMPO constructs primary and auxiliary forecast objectives from reference trajectories without introducing a new architecture.
- **Primary-guided gradient composition.** We treat additional forecast losses as auxiliary to the main prediction objective. Using the primary gradient as the reference direction and scale, TEMPO removes opposing auxiliary components and bounds the remaining auxiliary norm.
- **Evidence for forecast gains and their sources.** In the main comparison, TEMPO achieves the lowest Mean and Final errors on the five main tasks. Ninety matched runs examine auxiliary supervision, structure scoring, and gradient composition in direct and recurrent training. Residual spectra locate spatial error changes, while Flower-Tiny experiments test score transfer.

2 Related Work

From neural operators to reusable PDE predictors. Neural operators learn mappings from problem inputs to solution fields, with DeepONet and FNO providing influential formulations (Lu et al., 2021; Li et al., 2021a; Kovachki et al., 2023). Subsequent work broadens the spatial interactions and geometries that these mappings can represent. CNO develops convolutional operators, while GINO connects irregular geometries to regular latent grids (Raonić et al., 2023; Li et al., 2023b). For larger spatial systems, Transolver and Universal Physics Transformers reduce computational complexity through compressed representations (Wu et al., 2024; Alkin et al., 2024). For reuse across systems, PDE-Transformer addresses heterogeneous variables, PROSE-FD combines symbolic and numerical inputs, and Zebra conditions predictions on example trajectories (Holzschuh et al., 2025; Liu et al., 2024; Serrano et al., 2025). These developments expand the fields and task information that a predictor can handle. TEMPO studies the subsequent adaptation step, where target trajectories determine how the available predictor is trained for a particular system.

From trajectory supervision to coordinated updates. Continued forecasting depends on both prediction mechanisms and training signals. Pushforward training exposes a predictor to its own outputs, while PDE-Refiner uses iterative denoising to recover less dominant frequency components (Brandstetter et al., 2022; Lippe et al., 2023). Flowers introduces learned warping into the prediction mechanism (Muser et al., 2026). Additional supervision includes equation constraints in PINNs and PINO, derivative information in Sobolev training, and trajectory-based temporal alignment in PITA (Raissi et al., 2019; Li et al., 2021b; Czarnecki et al., 2017; Zhu et al., 2025). Because these criteria act on shared model parameters, their gradients may differ in magnitude and direction (Wang et al., 2021). GradNorm addresses magnitude imbalance, while PCGrad and CAGrad address competing update directions (Chen et al., 2018; Yu et al., 2020; Liu et al., 2021). TEMPO specifies both which forecast errors to supervise and how their losses update the predictor. Unlike these task-symmetric methods, it treats one objective as primary, removing opposing auxiliary components and bounding the auxiliary norm relative to the primary gradient.

3 Methodology

3.1 Problem Setting and Overview

We adapt a pretrained predictor using target-system trajectories. x_0 is the observed field and x_h its reference h steps later, with time spacing Δt and C dynamic variables on a spatial grid. All parameters of F_θ are fine-tuned from pretrained weights θ_0. Known inputs such as forcing are retained but omitted below. Figure 2 shows two prediction paths:

$$\hat{x}_h = F_\theta(x_0, h\Delta t), \qquad \hat{x}_k^R = F_\theta(\hat{x}_{k-1}^R, \Delta t), \qquad \hat{x}_0^R = x_0. \tag{1}$$

Direct queries use x_0 at each requested time, while recurrent forecasts use preceding predictions. Following Figure 2, $\hat{x}_h$ denotes a direct output, also written $\hat{x}_k^D$ when paired with a recurrent output $\hat{x}_k^R$. All calls share F_θ.

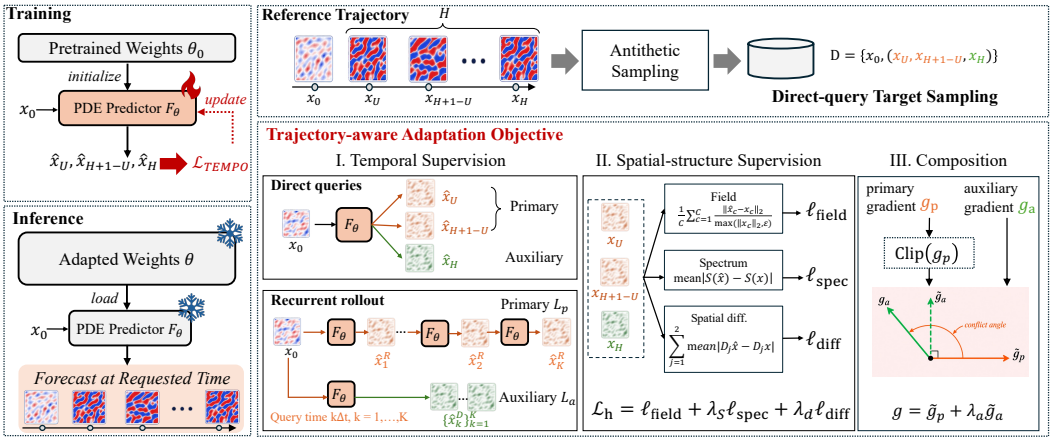


Figure 2: **TEMPO-PDE post-training.** One predictor supplies both primary and auxiliary forecasts. Direct training pairs sampled queries with endpoint supervision. Recurrent training uses a rollout and field-only direct queries from x_0. The primary gradient is clipped before the auxiliary gradient is projected, capped, and added. These operations are used only during training.

Forecast scores form a primary loss L_p for the main prediction path and an auxiliary loss L_a for complementary forecasts. Algorithm 1 combines their gradients to update the predictor. In Figure 2, $\mathcal{L}_{\text{TEMPO}}$ denotes this training procedure, not a plain sum of losses.

3.2 Trajectory-aware Supervision

Scoring a predicted field. The field score divides each variable's prediction error by its reference magnitude and averages the resulting relative errors:

$$\ell_{\text{field}}(\hat{x}, x) = \frac{1}{C} \sum_{c=1}^{C} \frac{\|\hat{x}_c - x_c\|_2}{\max(\|x_c\|_2, \epsilon)}. \tag{2}$$

Norms cover spatial grid points. Separate normalization balances variables of different magnitudes. Two further terms compare spatial structure, starting with logarithmic power spectra:

$$S(z) = \log\left(|\,\text{rFFT2}_{\text{ortho}}(z)|^2 + \epsilon\right), \qquad \ell_{\text{spec}}(\hat{x}, x) = \text{mean}\,|S(\hat{x}) - S(x)|. \tag{3}$$

The transform acts on both spatial dimensions, and the logarithm compares different power levels. The local term compares adjacent-grid changes:

$$\ell_{\text{diff}}(\hat{x}, x) = \sum_{j=1}^{2} \text{mean}\,|D_j \hat{x} - D_j x|. \tag{4}$$

D_j is a periodic forward difference along direction j, without division by grid spacing. Means include dynamic variables and the relevant spatial or frequency entries. These scores compare predictions with reference fields, not governing-equation residuals. Their combination at lead h is

$$\mathcal{L}_h = \ell_{\text{field}}(\hat{x}_h, x_h) + \lambda_s \ell_{\text{spec}}(\hat{x}_h, x_h) + \lambda_d \ell_{\text{diff}}(\hat{x}_h, x_h). \tag{5}$$

We use $\epsilon = 10^{-8}$ and $\lambda_s = \lambda_d = 0.01$ for both paths. These scores are used to construct the forecast objectives below.

Direct-query objectives. For a window of H steps, sample $U \sim \text{Uniform}\{1, \ldots, H\}$. The paired leads $U, H + 1 - U$ supply the primary loss, and the endpoint H supplies the auxiliary loss. All are queried from x_0:

$$L_p = \tfrac{1}{2}\left(\mathcal{L}_U + \mathcal{L}_{H+1-U}\right), \qquad L_a = \mathcal{L}_H. \tag{6}$$

Both sampled leads are marginally uniform, giving $\mathbb{E}_U[L_p] = H^{-1} \sum_{h=1}^{H} \mathcal{L}_h$ before gradient processing. The pair covers the window in expectation, while each update also supervises the endpoint.

Recurrent objectives. All K recurrent predictions contribute to the primary loss. Later field errors receive larger weights, while structure scores are averaged uniformly:

$$L_p = \sum_{k=1}^{K} \frac{2k}{K(K+1)} \ell_{\text{field}}(\hat{x}_k^R, x_k) + \frac{1}{K} \sum_{k=1}^{K} \left[\lambda_s \ell_{\text{spec}}(\hat{x}_k^R, x_k) + \lambda_d \ell_{\text{diff}}(\hat{x}_k^R, x_k) \right]. \tag{7}$$

The field weights sum to one and affect training, not the evaluation average. Auxiliary queries reach the same times directly from x_0:

$$L_a = \frac{1}{K} \sum_{k=1}^{K} \ell_{\text{field}}(\hat{x}_k^D, x_k), \qquad \hat{x}_k^D = F_\theta(x_0, k\Delta t). \tag{8}$$

This field-only auxiliary branch avoids intermediate predicted inputs while training the same parameters. L_p, L_a retain their roles across formulations, but their construction differs. Direct training uses $H = 11$. Recurrent training uses the curriculum $K \in \{1, 2, 4, 8, 16, 20\}$, detaching the carried state every four steps while prediction continues.

3.3 Primary-guided Gradient Composition

The two losses may produce gradients of different magnitudes and directions. TEMPO uses the primary gradient to limit auxiliary dominance and remove opposing components, building on gradient-conflict handling (Yu et al., 2020; Liu et al., 2021).

Let $g_p = \nabla_\theta L_p$ and $g_a = \nabla_\theta L_a$. For a norm bound c, define $\text{Clip}_c(v) = v \min(1, c/\|v\|_2)$ when $v \neq 0$ and $\text{Clip}_c(0) = 0$. Clipping first sets the scale of the primary gradient:

$$\tilde{g}_p = \text{Clip}_G(g_p). \tag{9}$$

A projection unit b is the group of parameters processed together. We remove the opposing component of $g_{a,b}$ only when $\langle g_{a,b}, \tilde{g}_{p,b} \rangle < 0$:

$$a_b^\perp = \begin{cases} g_{a,b} - \dfrac{\langle g_{a,b}, \tilde{g}_{p,b} \rangle}{\|\tilde{g}_{p,b}\|_2^2} \tilde{g}_{p,b}, & \langle g_{a,b}, \tilde{g}_{p,b} \rangle < 0, \\ g_{a,b}, & \text{otherwise.} \end{cases} \tag{10}$$

In Figure 2, the obtuse angle between g_a and $\tilde{g}_p$ marks conflict, while the right angle marks its removal. Direct training projects the full gradient vector as one unit. Recurrent training projects each parameter tensor's gradient separately.

We join the projected units into $a^\perp$ and limit their total norm:

$$\tilde{g}_a = \text{Clip}_{\kappa \|\tilde{g}_p\|_2}(a^\perp), \qquad g = \tilde{g}_p + \lambda_a \tilde{g}_a. \tag{11}$$

Projection and norm control address different effects of auxiliary supervision. A small auxiliary gradient can oppose the primary direction, while a large nonconflicting gradient can dominate the combination. Equation (10) removes only opposing components. Equation (11) then limits the remaining auxiliary norm relative to the clipped primary gradient. An auxiliary gradient can therefore be rescaled even when no projection is needed. Because the bound depends on $\|\tilde{g}_p\|_2$, it is recomputed at each update rather than fixed independently of the primary gradient. Both objectives still update the same parameters, instead of training separate predictors. The coefficient λ_a sets the contribution of this bounded auxiliary signal. The forecast targets remain unchanged.

We use $G = \kappa = 1$, with $\lambda_a = 1$ for direct training and 0.25 for recurrent training. AdamW (Loshchilov & Hutter, 2019) receives g without further clipping. The equations describe nondegenerate cases. In implementation, squared norms are floored at 10^{-12}, and parameters with no available primary gradient receive no auxiliary contribution.

Algorithm 1 follows the same order. **Step 1** forms the forecasts and their losses. Direct training uses two sampled primary queries and one endpoint query. Recurrent training uses its predicted sequence for primary supervision and direct queries from the same x_0 for auxiliary supervision. Both losses are minibatch averages.

Step 2 computes a separate gradient for each loss. The primary gradient is clipped first. Opposing auxiliary components are removed before the result is capped across all parameter tensors. There is no second clipping operation after addition. By contrast, the ordinary-sum approach clips the gradient of $L_p + \lambda_a L_a$ as a whole. The comparison concerns both composition and clipping, not projection alone.

Step 3 supplies the resulting gradient to AdamW and advances the learning-rate schedule. The train-tail validation rule retains one checkpoint for both forecast metrics. Processing the gradients does not guarantee that every AdamW update improves validation error. Optimizer and validation details are in Appendix B.

Algorithm 1 TEMPO-PDE post-training

Require: Data, weights θ_0, fixed formulation
Require: Schedule, validation rule, G, κ, λ_a
Initialize $\theta \leftarrow \theta_0$ and AdamW
for each training update **do**
  Sample starting states and references
  Step 1: Form the objectives
  if direct **then**
    Sample U uniformly in $1, \ldots, H$
    Query $U, H+1-U, H$ from x_0
    Form L_p, L_a by Eq. (6)
  else
    Set K from the curriculum
    Roll out K steps, detaching every four
    Query $1, \ldots, K$ directly from x_0
    Form L_p, L_a by Eqs. (7)–(8)
  end if
  Step 2: Process gradients
  $g_p \leftarrow \nabla_\theta L_p$, $g_a \leftarrow \nabla_\theta L_a$
  $\tilde{g}_p \leftarrow \mathrm{Clip}_G(g_p)$
  Project each unit by Eq. (10)
  Join units and cap by Eq. (11)
  Step 3: Update and select
  Set $g \leftarrow \tilde{g}_p + \lambda_a \tilde{g}_a$
  AdamW step with g, then schedule step
  Save the best F_θ at scheduled validation points
end for
return the saved F_θ

Deployment. The saved predictor in Algorithm 1 loads the adapted weights θ shown in Figure 2. Direct models query times within a block and then start the next block from the predicted endpoint. Recurrent models continue from their preceding predictions. Inference runs only F_θ, without scoring or gradient processing.

4 Experiments

We ask three questions about TEMPO-PDE. **RQ1:** How does TEMPO compare with existing predictors and strong same-backbone controls? **RQ2:** Which training components improve the predictions, and which spatial errors do they affect? **RQ3:** Do forecast gains persist over longer horizons, and does the per-step score remain useful on another architecture?

4.1 Experimental Settings

Tasks and evaluation. The main comparison uses five Poseidon tasks (Herde et al., 2024): NS-PwC, NS-SL, NS-SVS, FNS-KF, and CE-RPUI. NS-PwC, FNS-KF, and CE-RPUI use direct queries, while NS-SL and NS-SVS use recurrent prediction. For Poseidon, the last observed state is stored frame 9 under zero-based indexing. We denote it by x_0 and predict frames 10–20. At each step, relative-L_2 errors are averaged across dynamic variables. Mean averages steps uniformly, while Final uses the last step. Both metrics use the same checkpoint. The registered main comparison averages two training seeds on the common 128-trajectory development set. Three-seed follow-up studies retain their separate identities in Appendix C.3.

Training and comparisons. Matched Poseidon-B runs use 1,280 updates with 84 target trajectories for optimization and 16 for validation. Table 1 separates 13 external architectures from completed Poseidon-B controls. Table 2 evaluates NS-PwC with both prediction paths, giving six task–formulation combinations across the five benchmarks. Five variants and three seeds per combination yield 90 models, evaluated on Gate96, a separate 96-trajectory development set. Training settings, comparator configurations, and complete ablation results are detailed in Appendices B, C, and D.

Table 1: **Forecast errors on five main PDE benchmarks.** Mean relative-L_2 error in units of 10^{-3}, lower is better. **Bold** marks the best entry shown in each column. The original 58-configuration comparison and follow-up controls are reported in Appendix C.

Method	NS-PwC	NS-SL	NS-SVS	FNS-KF	CE-RPUI
Standardized external architectures					
FNO (Li et al., 2021a)	68.39	81.45	60.03	73.95	293.6
F-FNO (Tran et al., 2023)	94.46	119.5	100.2	98.76	358.2
U-NO (Rahman et al., 2023)	225.4	154.3	76.31	243.1	676.6
LSM (Wu et al., 2023)	261.5	139.0	51.00	255.0	634.7
FactFormer (Li et al., 2023a)	479.2	416.5	312.8	478.4	597.0
Transolver (Wu et al., 2024)	475.0	436.2	206.5	459.2	611.4
UPT (Alkin et al., 2024)	486.6	448.7	558.9	491.1	634.1
DRIFT-Net (Li & Salim, 2025)	564.1	517.2	460.1	577.2	658.2
SGNO (Li et al., 2026)	200.9	255.0	218.2	220.9	492.0
Flower-Tiny (Muser et al., 2026)	37.23	62.13	39.22	54.12	273.7
DPOT-M (Hao et al., 2024)	49.76	48.40	30.64	52.36	273.0
CNO-FM (Herde et al., 2024; Raonić et al., 2023)	154.4	142.0	72.23	149.3	547.5
PDE-Transformer (Holzschuh et al., 2025)	189.1	247.2	260.0	208.2	459.6
Registered same-backbone comparison					
Best registered Poseidon-B control[†]	9.075	17.72	20.47	9.748	126.5
TEMPO-PDE (Ours)	**8.974**	**17.19**	**19.77**	**9.507**	**124.1**

[†]Per-task minimum within the original registered Poseidon-B control pool, not a single training recipe or a deployable selection rule. Input histories and training budgets differ across configurations. Follow-up controls are reported separately in Appendix C.3.

4.2 Forecast Accuracy Across Tasks (RQ1)

Table 1 compares TEMPO with external architectures and strong adaptations of the same Poseidon-B backbone. The former provide architectural context, while the latter test gains beyond same-backbone training. Appendix C reports Final error and all evaluated configurations.

Obs. ❶ TEMPO leads the registered comparison on the five main tasks. TEMPO achieves the lowest Mean and Final errors on NS-PwC, NS-SL, NS-SVS, FNS-KF, and CE-RPUI. Follow-up controls for supervision, auxiliary weighting, and training budget are analyzed separately in Appendix C.3. Their rankings vary with the task and training configuration.

Obs. ❷ Gains persist relative to strong registered same-backbone controls. Across the five main tasks, Mean error decreases by 1.1–3.4% and Final error by 1.6–3.8% relative to the strongest registered comparator. These margins quantify the remaining gains over competitive adaptations of Poseidon-B. The larger differences from external architectures also reflect differences in model representations, input histories, and training configurations, and are not attributed to TEMPO alone.

4.3 Functional Contributions and Spatial Errors (RQ2)

Table 2 compares each intervention with Full TEMPO within the same task and prediction path. Primary and auxiliary specify which forecasts supply supervision. Field and structure specify how those forecasts are scored. Field-only primary removes auxiliary supervision and both structure terms, but retains the primary time targets and field score. Each column is a matched comparison. Differences across columns do not isolate the prediction path.

Obs. ❸ Complementary supervision improves the field-only baseline. Jointly removing auxiliary and structure supervision raises Mean by 8.9–15.0% in five of the six combinations, with higher Final error in each. Removing only auxiliary supervision raises Mean by 5.4–8.9% in these same combinations. CE-RPUI differs: the joint deletion lowers Mean by 1.54%, while its Final change

Table 2: **Functional ablations of TEMPO.** 11 forecast steps and 90 trained models. Full reports Mean error ($\times 10^{-3}$) with sample SD. Other rows report mean within-seed changes from Full (%), where positive is worse.

	Direct query			Recurrent rollout		
Configuration	NS-PwC	CE-RPUI	FNS-KF	NS-PwC	NS-SL	NS-SVS
Full TEMPO	9.26 ± 0.09	125.90 ± 0.32	9.55 ± 0.09	16.25 ± 0.09	16.56 ± 0.31	19.14 ± 0.19
w/o auxiliary	+8.77	+0.25	+5.43	+8.85	+6.73	+6.88
w/o structure	+3.46	−1.57	+3.07	+0.30	+3.51	+4.24
Ordinary sum	−0.27	+1.08	−0.40	+5.05	+2.88	+6.40
Field-only primary	+14.98	−1.54	+10.56	+11.86	+9.23	+8.87

Field-only primary removes both auxiliary and structure supervision, retaining the primary time targets. SD describes three training seeds, not a confidence interval. Full results are in Appendix D.

Table 3: **Score transfer to Flower-Tiny.** Native and TEMPO-score training use the same architecture. Errors are in $\times 10^{-3}$, with lower values better. Only the per-step score is transferred.

(a) Short forecasts on five tasks

	Mean error		Final error	
Task	Native	TEMPO	Native	TEMPO
NS-PwC	15.335	**10.933**	25.849	**18.258**
NS-SL	23.968	**23.159**	**41.428**	42.045
NS-SVS	23.702	**23.343**	**43.215**	44.029
FNS-KF	23.695	**13.809**	41.385	**23.886**
CE-RPUI†	191.725	**171.296**	289.684	**262.975**

(b) NS-PwC at 119 steps

Metric	Native	TEMPO
Mean	272.34	**200.93**
Final	660.63	**519.50**
Retained (%)	85.94	**92.71**

Retained: correlation ≥ 0.8 at all 119 steps. Higher is better. Matched learning rate.

(a) Three seeds on development Gate96, 11 steps. †CE-RPUI also changes the learning rate from 0.001 to 0.0003. (b) Two separate seeds on the repaired development cohort. See Appendix E.

of +0.13% has a paired interval spanning zero. The joint and individual comparisons therefore complement each other without implying a uniform benefit on every task.

Obs. ❹ The update rule matters when the forecast objectives are retained. The ordinary sum retains both objectives but replaces their gradient composition and clipping procedure. Mean error rises by 5.0%, 2.9%, and 6.4% in the recurrent NS-PwC, NS-SL, and NS-SVS configurations, respectively, and by 1.1% in direct-query CE-RPUI. Direct-query NS-PwC and FNS-KF instead show small average advantages for ordinary sum. These results support a setting-dependent contribution from primary-guided composition, with more consistent gains in recurrent training.

Obs. ❺ Structure scoring and gradient composition affect different spatial scales. Figure 3 groups residual power into radial wavenumber shells, with larger k representing finer spatial variation. Over $k = 33, \ldots, 64$, removing structure supervision raises residual power by 59% on direct NS-PwC and 48% on recurrent NS-SL, but lowers it by 11% on direct CE-RPUI. Ordinary sum remains close to Full in this band but has 7.0% higher residual power over $k = 1, \ldots, 8$ on NS-SL, indicating a larger change at coarser scales. These controls distinguish the spatial effects of scoring and update composition.

4.4 Longer Forecasts and Score Transfer (RQ3)

We next examine whether TEMPO's choices remain useful over longer forecasts and on other architectures. Figure 1 tests prediction beyond the standard window, while Table 3 evaluates score transfer to Flower-Tiny. Matched DPOT-L post-training comparisons are reported in Appendix E.5.

Obs. ❻ Short-window differences do not determine the longer forecast. Figure 1 evaluates fixed Poseidon-B checkpoints on NS-PwC for 247 steps. Direct prediction advances in 11-step

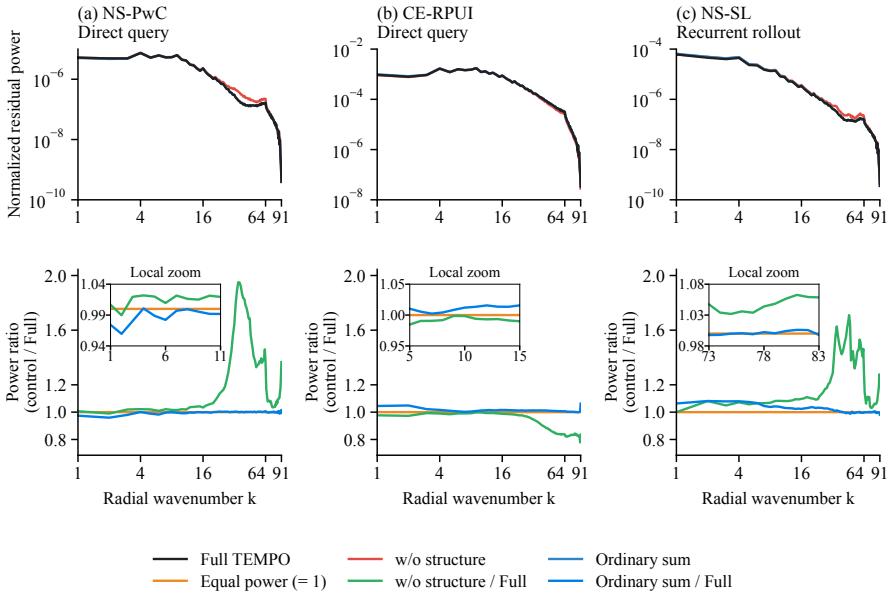


Figure 3: **Spatial effects of structure scoring and gradient composition.** Each column compares Full TEMPO, w/o structure, and ordinary sum on the same task and prediction path. **Top**: normalized residual power. **Bottom**: each control's residual power divided by Full. Values above 1 mean higher residual power than Full. Curves average three training seeds.

blocks, using each predicted endpoint as the next starting state. At the original learning rate and 1,280 updates, ordinary sum is within 0.1% of matched TEMPO over 11 steps but 7.5% higher over 247 steps; on 70 shared trajectories, a second lead-sampling realization gives 0.98% instead of 7.41% (Appendix E). At 2,944 updates, the sum is 9.4% lower over 11 steps yet 8.0% higher over 247. These checkpoints and their source-disjoint cohort differ from those in Table 1.

Obs. ❼ The per-step score transfers to another architecture. Table 3a trains single-state Flower-Tiny from random initialization with its native loss or the TEMPO score, without auxiliary supervision or gradient composition. Mean error decreases on all five main tasks and Final error on three; the CE-RPUI score run also uses a lower learning rate. To examine longer predictions, a separate two-seed NS-PwC study uses a matched learning rate (Table 3b). Over 119 steps, Mean and Final decrease by 26.2% and 21.4%, respectively. The improvement also extends to spatial-pattern retention: the share of forecasts whose correlation stays at or above 0.8 for all 119 steps rises from 85.94% to 92.71%, a gain of 6.77 percentage points.

5 Conclusion

We presented TEMPO-PDE, a post-training framework that uses reference trajectories to improve forecasts from pretrained PDE models. Its central idea is to design trajectory supervision and the resulting parameter updates together. Field, spectral, and spatial-difference scores capture complementary prediction errors, while primary-guided composition controls how auxiliary supervision contributes to learning. These operations adapt the existing predictor without adding an inference module. In the registered benchmark comparison, TEMPO achieves the lowest Mean and Final errors on the five main tasks. Longer forecasts show why early accuracy alone is insufficient to assess adaptation, and the Flowers experiments extend the evidence for forecast scoring to another architecture. Together, these results support using both temporal and spatial information in reference trajectories to guide post-training, with full forecasts providing the relevant measure of its effectiveness in predicting both field evolution and structure.

AI USE STATEMENT

We used generative AI assistants for feedback on methodology and control experiments, result interpretation, drafting and revising training, evaluation, and plotting code. We did not use generative AI tools to generate synthetic data, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, translate text, or clean or reformat data sets. Qualitative and thematic data analysis is not applicable to this work. We have reviewed all AI-assisted work: the authors ran the experiments, inspected and executed all code, retained the experimental records, and checked every reported number and reference against its source. We take responsibility for the final text, claims, code, and results, including content developed with AI assistance.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *arXiv preprint arXiv:2402.12365*, 2024. URL https://arxiv.org/abs/2402.12365.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.03376.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the International Conference on Machine Learning*, volume 80, pp. 794–803, 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

Wojciech M. Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/758a06618c69880a6cee5314ee42d52f-Abstract.html.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 17616–17635, 2024. URL https://proceedings.mlr.press/v235/hao24d.html.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/84e1b1ec17bb11c57234e96433022a9a-Abstract-Conference.html.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-Transformer: Efficient and versatile transformers for physics simulations. In *Proceedings of the International Conference on Machine Learning*, volume 267, pp. 23562–23602, 2025. URL https://proceedings.mlr.press/v267/holzschuh25a.html.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. URL https://doi.org/10.1073/pnas.2101784118.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, et al. Neural general circulation models for weather and climate. *Nature*, 632:1060–1066, 2024. URL https://doi.org/10.1038/s41586-024-07744-y.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces

with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023. URL https://doi.org/10.1126/science.adi2336.

Jiayi Li and Flora D. Salim. DRIFT-Net: A spectral-coupled neural operator for PDEs learning. *arXiv preprint arXiv:2509.24868*, 2025. URL https://arxiv.org/abs/2509.24868.

Jiayi Li, Zhaonan Wang, and Flora D. Salim. SGNO: Spectral generator neural operators for stable long horizon PDE rollouts. *arXiv preprint arXiv:2602.18801v1*, 2026. URL https://arxiv.org/abs/2602.18801v1.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for PDE surrogate modeling. *arXiv preprint arXiv:2305.17560*, 2023a. URL https://arxiv.org/abs/2305.17560.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021b. URL https://arxiv.org/abs/2111.03794.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/70518ea42831f02afc3a2828993935ad-Abstract-Conference.html.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2308.05732.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal PDE foundation model for learning multiple operators for forecasting fluid dynamics. *arXiv preprint arXiv:2409.09811*, 2024. URL https://arxiv.org/abs/2409.09811.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021. URL https://doi.org/10.1038/s42256-021-00302-5.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *Advances in Neural Information Processing Systems*, volume 37,

2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d7cb9db5ade2db7814fbd01ee59f4c7b-Abstract-Conference.html.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Régaldo-Saint Blancard, Francois Rozet, et al. Walrus: A cross-domain foundation model for continuum dynamics. *arXiv preprint arXiv:2511.15684*, 2025. URL https://arxiv.org/abs/2511.15684.

Till Muser, Alexandra Spitzer, Matti Lassas, Maarten V. de Hoop, and Ivan Dokmanić. Flowers: A warp drive for neural PDE solvers. *arXiv preprint arXiv:2603.04430*, 2026. URL https://arxiv.org/abs/2603.04430.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2204.11127.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. URL https://doi.org/10.1016/j.jcp.2018.10.045.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper/2023/hash/f3c1951b34f7f55ffaecada7fde6bd5a-Abstract-Conference.html.

Louis Serrano, Armand Kassaï Koupaï, Thomas X. Wang, Pierre Erbacher, and Patrick Gallinari. Zebra: In-context generative pretraining for solving parametric PDEs. In *Proceedings of the International Conference on Machine Learning*, volume 267, pp. 53940–53988, 2025. URL https://proceedings.mlr.press/v267/serrano25a.html.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized Fourier neural operators. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2111.13802.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021. doi: 10.1137/20M1318043.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional PDEs with latent spectral models. In *Proceedings of the International Conference on Machine Learning*, volume 202, pp. 37417–37438, 2023. URL https://proceedings.mlr.press/v202/wu23f.html.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 53681–53705, 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Congcong Zhu, Xiaoyan Xu, Jiayue Han, and Jingrun Chen. Physics-informed temporal alignment for auto-regressive PDE foundation models. In *Proceedings of the International Conference on Machine Learning*, volume 267, pp. 80223–80258, 2025. URL https://proceedings.mlr.press/v267/zhu25w.html.

A Evaluation Scope and Statistical Interpretation

Reporting scope. The displayed results cover NS-PwC, NS-SL, NS-SVS, FNS-KF, and CE-RPUI. The reported TEMPO entry is a fine-tuning fallback with Fresh128 Mean and Final errors of 24.04 and 42.44 ($\times 10^{-3}$), whereas other recorded configurations reach 0.976 and 1.348; the Flower-Tiny score transfer raises its Gate96 Mean from 3.977 to 4.013 and Final from 4.998 to 5.351.

Evaluation sets. The common comparison uses 128 trajectories per task. Appendix C displays the five-task subset of the original 58-configuration registry. This set, denoted Fresh128 in the experiment records, has been examined during development. Its ten leading task–metric cells are descriptive results on that common evaluation set. A separate one-shot test set, FinalTest144, on which TEMPO leads in eight of twelve task–metric cells, is not replaced by the Fresh128 comparison. Development Gate96 is used for the functional ablations and the Flowers score-transfer study. Long-forecast cohorts and their source-identity repairs are described in Appendix E. These sets are not interchangeable, even when they use the same prediction length.

Forecast metrics. For seed s, trajectory i, and forecast step h, let $e_{s,i,h}$ be the spatial relative-L_2 error, averaged equally across dynamic variables. For an evaluation window of L steps, the two trajectory-level metrics are

$$E_{s,i}^{\mathrm{Mean}} = \frac{1}{L}\sum_{h=1}^{L} e_{s,i,h}, \qquad E_{s,i}^{\mathrm{Final}} = e_{s,i,L}.$$

Mean uses a uniform time average, independent of the weights in the recurrent training loss. Both metrics are evaluated from the same selected checkpoint. Main results average two fitted training seeds. Functional results use three seeds, with the statistical treatment specified in Appendix D.

Main-comparison statistics. The analysis first averages the two fitted seeds within each trajectory and then uses 50,000 paired trajectory resamples. The strongest comparator is reselected within each resample. Its intersection–union test uses centered, studentized Rademacher wild resampling and Holm correction across the original twelve task–metric hypotheses. The correction family is not reduced by restricting the displayed tasks. These procedures condition on the fitted models and do not remove dependence on the development process. They are distinct from the unadjusted functional-ablation intervals.

Comparator identity. The strongest Poseidon-B reference in the main table is a descriptive per-task minimum within the original registered Poseidon-based control pool. It is not the unadapted pretrained model, one fixed training recipe, or a rule for selecting a model before evaluation. The contributing controls can differ in prediction path and training configuration. Configuration identities for the displayed tasks are retained in Appendix C.

B Training and Deployment Protocols

B.1 Data and optimization

A task uses a separately adapted copy of Poseidon-B. Its 100 registered target trajectories are partitioned into 84 for optimization and 16 for train-tail validation. The standard schedule contains 20 epochs of 64 updates each. Training uses batch size four, input-noise scale 0.01, bfloat16 forward computation, non-reentrant gradient checkpointing, and FP32 metric accumulation. All model parameters are fine-tuned.

AdamW uses its recorded parameter groups, with initial learning rates of 0.00005 for standard and no-decay groups and 0.0005 for embedding/recovery and time-embedding groups. Weight decay is 0.000001 except in the no-decay and time-embedding groups, where it is zero. The recorded optimizer has betas $(0.9, 0.999)$ and epsilon 10^{-8}. Cosine learning-rate decay advances after each optimizer step. An exponential moving average with decay 0.999 is saved, but the reported evaluations use selected raw model weights.

B.2 Forecast objectives and gradient processing

Direct training samples U uniformly from the eleven future leads and pairs it with $12 - U$. The two composite scores form the primary objective, and lead 11 provides an endpoint auxiliary objective. The spectral and spatial-difference weights are both 0.01, including in the endpoint score. The auxiliary coefficient is one. Known task-specific inputs, including forcing in FNS-KF, remain part of the corresponding data and model interface.

Recurrent training follows a curriculum with lengths in $\{1, 2, 4, 8, 16, 20\}$ and truncates cross-step gradients every four steps. Its primary field errors are weighted by $2k/[K(K+1)]$. Spectral and spatial-difference terms are uniformly averaged across the recurrent sequence, with weights 0.01. Direct queries from the same observed starting state provide an auxiliary field-only objective with coefficient 0.25. That auxiliary objective is also averaged uniformly over its queried leads.

Full TEMPO clips the primary gradient to global norm one. It removes conflicting auxiliary components using whole-vector projection for direct training and per-parameter-tensor projection for recurrent training. A global auxiliary norm cap of one times the clipped primary norm is then applied before weighted addition. No additional clipping is applied to the combined gradient. The nondegenerate formulas in Section 3 express this order. The implementation also guards small denominators and handles parameters without an available primary gradient. The gradient-space construction does not establish a monotonicity guarantee for the AdamW parameter step or validation error.

B.3 Checkpoint selection and deployment

Validation is performed on the 16 train-tail trajectories every four epochs. Direct-query configurations select the checkpoint by the equally weighted validation Mean and Final score. Recurrent configurations use validation Mean. One selected raw checkpoint serves both evaluation metrics. Configurations were selected by normalized minimax over predeclared candidates. The CE-RPUI configuration rerun with all four channels is an explicit exception to this rule.

The standard evaluation starts from observed frame 9 and predicts frames 10 through 20 for the five displayed tasks. Direct evaluation queries the requested times from the same observed state within each block. Long direct forecasts use the predicted block endpoint as the next starting state. Recurrent evaluation advances serially through predicted states. The model's normalization and decoding remain part of its prediction interface. Training losses and gradient processing are not executed at inference.

C Five-Task Comparison Grid and Follow-Up Controls

Input-history interfaces. The registered NS-PwC deployment audit uses one state for Poseidon-B and the standardized FNO, F-FNO, U-NO, LSM, FactFormer, Transolver, UPT, DRIFT-Net, SGNO, Flower-Tiny, CNO-FM, and PDE-Transformer implementations. DPOT-M uses a ten-state history. The separate Flowers author-interface configurations use four states, while the DPOT-L transfer study uses ten. These counts describe the evaluated implementations rather than mandatory input lengths for every version of an architecture.

The grid lists the 54 configuration groups with at least one result on the five displayed tasks, with their original identifiers. Input-history variants, objective-transfer variants, and configurations with incomplete task coverage are retained whenever they have a result on a displayed task. A missing value indicates that no corresponding common-evaluation result is available, not zero error. Errors are in $\times 10^{-3}$. Highlighting for the displayed columns is retained from the original unrounded comparison pool. It does not rank the separate follow-up studies. The standardized external configurations are the implementations evaluated in this study, not claims to reproduce every original paper's numerical results under its original protocol.

The main table's best-control row is an explicitly labeled descriptive reference. It is not a single deployed method assembled by selecting a different checkpoint for every metric. In contrast, every reported TEMPO configuration uses one checkpoint for both Mean and Final.

C.1 WINDOW-AVERAGED ERROR

Table 4: **Five-task comparison grid: Mean error** ($\times 10^{-3}$, Fresh128). Bold and underline mark the lowest and second-lowest unrounded values in each column.

Configuration	NS-PwC	NS-SL	NS-SVS	FNS-KF	CE-RPUI
FNO [r00]	68.39	81.45	60.03	73.95	293.6
F-FNO [r01]	94.46	119.5	100.2	98.76	358.2
U-NO [r02]	225.4	154.3	76.31	243.1	676.6
LSM [r03]	261.5	139.0	51.00	255.0	634.7
FactFormer [r04]	479.2	416.5	312.8	478.4	597.0
Transolver [r05]	475.0	436.2	206.5	459.2	611.4
UPT [r06]	486.6	448.7	558.9	491.1	634.1
DRIFT-Net [r07]	564.1	517.2	460.1	577.2	658.2
SGNO [r08]	200.9	255.0	218.2	220.9	492.0
Flower-Tiny [r09]	37.23	62.13	39.22	54.12	273.7
DPOT-M [r10]	49.76	48.40	30.64	52.36	273.0
CNO-FM [r11]	154.4	142.0	72.23	149.3	547.5
PDE-Transformer [r12]	189.1	247.2	260.0	208.2	459.6
Poseidon-B baseline portfolio [r13]	9.226	18.30	22.01	<u>9.748</u>	<u>126.5</u>
Flowers, 4-frame native [r14]	13.57	—	—	—	—
Flowers, 4-frame TEMPO target [r15]	11.68	—	—	—	—
Flowers, 4-frame MSE-H4 [r16]	15.15	—	—	—	—
DPOT-M, 10-frame, LR 3e-4 [r17]	23.74	—	—	—	—
DPOT-M, 10-frame, LR 1e-3 [r18]	23.46	—	—	—	—
DPOT-M, 10-frame, LR 3e-3 [r19]	144.3	—	—	—	—
Direct FT [r20]	13.97	29.48	29.99	17.21	—
Recurrent FT [r21]	17.29	—	—	20.99	147.0
CNO-FM, native direct [r22]	66.40	—	—	—	—
CNO-FM, native serial [r23]	290.5	—	—	—	—
CNO-FM, native span2 [r24]	75.21	—	—	—	—
PDE-Transformer, native EMA [r25]	53.40	—	—	—	—
PDE-Transformer, native raw [r26]	52.57	—	—	—	—
Flower-Tiny, U2560 [r27]	25.13	39.81	29.51	41.11	223.4
CNO-FM, LR $\times 0.3$ direct [r28]	69.68	—	—	—	—
CNO-FM, LR $\times 0.3$ serial [r29]	249.9	—	—	—	—
CNO-FM, LR $\times 0.3$ span2 [r30]	81.09	—	—	—	—
CNO-FM, LR $\times 3$ direct [r31]	66.36	—	—	—	—
CNO-FM, LR $\times 3$ serial [r32]	268.2	—	—	—	—
CNO-FM, LR $\times 3$ span2 [r33]	71.14	—	—	—	—
PDE-Transformer, LR 1e-4 EMA [r34]	61.67	—	—	—	—
PDE-Transformer, LR 1e-4 raw [r35]	59.56	—	—	—	—
Matched direct FT, CE [r36]	—	—	—	—	129.9
Flowers native, LR 3e-4 U2560 [r37]	35.41	43.84	32.38	60.90	235.1
Per-tensor temporal control [r38]	<u>9.075</u>	—	—	9.833	—
PITA data-only, LR 3e-4 [r39]	108.9	—	—	—	—
PITA data-only, LR 1e-3 [r40]	66.78	—	—	—	—
PITA data-only, LR 3e-3 [r41]	67.05	—	—	—	—
PITA literal residual, LR 3e-4 [r42]	420.7	—	—	—	—
PITA literal residual, LR 1e-3 [r43]	634.0	—	—	—	—
PITA literal residual, LR 3e-3 [r44]	958.9	—	—	—	—
PITA modified grid, LR 3e-4 [r48]	342.3	—	—	—	—
PITA modified grid, LR 1e-3 [r49]	610.0	—	—	—	—
PITA modified grid, LR 3e-3 [r50]	1027	—	—	—	—
Poseidon-B repaired reference [r51]	—	<u>17.72</u>	<u>20.47</u>	—	—
Two-channel CE variant (legacy input) [r53]	—	—	—	—	128.3
Flowers, Exact-H4 transfer [r54]	18.91	35.16	28.02	27.36	201.1
Flowers, Pareto transfer [r55]	23.60	35.43	28.43	33.15	232.7
DPOT-L, TEMPO transfer [r56]	42.15	45.09	28.60	44.35	272.6
TEMPO-PDE (Poseidon-B) [r57]	**8.974**	**17.19**	**19.77**	**9.507**	**124.1**

C.2 FINAL-STEP ERROR

Table 5: **Five-task comparison grid: Final error.** Units and highlighting as in Table 4.

Configuration	NS-PwC	NS-SL	NS-SVS	FNS-KF	CE-RPUI
FNO [r00]	111.1	141.0	109.3	120.9	402.3
F-FNO [r01]	155.1	193.7	172.3	162.6	478.4
U-NO [r02]	363.3	240.9	131.7	386.6	801.1
LSM [r03]	419.5	229.3	90.94	407.2	769.6
FactFormer [r04]	703.5	625.9	468.1	694.1	650.2
Transolver [r05]	733.5	664.5	361.0	729.6	663.3
UPT [r06]	717.9	665.6	748.2	717.6	678.9
DRIFT-Net [r07]	708.4	632.1	573.7	718.7	713.5
SGNO [r08]	325.5	414.0	374.6	351.6	606.4
Flower-Tiny [r09]	61.56	101.1	68.69	90.02	383.0
DPOT-M [r10]	76.23	76.01	49.22	81.25	358.1
CNO-FM [r11]	249.3	226.9	125.4	239.4	703.1
PDE-Transformer [r12]	316.7	395.0	422.0	354.7	612.0
Poseidon-B baseline portfolio [r13]	12.41	30.01	37.45	<u>13.82</u>	<u>184.3</u>
Flowers, 4-frame native [r14]	26.98	—	—	—	—
Flowers, 4-frame TEMPO target [r15]	19.77	—	—	—	—
Flowers, 4-frame MSE-H4 [r16]	23.90	—	—	—	—
DPOT-M, 10-frame, LR 3e-4 [r17]	37.84	—	—	—	—
DPOT-M, 10-frame, LR 1e-3 [r18]	38.46	—	—	—	—
DPOT-M, 10-frame, LR 3e-3 [r19]	225.7	—	—	—	—
Direct FT [r20]	22.62	60.58	67.03	27.91	—
Recurrent FT [r21]	27.35	—	—	34.05	224.2
CNO-FM, native direct [r22]	112.4	—	—	—	—
CNO-FM, native serial [r23]	522.4	—	—	—	—
CNO-FM, native span2 [r24]	115.7	—	—	—	—
PDE-Transformer, native EMA [r25]	92.62	—	—	—	—
PDE-Transformer, native raw [r26]	91.20	—	—	—	—
Flower-Tiny, U2560 [r27]	41.05	66.88	51.13	69.08	328.1
CNO-FM, LR $\times 0.3$ direct [r28]	115.6	—	—	—	—
CNO-FM, LR $\times 0.3$ serial [r29]	440.2	—	—	—	—
CNO-FM, LR $\times 0.3$ span2 [r30]	124.3	—	—	—	—
CNO-FM, LR $\times 3$ direct [r31]	114.1	—	—	—	—
CNO-FM, LR $\times 3$ serial [r32]	487.8	—	—	—	—
CNO-FM, LR $\times 3$ span2 [r33]	109.9	—	—	—	—
PDE-Transformer, LR 1e-4 EMA [r34]	107.4	—	—	—	—
PDE-Transformer, LR 1e-4 raw [r35]	103.6	—	—	—	—
Matched direct FT, CE [r36]	—	—	—	—	193.8
Flowers native, LR 3e-4 U2560 [r37]	57.55	72.10	55.42	100.5	344.4
Per-tensor temporal control [r38]	<u>12.38</u>	—	—	14.06	—
PITA data-only, LR 3e-4 [r39]	196.8	—	—	—	—
PITA data-only, LR 1e-3 [r40]	116.8	—	—	—	—
PITA data-only, LR 3e-3 [r41]	116.7	—	—	—	—
PITA literal residual, LR 3e-4 [r42]	544.2	—	—	—	—
PITA literal residual, LR 1e-3 [r43]	709.7	—	—	—	—
PITA literal residual, LR 3e-3 [r44]	1048	—	—	—	—
PITA modified grid, LR 3e-4 [r48]	532.4	—	—	—	—
PITA modified grid, LR 1e-3 [r49]	839.0	—	—	—	—
PITA modified grid, LR 3e-3 [r50]	1027	—	—	—	—
Poseidon-B repaired reference [r51]	—	<u>28.68</u>	<u>35.10</u>	—	—
Two-channel CE variant (legacy input) [r53]	—	—	—	—	191.5
Flowers, Exact-H4 transfer [r54]	32.45	62.28	51.78	48.75	297.7
Flowers, Pareto transfer [r55]	40.01	61.59	51.58	57.06	334.7
DPOT-L, TEMPO transfer [r56]	64.28	68.69	46.72	68.09	349.3
TEMPO-PDE (Poseidon-B) [r57]	**12.11**	**28.03**	**33.77**	**13.48**	**181.3**

C.3 FOLLOW-UP POST-TRAINING CONTROLS

The following studies extend the original registered comparison without replacing its trained checkpoints. They are reported separately because they answer controlled questions about supervision, update composition, and budget. The Fresh128 study evaluates three fitted seeds per configuration on the same 128-trajectory development set used for the original grid. The static-weight study instead uses development Gate96. Neither study is a new blind test. The five-variant matched study retains 1,280 updates and each context's prediction path and selection rule.

Matched configurations on the common evaluation set. Table 6 reports the complete five-variant, six-context matrix. Full improves both metrics over field-only primary in five contexts, but the relative ranking among all five variants is not uniform. These are different trained groups from the two-seed TEMPO entry in Table 1. No per-task choice between those groups is used to redefine the main-table result.

Table 6: Follow-up matched errors on Fresh128. Mean and Final errors are in $\times 10^{-3}$. Each entry is the mean and sample SD of three training-seed means, not a confidence interval.

Configuration	Mean error	Final error
NS-PwC / direct		
Full TEMPO	9.127 ± 0.090	12.362 ± 0.177
w/o auxiliary	9.927 ± 0.801	14.716 ± 1.439
w/o structure	9.453 ± 0.144	12.690 ± 0.246
Ordinary sum	9.120 ± 0.077	12.338 ± 0.101
Field-only primary	10.473 ± 1.043	15.469 ± 1.725
CE-RPUI / direct		
Full TEMPO	124.647 ± 0.173	181.995 ± 0.336
w/o auxiliary	124.345 ± 2.147	183.306 ± 2.641
w/o structure	122.331 ± 0.424	179.369 ± 0.321
Ordinary sum	125.919 ± 0.409	183.301 ± 0.334
Field-only primary	122.087 ± 0.479	180.652 ± 0.309
FNS-KF / direct		
Full TEMPO	9.707 ± 0.116	13.970 ± 0.257
w/o auxiliary	10.271 ± 0.051	16.331 ± 0.198
w/o structure	10.016 ± 0.294	14.246 ± 0.527
Ordinary sum	9.652 ± 0.216	13.689 ± 0.421
Field-only primary	10.810 ± 0.086	16.849 ± 0.383
NS-PwC / recurrent		
Full TEMPO	16.062 ± 0.071	25.334 ± 0.267
w/o auxiliary	17.403 ± 0.802	28.171 ± 1.484
w/o structure	16.007 ± 0.203	24.477 ± 0.572
Ordinary sum	16.913 ± 0.634	27.136 ± 1.494
Field-only primary	17.951 ± 0.623	28.309 ± 1.138
NS-SL / recurrent		
Full TEMPO	16.673 ± 0.193	27.095 ± 0.282
w/o auxiliary	17.677 ± 0.251	28.975 ± 0.407
w/o structure	17.188 ± 0.113	27.803 ± 0.345
Ordinary sum	17.087 ± 0.024	27.850 ± 0.082
Field-only primary	18.063 ± 0.198	29.396 ± 0.225
NS-SVS / recurrent		
Full TEMPO	18.937 ± 0.185	32.293 ± 0.456
w/o auxiliary	20.418 ± 0.368	35.370 ± 0.416
w/o structure	19.958 ± 0.585	34.088 ± 1.321
Ordinary sum	20.226 ± 0.631	34.963 ± 1.125
Field-only primary	20.661 ± 0.278	35.322 ± 0.481

Extended ordinary-sum training. The extended schedules retain the ordinary-sum objectives and use 2,944 updates on NS-PwC and 3,200 on CE-RPUI. They also change the cosine schedule and number of validation opportunities. They are not single-factor update-count interventions or exact measured compute matches. On Fresh128, extended sum is lower than the matched Full on both NS-PwC metrics, whereas Full is lower on both CE-RPUI metrics. This short-window result and the long-window result in Figure 1 refer to different forecast lengths and cohorts. Fresh128 percentage changes form the ratio of cohort means after averaging the fixed seeds. They differ from the mean within-seed changes used in Table 2 and the static-weight study below.

Table 7: Extended-budget controls on Fresh128. Errors are in $\times 10^{-3}$. Conditional intervals describe paired trajectory resampling with the three fitted seeds held fixed.

Context	Metric	Full	Extended sum	Change (%)	95% interval
NS-PwC	Mean	9.127	8.294	-9.12	[-9.4649, -8.7626]
NS-PwC	Final	12.362	11.847	-4.17	[-4.7434, -3.5930]
CE-RPUI	Mean	124.647	127.568	+2.34	[+1.5447, +3.1637]
CE-RPUI	Final	181.995	191.316	+5.12	[+4.0874, +6.2712]

Validation-selected static auxiliary weights. For ordinary-sum training, the auxiliary coefficient is $\beta = r\lambda_a$, with $r \in \{0, 0.03, 0.1, 0.3, 1\}$. The endpoints $r = 0$ and $r = 1$ reuse the recorded no-auxiliary and ordinary-sum configurations. The three intermediate coefficients add 27 training runs across three contexts and three seeds. One coefficient per context is selected by the mean train-tail score across all three seeds before the new Gate96 evaluations begin. The selected effective coefficients are 1 for direct NS-PwC, 0.1 for direct CE-RPUI, and 0.25 for recurrent NS-SL. The selection is not performed separately by seed or by evaluation metric. Gate96 and the endpoint results had been used during development.

Table 8 retains all selected comparisons. Static weighting has lower Mean in the two direct contexts, while Full has lower Mean and Final on recurrent NS-SL. The direct Final intervals span zero. Thus this test does not establish a uniform benefit over a validation-tuned scalar weight.

Table 8: Validation-selected static weighting on Gate96. Errors are in $\times 10^{-3}$. Change is the mean within-seed percentage difference relative to Full. Positive values favor Full. Intervals condition on the fitted models.

Context	Metric	Full	Static	Change (%)	95% interval
NS-PwC (D)	Mean	9.258	9.233	-0.269	[-0.452, -0.088]
NS-PwC (D)	Final	12.677	12.634	-0.330	[-0.765, +0.102]
CE-RPUI (D)	Mean	125.898	125.201	-0.554	[-0.832, -0.276]
CE-RPUI (D)	Final	182.539	183.012	+0.259	[-0.138, +0.683]
NS-SL (R)	Mean	16.558	17.031	+2.881	[+2.252, +3.512]
NS-SL (R)	Final	26.331	27.366	+3.955	[+2.987, +4.930]

The full coefficient grid is reported in Table 9. It is not used to make a second, evaluation-selected coefficient choice. All intervals use 50,000 paired trajectory resamples with fixed fitted seeds and do not include retraining or hyperparameter-selection uncertainty. The main registered-model intervals are not reused for these new comparisons.

Table 9: Complete static-weight sensitivity grid on Gate96. Entries are mean ± sample SD across three seeds in $\times 10^{-3}$.

Configuration	Mean error	Final error
NS-PwC / direct		
Full TEMPO	9.258 ± 0.091	12.677 ± 0.178
$r = 0$	10.066 ± 0.808	15.095 ± 1.377

Configuration	Mean error	Final error
$r = 0.03$	9.634 ± 0.205	14.369 ± 0.550
$r = 0.1$	9.562 ± 0.054	13.879 ± 0.111
$r = 0.3$	9.302 ± 0.170	13.188 ± 0.311
$r = 1$	9.233 ± 0.062	12.634 ± 0.086
CE-RPUI / direct		
Full TEMPO	125.898 ± 0.318	182.539 ± 0.147
$r = 0$	126.214 ± 1.910	184.990 ± 2.199
$r = 0.03$	125.727 ± 0.792	184.058 ± 0.918
$r = 0.1$	125.201 ± 0.671	183.012 ± 0.867
$r = 0.3$	125.371 ± 0.347	182.387 ± 0.302
$r = 1$	127.257 ± 0.305	184.041 ± 0.387
NS-SL / recurrent		
Full TEMPO	16.558 ± 0.307	26.331 ± 0.482
$r = 0$	17.671 ± 0.243	28.525 ± 0.406
$r = 0.03$	17.763 ± 0.321	28.566 ± 0.637
$r = 0.1$	17.575 ± 0.367	28.299 ± 0.604
$r = 0.3$	17.577 ± 0.087	28.288 ± 0.301
$r = 1$	17.031 ± 0.104	27.366 ± 0.186

D Complete Functional Ablation Metrics

D.1 Completed experimental matrix

The completed study contains 90 models, with five variants and three training seeds for each of six task–formulation combinations. Direct experiments cover NS-PwC, CE-RPUI, and FNS-KF. Recurrent experiments cover NS-PwC, NS-SL, and NS-SVS. Each comparison uses development Gate96 over 11 forecast steps. Field-only primary jointly removes auxiliary and structure supervision while retaining the primary time targets and the original selection rule.

Within a column, all variants retain the training data, 1,280 updates, initialization seed, prediction path, and checkpoint-selection rule. The seeds are 9841, 9842, and 9843. The *w/o auxiliary* variant prevents the auxiliary objective from contributing to the parameter gradient while retaining its forward computation. The *w/o structure* variant sets the spectral and spatial-difference weights to zero and retains field supervision in its primary and auxiliary roles. The *ordinary sum* variant retains both objectives and applies global clipping after they are added. It changes the composition and clipping procedure, not projection alone.

The two formulations differ in their objectives, training schedules, and checkpoint-selection criteria. Cross-column differences therefore do not isolate prediction path as the only changing factor. The recurrent NS-PwC study is a mechanism study and does not replace the main-table deployment.

D.2 Absolute errors and seed variation

For each training seed, errors are first averaged over the 96 evaluation trajectories. The reported mean and sample standard deviation are then calculated across the three seed means, using two degrees of freedom. The notation mean ± SD describes variation across these fitted runs. It is not a confidence interval. Tables 10 and 11 retain both forecast metrics for all variants.

Table 10: **Direct-query ablation errors.** Mean ± sample SD over three training seeds on development Gate96. Errors are in $\times 10^{-3}$.

Configuration	NS-PwC	CE-RPUI	FNS-KF
Mean error			
Full TEMPO	9.258 ± 0.091	125.898 ± 0.318	9.546 ± 0.091
w/o auxiliary	10.066 ± 0.808	126.214 ± 1.910	10.063 ± 0.105

Configuration	NS-PwC	CE-RPUI	FNS-KF
w/o structure	9.579 ± 0.159	123.926 ± 0.348	9.840 ± 0.264
Ordinary sum	9.233 ± 0.062	127.257 ± 0.305	9.509 ± 0.220
Field-only primary	10.640 ± 1.013	123.957 ± 0.493	10.553 ± 0.057
Final error			
Full TEMPO	12.677 ± 0.178	182.539 ± 0.147	13.612 ± 0.211
w/o auxiliary	15.095 ± 1.377	184.990 ± 2.199	15.858 ± 0.327
w/o structure	12.996 ± 0.297	180.516 ± 0.108	13.853 ± 0.454
Ordinary sum	12.634 ± 0.086	184.041 ± 0.387	13.397 ± 0.407
Field-only primary	15.879 ± 1.630	182.777 ± 0.304	16.364 ± 0.320

Table 11: **Recurrent ablation errors.** Mean ± sample SD over three training seeds on development Gate96. Errors are in $\times 10^{-3}$.

Configuration	NS-PwC	NS-SL	NS-SVS
Mean error			
Full TEMPO	16.253 ± 0.091	16.558 ± 0.307	19.139 ± 0.191
w/o auxiliary	17.691 ± 0.747	17.671 ± 0.243	20.455 ± 0.443
w/o structure	16.301 ± 0.165	17.135 ± 0.189	19.951 ± 0.587
Ordinary sum	17.072 ± 0.573	17.031 ± 0.104	20.362 ± 0.780
Field-only primary	18.181 ± 0.643	18.084 ± 0.174	20.838 ± 0.315
Final error			
Full TEMPO	25.581 ± 0.128	26.331 ± 0.482	32.708 ± 0.516
w/o auxiliary	28.631 ± 1.383	28.525 ± 0.406	35.156 ± 0.526
w/o structure	25.007 ± 0.561	27.298 ± 0.452	33.846 ± 1.118
Ordinary sum	27.390 ± 1.414	27.366 ± 0.186	35.178 ± 1.482
Field-only primary	28.658 ± 1.192	28.984 ± 0.186	35.544 ± 0.428

D.3 Paired changes and conditional intervals

For variant v, let $\bar{E}_{s,v}$ be the trajectory-averaged error at training seed s. The change reported in the main ablation table is

$$\Delta_v = \frac{100}{3} \sum_{s=1}^{3} \frac{\bar{E}_{s,v} - \bar{E}_{s,\mathrm{Full}}}{\bar{E}_{s,\mathrm{Full}}}.$$

Positive changes indicate higher error after the intervention. This is the mean of three within-seed percentage changes, not a ratio of rounded table means.

The uncertainty analysis jointly resamples the paired trajectory identities 50,000 times while holding the three fitted training seeds fixed. Its pointwise 95% intervals describe trajectory-sampling uncertainty conditional on those models. They are not training-seed-population intervals and are not corrected for multiple comparisons. Table 12 reports each interval alongside the three seed-specific changes so that these two sources of information are not conflated. Intervals come from this resampling analysis, not from marginal standard deviations.

Table 12: **Paired ablation effects and uncertainty.** Positive changes mean higher error than Full. Intervals resample paired trajectories with the three fitted seeds held fixed. Seed columns report the same relative change for each training seed. All values are percentages.

Variant	Metric	Mean change	Conditional 95% interval	Seeds 9841 / 9842 / 9843
Direct query / NS-PwC				
w/o auxiliary	Mean	+8.77	[+8.28, +9.29]	+19.88 / +3.24 / +3.20

Variant	Metric	Mean change	Conditional 95% interval	Seeds 9841 / 9842 / 9843
w/o auxiliary	Final	+19.18	[+18.45, +19.93]	+33.44 / +12.61 / +11.48
w/o structure	Mean	+3.46	[+3.21, +3.71]	+4.97 / +1.66 / +3.76
w/o structure	Final	+2.52	[+2.05, +3.01]	+4.06 / +0.18 / +3.31
Ordinary sum	Mean	-0.27	[-0.45, -0.09]	+1.17 / -0.91 / -1.07
Ordinary sum	Final	-0.33	[-0.76, +0.11]	+1.23 / -1.03 / -1.19
Field-only primary	Mean	+14.98	[+14.43, +15.58]	+28.73 / +9.03 / +7.19
Field-only primary	Final	+25.38	[+24.46, +26.37]	+41.99 / +19.14 / +15.02
Direct query / CE-RPUI				
w/o auxiliary	Mean	+0.25	[-0.13, +0.64]	-0.33 / -0.67 / +1.75
w/o auxiliary	Final	+1.34	[+0.74, +1.98]	+1.45 / +0.15 / +2.43
w/o structure	Mean	-1.57	[-1.78, -1.38]	-1.86 / -1.39 / -1.45
w/o structure	Final	-1.11	[-1.39, -0.84]	-1.11 / -1.03 / -1.18
Ordinary sum	Mean	+1.08	[+0.85, +1.33]	+1.07 / +1.56 / +0.61
Ordinary sum	Final	+0.82	[+0.49, +1.18]	+0.98 / +0.72 / +0.77
Field-only primary	Mean	-1.54	[-1.95, -1.15]	-1.82 / -1.51 / -1.29
Field-only primary	Final	+0.13	[-0.41, +0.68]	-0.12 / +0.36 / +0.15
Direct query / FNS-KF				
w/o auxiliary	Mean	+5.43	[+4.93, +5.94]	+6.93 / +3.43 / +5.92
w/o auxiliary	Final	+16.52	[+15.67, +17.40]	+19.83 / +12.82 / +16.93
w/o structure	Mean	+3.07	[+2.84, +3.29]	+1.07 / +5.04 / +3.10
w/o structure	Final	+1.76	[+1.35, +2.16]	-0.51 / +3.68 / +2.10
Ordinary sum	Mean	-0.40	[-0.60, -0.20]	-1.08 / +1.17 / -1.28
Ordinary sum	Final	-1.59	[-1.97, -1.23]	-2.10 / +0.04 / -2.73
Field-only primary	Mean	+10.56	[+9.92, +11.25]	+11.21 / +8.72 / +11.76
Field-only primary	Final	+20.26	[+19.16, +21.40]	+22.19 / +15.49 / +23.09
Recurrent rollout / NS-PwC				
w/o auxiliary	Mean	+8.85	[+8.02, +9.71]	+13.79 / +4.40 / +8.37
w/o auxiliary	Final	+11.94	[+10.66, +13.25]	+17.99 / +6.05 / +11.79
w/o structure	Mean	+0.30	[-0.36, +0.93]	-0.11 / -1.04 / +2.04
w/o structure	Final	-2.24	[-3.26, -1.24]	-2.60 / -4.35 / +0.22
Ordinary sum	Mean	+5.05	[+4.35, +5.76]	+7.53 / +0.52 / +7.09
Ordinary sum	Final	+7.08	[+5.99, +8.23]	+9.61 / +0.34 / +11.31
Field-only primary	Mean	+11.86	[+11.14, +12.59]	+16.03 / +10.54 / +9.01
Field-only primary	Final	+12.04	[+10.98, +13.08]	+17.62 / +10.71 / +7.78
Recurrent rollout / NS-SL				
w/o auxiliary	Mean	+6.73	[+5.90, +7.56]	+6.34 / +5.50 / +8.35
w/o auxiliary	Final	+8.35	[+7.13, +9.59]	+7.97 / +6.27 / +10.82
w/o structure	Mean	+3.51	[+2.84, +4.17]	+1.59 / +4.67 / +4.26
w/o structure	Final	+3.69	[+2.71, +4.68]	+1.48 / +5.20 / +4.38
Ordinary sum	Mean	+2.88	[+2.25, +3.51]	+0.61 / +3.62 / +4.41
Ordinary sum	Final	+3.95	[+2.98, +4.93]	+1.71 / +4.32 / +5.83
Field-only primary	Mean	+9.23	[+8.39, +10.06]	+8.34 / +9.03 / +10.33
Field-only primary	Final	+10.09	[+8.80, +11.36]	+9.13 / +9.18 / +11.97
Recurrent rollout / NS-SVS				
w/o auxiliary	Mean	+6.88	[+6.10, +7.67]	+9.26 / +6.36 / +5.01
w/o auxiliary	Final	+7.49	[+6.25, +8.73]	+8.99 / +6.67 / +6.81
w/o structure	Mean	+4.24	[+3.37, +5.07]	+7.12 / +4.12 / +1.48
w/o structure	Final	+3.49	[+2.03, +4.91]	+7.79 / +1.38 / +1.31
Ordinary sum	Mean	+6.40	[+5.50, +7.31]	+11.27 / +4.52 / +3.40
Ordinary sum	Final	+7.57	[+6.15, +8.99]	+13.20 / +4.96 / +4.55
Field-only primary	Mean	+8.87	[+7.96, +9.79]	+10.19 / +8.71 / +7.72
Field-only primary	Final	+8.68	[+7.28, +10.04]	+9.86 / +7.78 / +8.39

Table 13: Detailed time-sampling and structure factors at the original 1,280-update budget. Values use 10^{-3} units and summarize all three fixed seeds. This table is separate from the full-configuration deletions in Table 2.

			NS-PwC		CE-RPUI	
Sampling	Structure	Update	Mean	Final	Mean	Final
IID	Off	Sum	9.586	12.928	125.958	182.161
Antithetic	Off	Sum	9.669	13.092	125.640	182.075
IID	On	Sum	9.254	12.637	127.626	183.885
Antithetic	On	Sum	9.233	12.634	127.257	184.041
Antithetic	On	Bounded	9.258	12.677	125.898	182.539

D.4 Sampling factors and extended-budget controls

A separate IID/antithetic and structure-off/on study tests factors within direct-query training. Table 13 retains that study separately from the functional interventions above. Its rows are not relabeled as joint deletions from the Full configuration.

The extended ordinary-sum schedules use 2,944 updates on NS-PwC and 3,200 on CE-RPUI. They change schedule length and validation opportunities as well as training duration. On Gate96, the extended sum has Mean 8.457 on NS-PwC and 127.811 on CE-RPUI, compared with 9.258 and 125.898 for Full ($\times 10^{-3}$). These are extended-budget controls, not exact compute matches. Their short-window values must not be substituted for the prefixes of the separate long-reference study.

D.5 Spatial residual diagnostics

Figure 3 uses 27 spectral evaluations covering Full, w/o structure, and ordinary sum for three seeds on direct NS-PwC, direct CE-RPUI, and recurrent NS-SL. The nine ordinary-sum evaluations reproduce the corresponding stored trajectory-level Mean and Final errors. Within each column, both plotted rows use the same predictions. The top row reports normalized residual power, and the bottom row divides each control's mean power by Full. The curves do not represent confidence intervals.

Normalization and averaging. Let $\mathcal{F}$ be the orthonormal two-dimensional Fourier transform and B_k the set of Fourier modes assigned to radial shell k. Each shell groups Fourier modes with similar magnitudes of the two-dimensional wavevector. For training seed s, trajectory i, forecast step h, and dynamic variable c, we normalize the residual shell power by the total reference energy of that same field:

$$P_{s,i,h,c}(k) = \frac{\sum_{q \in B_k} |\mathcal{F}(\hat{x}_{s,i,h,c} - x_{i,h,c})(q)|^2}{\sum_{q} |\mathcal{F}(x_{i,h,c})(q)|^2}.$$

The denominator includes all Fourier modes, not only shell k. Normalization is performed before averaging. Within each trajectory and step, valid dynamic variables receive equal weight. We then average over forecast steps, trajectories, and fitted training seeds. A variable with exactly zero total reference energy is excluded at that step for all methods and its exclusion is recorded. A frame with no valid variable stops the analysis rather than being silently omitted. The lower row divides the resulting mean control spectrum by the mean Full spectrum. It is not a mean of individual power ratios. Band summaries sum the shell powers before forming the ratio. Summing all shells recovers the squared relative-error decomposition for nonzero reference fields, not Table 2's unsquared Mean.

All non-DC shells from 1 through 91 are shown. DC and the complete 92-shell numerical results are not shown. The fixed summary bands are 1–8, 9–32, and 33–64. The last is a diagnostic interval, not a universal physical threshold or the set of all high-frequency modes. Its ratios are formed by summing band power and then taking the ratio, not by averaging per-shell ratios. The band-33–64 ratios of w/o structure to Full are 1.588407 for direct NS-PwC, 0.887456 for direct CE-RPUI, and 1.484502 for recurrent NS-SL. Residual power is a squared-error diagnostic. Its percentage changes

are not numerically interchangeable with the unsquared, channel-averaged relative-L_2 changes in Table 2.

E Long Prediction and Cross-Interface Transfer

E.1 NS-PwC long-forecast identity and rollout

Figure 1 uses a source-disjoint repair of a previously examined development cohort; it is not a new blind test. The source repair retains 70 eligible trajectories and deterministically replaces 26 with trajectories outside the registered training, selection, and reserved-calibration identities. The same 31 fixed checkpoints, organized into thirteen configuration groups, are evaluated on the 96 repaired trajectories. The figure shows a subset with explicit comparison roles rather than selecting a new model for each horizon.

The standard short reference has 21 frames. Starting from observed frame 9 gives eleven future states. The long reference has 257 frames, so the same starting index gives 247 future states, with $\Delta t = 0.05$. Direct forecasts advance in eleven-step blocks using predicted endpoints as subsequent block inputs. They do not query lead 247 from the original observed state in one model call. The long reference is a numerical continuation rather than an official long-horizon test set. The records used here do not establish a spatial-convergence study for that continuation.

E.2 Figure 1 comparisons

Both panels divide mean per-step errors by matched TEMPO at the original learning rate. The left panel includes the registered FT and temporal-supervision controls, each retaining its own training recipe. The right panel includes paired ordinary sum at 1,280 and 2,944 updates, IID sampling controls, and the half-learning-rate configurations. The matched composition comparison is between original-learning-rate TEMPO and paired ordinary sum at the same 1,280-update budget. The reference line equals one by normalization, not because TEMPO's absolute error is constant.

The full-window Mean comparison first averages raw errors over time and then forms the ratio. It is not the average of the figure's per-step ratios. Table 14 records the main short- and long-window comparisons and their conditional trajectory intervals. The first eleven steps here belong to the repaired long-source cohort, not the standard Gate96 observations in Table 2.

Table 14: **Matched NS-PwC prefix comparisons.** Mean error change relative to TEMPO in percent. Positive is worse. Intervals are conditional paired trajectory intervals for the three fixed fitted seeds.

Control	Prefix length	Change (%)	Conditional 95% interval
Ordinary sum	11	-0.099	[-0.306, +0.102]
Ordinary sum	247	+7.486	[+4.566, +10.461]
Extended ordinary sum	11	-9.445	[-9.865, -8.994]
Extended ordinary sum	247	+8.031	[+4.494, +11.550]

The extended sum control has lower prefix Mean for lengths 1–35 and higher prefix Mean for lengths 36–247. Its per-step error is consistently higher from step 24 onward. These crossings describe the saved predictions and are not trained switching rules or statistically established universal thresholds. The intervals hold fitted seeds fixed and are pointwise, not simultaneous confidence bands over all 247 prefix lengths. A small observed short-window difference is not an equivalence test.

Sensitivity to the lead-sampling realization. A second recorded realization changes the lead-sampling stream and reuses seed IDs 9841 and 9842. On the 70 common source trajectories, the ordinary-sum excess is 0.98%, with individual seed changes of +4.27% and -2.21%, compared with 7.41% for the first realization on that subset. These are sensitivity results, not an independent five-seed replication. The smaller gap limits claims that the composition gain has a fixed magnitude across training realizations.

The long-forecast study does not alter the main-table checkpoints or the separate one-shot test.

E.3 FLOWER-TINY SCORE-TRANSFER RESULTS

The score-transfer experiment trains Flower-Tiny from random initialization with a single-state input interface. It uses 10,240 updates, batch size four, a four-step training window, and per-step gradient detachment. The training seeds are 9701, 9702, and 9703. The native loss is compared with the TEMPO per-step score. This transfer contains no paired-lead sampling, auxiliary objective, or gradient composition. It differs from the four-frame input recipe of the original Flowers implementation.

Both objectives use learning rate 0.001 except for CE-RPUI, where the TEMPO-score configuration uses 0.0003. These rates were selected during development. The CE row therefore changes more than the loss and is not used as an isolated loss-only effect. Sample standard deviations and improved-seed counts come from the recorded per-seed results.

Table 15: **Flower-Tiny score transfer on the five displayed tasks.** Errors are mean ± sample SD across three training seeds in $\times 10^{-3}$. The count gives the seeds with lower error under the TEMPO score. CE-RPUI changes both the score and the development-selected learning rate.

Task	Metric	Native	TEMPO score	Improved seeds
NS-PwC	Mean	15.335 ± 1.051	10.933 ± 0.106	3/3
NS-PwC	Final	25.849 ± 2.043	18.258 ± 0.088	3/3
NS-SL	Mean	23.968 ± 0.421	23.159 ± 0.923	2/3
NS-SL	Final	41.428 ± 0.906	42.045 ± 1.421	1/3
NS-SVS	Mean	23.702 ± 0.099	23.343 ± 0.292	3/3
NS-SVS	Final	43.215 ± 0.259	44.029 ± 0.448	0/3
FNS-KF	Mean	23.695 ± 3.702	13.809 ± 1.730	3/3
FNS-KF	Final	41.385 ± 6.937	23.886 ± 3.273	3/3
CE-RPUI	Mean	191.725 ± 8.857	171.296 ± 3.489	3/3
CE-RPUI	Final	289.684 ± 8.090	262.975 ± 5.665	3/3

Among the five displayed tasks, Mean decreases on all five and Final on three. For NS-SL and NS-SVS, a lower Mean coexists with a higher Final. This summary applies only to the displayed subset.

E.4 FLOWERS LONG FORECASTS AND CORRELATION RETENTION

The separate 119-step study retains its eight original checkpoints, covering two objectives, two learning rates, and seeds 9501 and 9502. Its source-identity repair does not change the checkpoints or rates. Table 3b uses the matched learning rate 0.001. Table 16 also retains the 0.0003 pair. These fitted seeds are distinct from the three-seed short-window study and are not pooled into a five-seed experiment.

Table 16: **All retained Flower-Tiny 119-step configurations.** Errors use $\times 10^{-3}$. Retention is the percentage of forecasts with correlation at least 0.8 for the full window. Each row averages two fixed training seeds.

Objective	Learning rate	Mean error	Final error	Retained (%)
Native	0.001	272.341	660.633	85.94
TEMPO score	0.001	200.933	519.495	92.71
Native	0.0003	355.922	812.352	62.50
TEMPO score	0.0003	254.249	625.487	87.50

For prediction $\hat{x}_h$ and reference x_h, spatial correlation compares their centered spatial patterns. The recorded evaluator centers its flattened dynamic-field inputs and computes their normalized inner product. A forecast is counted as retained if this correlation stays at or above 0.8 at all 119 steps. At learning rate 0.001, the retained counts are 81 and 84 out of 96 for Native, and 87 and 91 for

TEMPO score. These give retention rates of 85.94% and 92.71%. Correlation complements field error but does not replace it, since pattern alignment alone does not measure amplitude accuracy.

The repaired matched-rate Mean reduction is 26.22%. Final decreases by 21.36%. The study remains a development comparison and does not update the primary Poseidon-B result.

E.5 Matched post-training on DPOT-L

This study uses the official pretrained DPOT-L weights and compares post-training configurations within the same backbone. For each task, training uses 100 registered target trajectories, a ten-frame input history, and a four-step recurrent training window. The carried state is detached after each training step. All configurations use 1,280 updates with effective batch size eight, implemented by micro-batch size one and eight accumulation steps. Training and evaluation use FP32. AdamW uses learning rate 10^{-4}, weight decay 10^{-3}, and a OneCycle schedule with `pct_start=0.3`. The two training seeds are 8401 and 8402. The final-update checkpoint is evaluated without selection on Fresh128. For the five main tasks, evaluation uses the same 128 trajectories within each task and eleven future prediction steps.

Compared objectives. MSE-H4, denoted B0 in the experiment records, uses normalized-state MSE and global gradient clipping at one. Score-H4, denoted C0, replaces MSE with the exact four-step average of the field, spectral, and spatial-difference scores, with both structure weights set to 0.01. It has no auxiliary objective and uses the same global clipping bound. The clip-two sensitivity control changes only this clipping bound and was evaluated on NS-PwC, NS-SVS, and CE-RPUI.

TEMPO-AR uses antithetically sampled recurrent scores as its primary objective and a recurrent endpoint risk as its auxiliary objective. It applies whole-vector conflict projection, primary clipping at one, an auxiliary norm cap relative to the clipped primary, and auxiliary coefficient one. This is a formulation for DPOT's history-based recurrent interface. Its auxiliary objective is not the direct-query objective of Eq. (8).

Table 17: **Matched DPOT-L post-training on the five main tasks.** Mean and Final relative-L_2 errors in units of 10^{-3}, averaged over two training seeds. All entries in a task use the same Fresh128 evaluation trajectories. Lower is better. – denotes a sensitivity configuration that was not evaluated.

Configuration	NS-PwC	NS-SL	NS-SVS	FNS-KF	CE-RPUI
Mean error					
MSE-H4	49.635	50.674	29.475	51.584	306.454
Score-H4	41.771	40.904	25.655	43.582	264.567
Score-H4, clip 2	41.733	–	25.726	–	264.772
TEMPO-AR	42.154	45.090	28.603	44.348	272.551
Final error					
MSE-H4	75.872	79.062	48.563	79.263	392.845
Score-H4	65.106	65.085	44.448	68.346	346.138
Score-H4, clip 2	65.032	–	44.456	–	346.677
TEMPO-AR	64.285	68.690	46.723	68.094	349.328

Interpretation. Both Score-H4 and TEMPO-AR improve Mean and Final over MSE-H4 on the five main tasks. Score-H4 has lower Mean than TEMPO-AR on these tasks, so the comparisons support the transfer of forecast scoring without establishing a uniform additional gain from gradient composition. TEMPO-AR has lower Final on NS-PwC and FNS-KF than Score-H4. The conditional paired interval includes zero for the FNS-KF difference. The clip-two controls retain the same broad comparison pattern on the three evaluated tasks.

F Inference Measurement and Total Adaptation Cost

Measurement protocol. The completed timing study evaluates fixed deployment configurations on an RTX 5090 at batch size one. Each configuration receives twenty warm-up forecasts and then

Table 18: **Median and interquartile range of complete-forecast latency.** Values are milliseconds. The intervals are the observed 25th and 75th percentiles, not confidence intervals.

Task	Matched-path FT	TEMPO
NS-PwC	729.353 [725.099, 735.012]	729.384 [725.772, 736.362]
NS-SL	768.019 [753.235, 775.372]	771.682 [758.841, 801.946]
NS-SVS	710.643 [706.394, 716.397]	712.941 [708.888, 785.146]
FNS-KF	709.921 [706.501, 713.022]	710.843 [706.479, 757.409]
CE-RPUI	728.981 [725.360, 731.597]	731.462 [727.101, 809.935]

thirty timing cycles over four fixed Gate96 inputs, giving 120 recorded forecasts. All five displayed tasks use eleven forecast steps. Evaluation mode disables training losses and backpropagation.

Timing boundary. CUDA is synchronized before and after each complete prediction. Timing includes model calls, normalization, decoding, history updates, finite-value checks, and prediction-input packing. It excludes model loading, data I/O, the initial host-to-device transfer, initialization, reference scoring, output serialization, and warm-up. Both registered eleven-step paths generate every requested frame and execute eleven model calls. Table 18 retains the measured distributions.

Interpretation. Execution order is balanced overall but fixed within each input. Very small median differences cannot be separated completely from ordering effects. The results support comparable observed median latency on the measured checkpoints, inputs, and device, not zero overhead or cross-architecture speedup. The five displayed rows are distinct matched-path timing comparisons. No numerical-solver speedup is inferred from these measurements.

Adaptation cost. The five-task candidate portfolios contain thirty runs each. Their recorded training-stage durations total 51.35 hours for TEMPO and 28.75 hours for the baseline portfolio. These are not the costs of individual deployed models, exclusive GPU-hours, or complete research costs. The twelve-run recurrent NS-PwC study adds 22.21 hours of recorded training-stage time and is kept separate from the candidate portfolios. Selected checkpoints are subsets of their training studies and are not counted again. All outcomes of the 90-model ablation matrix are reported in Appendix D. The portfolio totals do not include the cost of the ablation study.