FUTURE-GUIDED ON-POLICY SELF-DISTILLATION FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting has a training asymmetry: each training example eventually reveals the future, but a deployed model sees only history. That future contains both history-predictable structure and innovations. Standard pointwise supervision uses the realised values as targets; can their ordered structure also guide learning without entering the deployed forecast? FoldOPSD turns this training-time context into on-policy self-distillation. A teacher view of the current forecaster receives summaries of observed future segments, while a history-only student uses a learned constant correction. Their shared forecasting body lets teacher supervision update the model that will be deployed; detached residual matching and an EMA student add teaching targets. Because both views share the forecasting body, the answer changes the training update rather than the deployed input. Analysis identifies when ideal answer conditioning reduces gradient noise and how answer hints reweight residual directions. At a fixed DLinear/ETTm2 checkpoint, the joint shared-body gradient lowers reference MSE by 28.3% on training and 35.0% on validation, while across-batch variance falls by 33.4% and 33.9%, respectively. With a compatible native output bias, the student correction folds exactly into that bias, adding no inference operations. Across four datasets and five seeds, FoldOPSD lowers mean test MSE in all 24 DLinear and iTransformer settings, by up to 10.85%. True future hints outperform noise and dataset-level summaries in all 16 matched ETT settings with the basic OPSD objective.

1 INTRODUCTION

Forecasting has a structural asymmetry: the future is available when a training example is formed, but not when its prediction is deployed. Under squared error, the deployed model therefore learns the conditional mean of the future given history alone. Existing architectures improve this mapping by exposing trend–seasonal structure, multiple temporal resolutions, and cross-variable dependencies (Wu et al., 2021; Zeng et al., 2023; Wu et al., 2023; Wang et al., 2025; Nie et al., 2023; Liu et al., 2024). Yet the realised future is more than a pointwise target: once history is fixed, its ordered segments reveal level, variability, and trend. The central question is whether this answer structure can shape how the shared forecaster learns while its deployed input remains history-only.

Training-only supervision provides the natural interface for this question: information can shape parameter learning while remaining absent from the deployed computation. Privileged-information learning and generalised distillation establish this separation (Vapnik and Vashist, 2009; Vapnik and Izmailov, 2015; Lopez-Paz et al., 2016); TimeKD brings future information through a separate language-model teacher (Liu et al., 2025), while OPSD lets a reference-conditioned view of the current model teach its own rollouts (Zhao et al., 2026). Forecasting sharpens this distinction: deployment still requires the history-only conditional mean, so future context is useful only when the teacher changes the update of the shared body that the student will deploy.

FoldOPSD instantiates this shared-body construction by making answer access, rather than model identity, distinguish teacher from student. The teacher receives ordered future-segment answer hints, while the student uses a learned constant through the same correction interface. Teacher supervision changes the update of the body that will be deployed; detached residual matching and an EMA student provide current and EMA targets. The answer is thus a training scaffold, not an inference feature. On compact hosts, a compatible native bias absorbs the student's constant correction (Lin et al., 2024; Ma et al., 2026), so the deployed predictor remains history-only.

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.

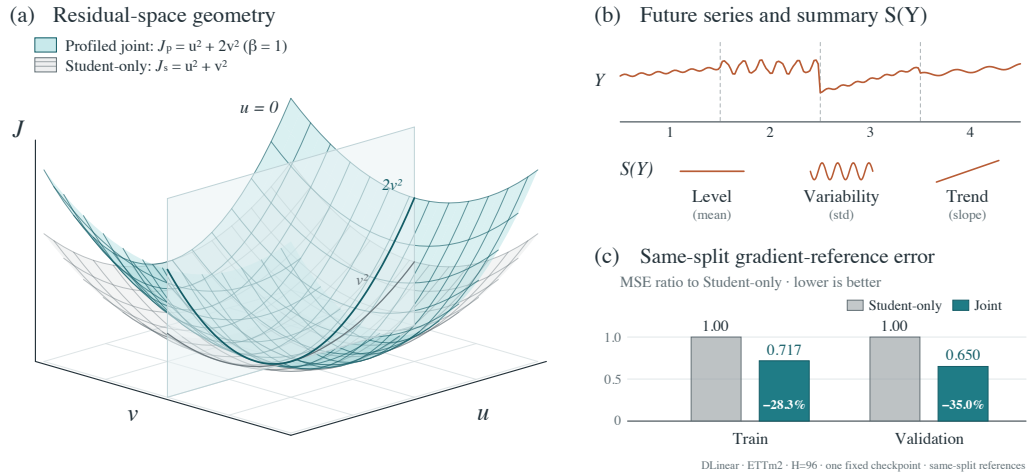

Figure 1: Answer structure and the shared learning signal. **(a)** An analytic two-direction slice of the profiled criterion developed in Methods: the summary-aligned residual direction retains unit weight, while the orthogonal direction receives weight two. **(b)** Ordered future patches supply local mean, standard deviation, and least-squares slope as teacher hints. **(c)** At one fixed DLinear/ETTm2 checkpoint ($H = 96$), the normalized joint gradient reduces gradient-error MSE to 0.717 and 0.650 times the student-only value on disjoint same-split training and validation references, respectively.

The statistical reason this scaffold can help is that squared-error learning mixes history-predictable structure with realised innovation. An answer summary can expose innovation that becomes predictable after the future is observed; when its correction is conditionally centred, shared supervision preserves the history-only gradient mean while reducing conditional covariance. The analysis below turns this principle into a residual-geometry criterion and tests the learned correction directly.

The practical objective turns answer context into shared-body updates without changing the student's input. Future-segment *answer hints* condition the teacher; a learned constant conditions the student. Teacher supervision updates the shared body, detached residual matching supplies a current-teacher target, and an EMA student adds historical guidance (Tarvainen and Valpola, 2017). A separate quadratic analysis describes how the learned interface can reweight residual directions. A fixed-checkpoint probe then measures gradient variation and mean shift (Figure 1).

Deployment retains only the history-only student. When a compatible native output bias exists, its constant correction merges into that bias before input-scale restoration (Ding et al., 2019; 2021b). The substitution preserves its forecast and the host's inference operations in exact arithmetic. For these hosts, answer context selects a predictor from the unchanged deployment function class.

Matched experiments show that answer-derived teaching improves the deployed history-only student in the main evaluation. Across five seeds, mean test MSE decreases in all 24 main cells on DLinear and iTransformer across four datasets, by up to 10.85%. Under basic OPSD, true hints outperform noise and dataset-level summaries in all 16 matched ETT settings, supporting the value of example-specific answer context. Loss interventions test the shared teacher path in five configurations. A separate 96-cell, four-host study finds MSE improvements in 68 cells and maps architectural heterogeneity. Answer-derived teaching thus improves learning while preserving history-only deployment.

2 Related Work

Forecasting and privileged information. Forecasting research has used temporal structure in the observed history and, separately, information available only during training. Existing architectures encode temporal and cross-variable relationships (Zhou et al., 2021; Wu et al., 2021; Nie et al., 2023; Zeng et al., 2023; Liu et al., 2024; Wu et al., 2023; Wang et al., 2025). Privileged-information learning (Vapnik and Vashist, 2009; Vapnik and Izmailov, 2015) and generalised distillation (Lopez-Paz et al., 2016) use training-only inputs to guide a deployable student (Karlsson et al., 2022). In forecasting, TimeKD supplies future information through a separate language-model teacher (Liu et al., 2025); TimeDistill uses cross-architecture distillation into an MLP student (Ni et al., 2026). FoldOPSD instead uses numerical future summaries to condition a view of the current forecasting body.

On-policy self-distillation and historical guidance. OPSD makes a view of the current model, rather than an independent teacher, the source of its student's targets. Classical distillation transfers a teacher's outputs to a student (Hinton et al., 2015); self-distillation uses model generations, peers, or network depths (Furlanello et al., 2018; Zhang et al., 2018; 2019).

FoldOPSD uses answer hints to reweight the shared body during training, not as an inference feature; folding leaves the deployed predictor history-only.

In OPSD, a reference-conditioned view of the current model teaches its own rollouts (Zhao et al., 2026). FoldOPSD carries that current-model relationship into full-horizon regression: teacher supervision updates the shared forecasting body, while a detached teacher residual guides the student. An EMA student adds historical guidance through parameter averaging (Polyak and Juditsky, 1992; Izmailov et al., 2018) and Mean Teacher (Tarvainen and Valpola, 2017). Matched hint controls separate answer content from its interface, as in OPSD (Ichihara et al., 2026).

Structural reparameterisation for deployment. Structural reparameterisation removes a training-time path when its effect can be absorbed into an equivalent inference operator. ACNet, RepVGG, and diverse branch blocks merge inference branches (Ding et al., 2019; 2021b;a; Vasu et al., 2023). FoldOPSD applies the same equivalence principle to a different object: its constant student correction. For hosts with a compatible native bias, the correction merges into that bias before input-dependent scale restoration (Kim et al., 2022). Table 1 isolates the additions to OPSD: shared-body teacher supervision and exact bias folding in compatible hosts.

Table 1: Teaching and deployment mechanisms. FoldOPSD uses answer hints to supervise the current forecasting body and folds the student's correction exactly into a compatible native output bias.

	Training mechanism			Deployment	
Approach	Privileged input	Shared current weights	Teacher loss to shared body	Exact branch fusion	Teacher-free inference
Generalised distillation (Lopez-Paz et al., 2016)	✓	✗	✗	✗	✓
TimeKD (Liu et al., 2025)	✓	✗	✗	✗	✓
OPSD (Zhao et al., 2026)	✓	✓	✗	✗	✓
Mean Teacher (Tarvainen and Valpola, 2017)	✗	✗	✗	✗	✓
RepVGG (Ding et al., 2021b)	✗	✗	✗	✓	✓
FoldOPSD	✓	✓	✓	✓	✓

✓: present; ✗: absent in the cited construction. Shared weights means the teacher and student use the same current body, not an EMA copy. Teacher loss means direct supervision through the teacher view to that shared body, rather than a detached distillation target. Exact fusion merges a training branch into inference parameters.

3 METHODS

Future observations can guide training but cannot enter a history-only forecast. FoldOPSD uses them as a training-time intervention on the shared deployment body's geometry, not as a better inference target: the teacher sees answer hints, the student a constant slot, and their shared body receives teacher supervision. The teacher and hint extractor vanish at deployment, while a compatible native bias absorbs the constant student correction; we derive the resulting fusion condition.

3.1 FORECASTING SETUP

The correction enters before input-scale restoration so that a constant student correction can merge into a compatible native bias. Given L observations of D channels, $\mathbf{X} \in \mathbb{R}^{L\times D}$, the task is to predict the next H steps, $\mathbf{Y} \in \mathbb{R}^{H\times D}$. The host computes $N_{\mathbf{X}}(f_\theta(\mathbf{X}))$, where f_θ is its forecasting body and $N_{\mathbf{X}}$ restores the scale and offset removed from the input. All restoration statistics are computed from $\mathbf{X}$. For DLinear, $N_{\mathbf{X}}$ is the identity. Normalised hosts use the affine map $N_{\mathbf{X}}(z) = a_{\mathbf{X}} \odot z + m_{\mathbf{X}}$, with scale $a_{\mathbf{X}}$, offset $m_{\mathbf{X}}$, and element-wise product $\odot$. Because the correction is added at this location, it can merge into a compatible native bias even with input-dependent scale restoration.

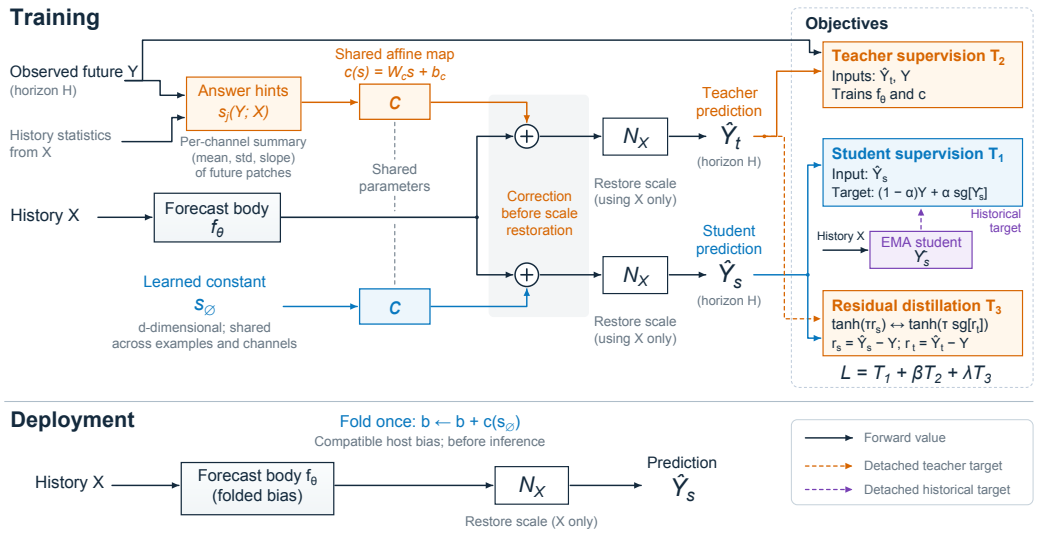

Figure 2: On-policy self-teaching and host-only inference. The shared map c converts the teacher's answer hints and the student's constant into corrections before scale restoration $N_{\mathbf{X}}$. Teacher supervision T_2 updates the shared body and correction, T_1 supplies the EMA-blended student target, and T_3 matches bounded residuals. Solid edges carry forward values. Orange and purple dashed edges carry detached current-teacher and EMA-student targets. Folding absorbs the constant student correction into a compatible native bias b.

3.2 Answer-conditioned views of the current model

Answer hints describe each future segment's level, variability, and trend to the teacher.

Each channel is divided into $M = \lfloor H/P \rfloor$ non-overlapping patches of P steps (Nie et al., 2023). A patch contributes its mean, standard deviation, and least-squares slope; concatenating them in patch order retains their temporal locations. We centre the means by the lookback mean, scale all statistics by a stabilised lookback standard deviation, and clip to $[-10, 10]$. These *answer hints* form a descriptor $\mathbf{s}_j(\mathbf{Y}; \mathbf{X}) \in \mathbb{R}^d$ for channel j, with $d = 3M$ (Appendix A.3).

Patch length controls the granularity of this answer view. We use $P = 24$ on ETTh1, Weather, and Traffic, and $P = 48$ on ETTm2. A 96-step horizon gives 12 or 6 statistics per channel. Trailing incomplete patches are omitted from the descriptor, not the prediction target. Different trajectories can share these statistics, making the context coarse.

One shared affine map turns answer hints and a constant student slot into two views of the current host: $\mathrm{c}(s) = W_{\mathrm{c}} s + b_{\mathrm{c}}$, with $W_{\mathrm{c}} \in \mathbb{R}^{H \times d}$ and $b_{\mathrm{c}} \in \mathbb{R}^H$. The teacher supplies its channel descriptor; the student uses a learned vector $\mathbf{s}_\varnothing \in \mathbb{R}^d$ shared across examples and channels. The full-horizon predictions are

$$\hat{\mathbf{Y}}_t = N_{\mathbf{X}}\big(f_\theta(\mathbf{X}) + [\mathrm{c}(\mathbf{s}_j(\mathbf{Y}; \mathbf{X}))]_{j=1}^{D}\big)\,,$$
$$\hat{\mathbf{Y}}_s = N_{\mathbf{X}}\big(f_\theta(\mathbf{X}) + \mathrm{c}(\mathbf{s}_\varnothing)\mathbf{1}_D^{\top}\big)\,, \tag{1}$$

where $\mathbf{1}_D$ is the length-D vector of ones. Both views use the same current parameters and are recomputed on the same training batch at every update. This is the current-model teaching relationship we inherit from OPSD (Zhao et al., 2026): for direct regression, the full-horizon forecast takes the place of a student rollout. Only the teacher has answer context; the constant student slot is a learned parameter whose constant correction can be retained or folded at export.

Zero initialisation makes both views begin with the host forecast. We set W_{c}, b_{c}, and $\mathbf{s}_\varnothing$ to zero; the interface adds $Hd + H + d$ trainable parameters. The teacher replays the student's random state to match dropout masks, and the student alone updates batch-normalisation running statistics.

3.3 On-policy self-distillation with shared supervision

The current teacher and an EMA student guide learning at different time scales. Teacher supervision updates the shared body, while a detached teacher residual supplies a target for the current student.

The EMA student instead supplies EMA guidance (Tarvainen and Valpola, 2017). Let ϕ collect θ, W_c, b_c, and $\mathbf{s}_\varnothing$. Initialised from the current model, its EMA is $\bar{\phi}_k = \rho\bar{\phi}_{k-1} + (1-\rho)\phi_k$, with update index k and decay ρ. In evaluation mode without gradients, it predicts $\bar{\mathbf{Y}}_s$ from history and its constant slot; it receives no answer hints. The current teacher is recomputed every update; the EMA averages earlier students. At $\rho = 0.999$, the half-life is $\log(0.5)/\log(\rho) \approx 693$ optimizer updates; the equivalent number of epochs depends on the batches per epoch.

The loss combines an EMA student target, current-teacher supervision, and residual matching. Let $\ell(A, B)$ be MSE averaged over batch elements, forecast steps, and channels. Define student and teacher residuals $r_s = \hat{\mathbf{Y}}_s - \mathbf{Y}$ and $r_t = \hat{\mathbf{Y}}_t - \mathbf{Y}$. The objective is

$$T_1 = \ell\Big(\hat{\mathbf{Y}}_s, (1-\alpha)\mathbf{Y} + \alpha\,\mathrm{sg}[\bar{\mathbf{Y}}_s]\Big), \tag{2}$$

$$T_2 = \ell(\hat{\mathbf{Y}}_t, \mathbf{Y}), \tag{3}$$

$$T_3 = \ell(\tanh(\tau r_s), \tanh(\tau\,\mathrm{sg}[r_t])), \tag{4}$$

$$\mathcal{L} = T_1 + \beta T_2 + \lambda T_3, \tag{5}$$

where sg stops gradients through its argument. T_1 fits the student to the observed future blended with its EMA prediction, with mixture weight α. T_2 fits the future-conditioned teacher and updates both the shared body and the correction map, with loss weight β. T_3 matches the student's residual to a detached current-teacher residual, with loss weight λ.

The bounded residual target in T_3 changes how teacher errors guide the student. Near zero, $\tanh(\tau r) \approx \tau r$ retains residual magnitude up to scale; for larger residuals it approaches sign, bounding each squared discrepancy by four. The scale τ sets saturation, while λ sets the weight of distillation. We use $\alpha = 0.5$, $\rho = 0.999$, $\beta = 1$, $\lambda = 0.04$, and $\tau = 10$ by default. Appendix A.4 defines alternatives that separately change the EMA target and this residual distance.

3.4 How answer context changes shared learning

Weight sharing makes the teacher's supervised loss a learning signal for the deployed predictor. Let $Q = [W_c(\mathbf{s}_j - \mathbf{s}_\varnothing)]_{j=1}^{D}$ be the difference between the two corrections. Under affine scale restoration,

$$r_t = r_s + a_{\mathbf{X}} \odot Q, \qquad \frac{\partial T_2}{\partial z} = \frac{2}{K} a_{\mathbf{X}} \odot (r_s + a_{\mathbf{X}} \odot Q), \quad z = f_\theta(\mathbf{X}), \tag{6}$$

where K counts the scalar forecast targets in the batch. Relative to ordinary student-label MSE, teacher supervision adds a hint-dependent term proportional to $a_{\mathbf{X}}^2 \odot Q$ to the error signal entering the shared body. Thus the future descriptor can change the body's update without appearing in the student's input. This derivative holds rescaling parameters fixed; trainable normalisation parameters are updated normally. The common correction bias cancels in Q.

The derivative identifies how answer context enters the update; its statistical value depends on whether the added signal preserves the desired target. Squared-error forecasting separates the learnable target $\mu(\mathbf{X}) = \mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$ from the realised innovation $\varepsilon = \mathbf{Y} - \mu(\mathbf{X})$. For $q^* = \mathbb{E}[\varepsilon \mid \mathbf{X}, S]$, the oracle gap is conditionally centred and reduces gradient covariance without changing the history-only gradient mean. To connect this claim to the implemented gap, write $Q_{\mathrm{out}} = m_Q(\mathbf{X}) + q^* + e$, where $m_Q = \mathbb{E}[Q_{\mathrm{out}} \mid \mathbf{X}]$ and e is centred approximation error. Proposition 4 shows how m_Q enters the mean update and how e changes the covariance; the fixed-checkpoint probe measures these two effects for the implemented output gap.

To assess the learned correction at a fixed checkpoint, we compare paired shared-body gradients g_s and g_t from student and teacher label losses. With $c = g_t - g_s$ and $\kappa = \beta/(1+\beta)$, their normalized combination is $g_s + \kappa c$. The covariance identity makes the intervention measurable: covariance with c records the stabilizing component, while the fixed history-only reference records mean alignment. Appendix C.1 derives both comparisons and specifies the isolated supervised paths.

At zero initialisation, teacher supervision first activates the information asymmetry. The views coincide and T_3 has zero gradient, while T_2 can update W_c through the descriptor. Subsequent updates can then use both learning routes. Section 4.3 tests content replacement and loss removal separately to connect teaching information with its update paths.

The gradient view identifies the update route. To determine the corresponding optimization preference, we profile a restricted supervised criterion for answer-conditioned residual reweighting. Consider the two supervised losses with $\alpha = \lambda = 0$, identity rescaling, $\beta > 0$, and a fixed student slot.

Stack example–channel pairs as rows of the student residual matrix $R \in \mathbb{R}^{n\times H}$. Let $S \in \mathbb{R}^{n\times d}$ contain the centred descriptors $(\mathbf{s}_j - \mathbf{s}_\varnothing)^\mathsf{T}$ and set $A = W_\mathrm{c}^\mathsf{T}$. The teacher residual is $R + SA$. As A varies, adjusting b_c to offset changes in $W_\mathrm{c}\mathbf{s}_\varnothing$ holds the student correction fixed; this algebra does not require a native host bias. Up to the common MSE factor, the joint criterion is $\mathcal{J}(R, A) = \|R\|_F^2 + \beta\|R + SA\|_F^2$.

Proposition 1. *Let P_S be the orthogonal projector onto the column space of S, I the identity matrix, and $\|\cdot\|_F$ the Frobenius norm. Then*

$$\min_A \mathcal{J}(R, A) = \|P_S R\|_F^2 + (1 + \beta)\|(I - P_S)R\|_F^2. \tag{7}$$

Equation (7) makes the teacher's role precise: it changes residual-direction weights for the shared deployment body, rather than defining a new inference target. Profiling gives unit weight to $P_S R$ and $1 + \beta$ to its orthogonal complement; $S = 0$ only rescales the loss. The summary is thus a training-time optimization scaffold operating through a d-dimensional summary subspace across the n example–channel pairs in each step (proof in Appendix C.3). The profile identifies this geometric intervention, while Proposition 3 specifies when a centred gap preserves the history-only risk target and Proposition 4 separates the implemented gap's mean shift from approximation error. EMA and residual matching extend this mechanism without changing history-only deployment.

3.5 Folding for inference

The student correction $v = W_\mathrm{c}\mathbf{s}_\varnothing + b_\mathrm{c}$ is independent of example and channel. In hosts with a compatible native output bias, it can merge into that bias and leave no teaching branch at inference.

Proposition 2. *Suppose the host body contains an output bias $b \in \mathbb{R}^H$ whose addition is broadcast across channels immediately before $N_\mathbf{X}$. At inference, the student in* (1) *is equivalent, in exact arithmetic, to the host with b replaced by $b + \mathrm{c}(\mathbf{s}_\varnothing)$.*

Proof. Write the host body as $u_\theta(\mathbf{X}) + b\mathbf{1}_D^\mathsf{T}$. The student's output is then

$$N_\mathbf{X}\left(u_\theta(\mathbf{X}) + (b + v)\mathbf{1}_D^\mathsf{T}\right),$$

which is the host forecast with the substituted bias and unchanged rescaling. □

The equality includes input-dependent scale restoration. A constant added after $N_\mathbf{X}$ would generally require a window-dependent bias change $v/a_\mathbf{X}$; placing it before restoration permits one fixed substitution. DLinear has no output rescaling, so either placement is equivalent; we add v to one of its two projection biases. The identity holds in evaluation mode with dropout inactive. The native bias used for each architecture is specified in Appendix B.

For compatible hosts, one-time export retains the host's layers and parameter shapes without the teacher, EMA model, hint extractor, or correction map. Every such trained student has an equivalent host through the substitution, and zero correction recovers every original host. The deployable function class is therefore unchanged: the additional training information acts on which predictor is learned within it. When no compatible bias exists, the fixed correction remains. Appendix E specifies each host's deployment: DLinear, FACT, and iTransformer fuse the correction into a native bias, whereas MixLinear retains it in both compared arms.

4 Experiments

The experiments test whether answer-derived teaching improves the history-only forecast, then examine answer content, shared update paths, and cross-host deployment.

4.1 Evaluation setup

The main evaluation uses two hosts with compatible native-bias folding: DLinear (Zeng et al., 2023) and iTransformer (Liu et al., 2024). We use lookback $L = 96$ and horizons $H \in \{96, 192, 336, 720\}$ for ETTh1 and ETTm2, and $H \in \{96, 720\}$ for Weather and Traffic. A *cell* fixes host, dataset, and horizon; a *configuration* also fixes learning rate and schedule. Data processing follows the benchmark pipeline (Wu et al., 2023). The data splits and architecture settings appear in Appendix A.

The paired reference isolates teaching from added training capacity. It shares the student's constant slot, initial state, and matched training random streams, but optimises student MSE against $\mathbf{Y}$ alone;

Table 2: Five-seed ETT test results. Ref. is the supervised constant-slot student; ours is full FoldOPSD. Δ reports paired MSE changes as mean $\pm$ SEM. Bold/† denotes a reduction meeting the paired-test and sign-agreement criterion of Section 4. Noise and ds-mean are basic-OPSD controls; their matched true-hint comparison is in Table 9. Rates are reference-validation-selected (Table 8); none is constant and type1 halves the rate each epoch.

Host	H	MSE (ref.)	MSE (ours)	Δ (%)	noise (%)	ds-mean (%)	lr/schedule
ETTh1							
DLinear	96	0.3941	0.3819	**-3.07 ± 0.56**†	-0.00	-0.00	1e-3 / none
DLinear	192	0.4684	0.4397	**-6.00 ± 1.74**†	-0.14	+0.02	3e-3 / none
DLinear	336	0.4997	0.4745	**-5.04 ± 0.45**†	-0.03	+0.01	3e-3 / none
DLinear	720	0.5374	0.4789	**-10.85 ± 0.83**†	+0.10	-0.06	3e-3 / none
ETTm2							
DLinear	96	0.1847	0.1836	-0.56 ± 0.89	+0.02	+0.03	3e-4 / none
DLinear	192	0.2495	0.2473	-0.89 ± 0.40	-0.00	+0.05	1e-3 / none
DLinear	336	0.3185	0.3097	**-2.76 ± 0.58**†	+0.07	+0.04	1e-3 / none
DLinear	720	0.4398	0.4145	-5.59 ± 2.48	-0.05	-0.23	3e-3 / none
ETTh1							
iTransformer	96	0.3832	0.3807	**-0.65 ± 0.10**†	+0.00	+0.00	3e-4 / type1
iTransformer	192	0.4356	0.4273	**-1.90 ± 0.22**†	-0.00	+0.00	1e-4 / none
iTransformer	336	0.4762	0.4663	**-2.08 ± 0.41**†	-0.00	-0.28	1e-4 / none
iTransformer	720	0.4815	0.4660	**-3.21 ± 0.46**†	-0.00	+0.02	1e-4 / none
ETTm2							
iTransformer	96	0.1851	0.1805	**-2.51 ± 0.43**†	+0.03	-0.00	3e-4 / type1
iTransformer	192	0.2519	0.2498	**-0.82 ± 0.16**†	+0.00	+0.00	1e-5 / none
iTransformer	336	0.3134	0.3077	**-1.82 ± 0.09**†	-0.00	-0.00	3e-5 / none
iTransformer	720	0.4107	0.4077	**-0.73 ± 0.12**†	+0.00	-0.07	1e-5 / none

both arms therefore deploy within the same host architecture. Reference validation MSE selects the configuration using five-seed means over ETT candidates or two-seed means over three learning rates for Weather and Traffic. All arms then use that configuration with seeds 0–4, and each selects its own checkpoint by history-only student validation MSE. The future-conditioned teacher neither selects the checkpoint nor enters the reported test forecast (Appendix A.2).

For paired arm and reference test MSEs m_i^a and m_i^0, we report the five-seed mean and standard error of $\Delta_i^a = 100(m_i^a - m_i^0)/m_i^0$. Negative values favour the method. A dagger marks a cell that meets the two-sided paired t-criterion at 0.05 and agrees in direction for at least four seeds. Differences between these changes are expressed in percentage points (pp). The noise and ds-mean columns use basic OPSD; the full method adds the EMA target and bounded residual matching. Appendix A.4 defines the matched objectives.

4.2 Self-teaching improves the deployed forecast

Full FoldOPSD lowers five-seed mean test MSE against the matched reference in all 24 main cells. On ETT (Table 2), all eight iTransformer cells and five of eight DLinear cells meet the per-cell criterion. The largest paired reduction is 10.85% on DLinear/ETTh1 at $H = 720$.

All eight Weather/Traffic cells improve by 0.41–4.18% and meet the per-cell criterion (Table 3); Weather reaches 4.18% for DLinear at $H = 720$ and 3.74% for iTransformer at $H = 96$.

4.3 Answer context and its shared teaching paths

With data, seeds, rates, and schedules held fixed, basic OPSD replaces descriptors to test answer content; full-objective loss interventions test the shared update paths.

The current answer supplies useful teaching context. Matched descriptor replacements isolate example-specific answer content under basic OPSD. Five-seed paired means favour true hints over scale-matched noise and a dataset-level mean in all 16 ETT configurations (Figure 3; Table 9). Noise breaks patch correspondence; ds-mean removes cross-example variation. True-minus-noise changes range from -0.23 ± 0.06 to -7.02 ± 0.57 pp; mean reductions against noise and ds-mean are 2.20

Table 3: Weather/Traffic transfer. Conventions follow Table 2. Noise and ds-mean replace only the teacher descriptor under basic OPSD (Table 9). Two-seed mean reference validation selects rates; final paired results use five seeds per arm.

Host	H	MSE (ref.)	MSE (ours)	Δ (%)	noise (%)	ds-mean (%)	lr/schedule
Traffic							
DLinear	96	0.6493	0.6430	**-0.98** $\pm$ **0.03**†	+0.00	-0.00	3.0e-04 / none
DLinear	720	0.6467	0.6393	**-1.14** $\pm$ **0.02**†	-0.00	+0.01	3.0e-04 / none
Weather							
DLinear	96	0.1972	0.1919	**-2.69** $\pm$ **0.45**†	+0.22	+0.21	3.0e-04 / none
DLinear	720	0.3508	0.3361	**-4.18** $\pm$ **0.19**†	-0.13	-0.13	3.0e-04 / none
Traffic							
iTransformer	96	0.4062	0.4013	**-1.21** $\pm$ **0.15**†	-0.13	-0.16	3.0e-04 / none
iTransformer	720	0.4656	0.4637	**-0.41** $\pm$ **0.11**†	+0.02	+0.24	3.0e-04 / none
Weather							
iTransformer	96	0.1776	0.1709	**-3.74** $\pm$ **0.84**†	+0.38	+0.50	1.0e-04 / none
iTransformer	720	0.3567	0.3525	**-1.16** $\pm$ **0.06**†	-0.01	-0.07	3.0e-05 / none

Table 4: Matched contrasts across 16 constant-slot ETT configurations (five paired seeds). Entries average per-seed reductions $100(m^{\text{control}} - m^{\text{treatment}})/m^0$ from paired references.

Ingredient	Matched comparison	Mean MSE reduction (pp)	Improving
EMA target	no EMA $\rightarrow$ EMA (output-MSE KD)	$+0.76$	13/16
Privileged content	noise $\rightarrow$ true future hints	$+2.20$	16/16
Example-specific content	dataset mean $\rightarrow$ true future hints	$+2.17$	16/16
Residual distance	output MSE $\rightarrow$ bounded residuals	$+0.07$	12/16

Positive favours treatment. EMA holds output-MSE distillation fixed; both hint rows use basic OPSD; residual distance holds EMA fixed. Table 9 gives cell-level paired means and errors.

and 2.17 pp (Table 4). These results show that answer-dependent content helps train the history-only student in these ETT settings and can change the residual subspace in Proposition 1.

Example-specific alignment carries the teaching signal. A within-batch cyclic shift of real summaries in the related history-generated-slot interface preserves values and scale but breaks sample pairing; aligned hints lower mean test MSE by 0.79–4.64 pp in all four settings, with 4/5 or 5/5 seeds agreeing (Appendix A.5, Table 10).

Teacher supervision gives that context a shared update path. At zero initialisation the views coincide and residual matching has zero gradient; T_2 can first learn a hinted correction and transmit its error through (6). Removing T_2 ($\beta = 0$) while retaining EMA and residual matching increases test MSE in four of five tested $H = 96$ configurations, a 1.83 pp mean difference (Table 5). Detaching the teacher's body output retains correction learning but removes its direct body update, costing 2.59–5.36 pp in four of five configurations (Appendix C). Together with content replacements, these interventions identify the measured routes for ordinary-label and hint-dependent gradients.

The shared gradient records the teaching intervention. At DLinear/ETTm2 ($H = 96$), a validation-selected checkpoint directly exposes the shared-body update. Isolating the student- and teacher-label paths, the joint gradient lowers same-split reference MSE by 28.3% on train and 35.0% on validation, while across-batch variance falls by 33.4% and 33.9% (Appendix C.2).

EMA adds a separate averaged-student target. The hint-content advantage appears without EMA. Adding its historical EMA student target to basic OPSD lowers mean error in 13 of 16 ETT settings by 0.76 pp (Table 4). EMA is therefore a separate target-smoothing component, while the answer-content comparison already holds without it. The two comparisons answer different questions: the hint control changes the information available to the teacher, whereas the EMA arm changes the temporal reference used by the student. Keeping these roles separate makes the measured answer-content effect independent of target smoothing.

Table 5: Removing training signals at $H = 96$. Values are five-seed paired test-MSE changes (%) against each arm's supervised reference, reported as mean ± SEM; negative is better.

Host	Dataset and rate	Full	No B ($\beta = 0$)	No KD ($\lambda = 0$)
DLinear	ETTh1, 10^{-3}	-3.07 ± 0.56	-1.82 ± 0.38	-2.90 ± 0.55
DLinear	ETTm2, $3 \cdot 10^{-4}$	-0.56 ± 0.89	$+1.67 \pm 1.17$	-0.33 ± 0.98
iTransformer	ETTh1, $3 \cdot 10^{-4}$ (type1)	-0.65 ± 0.10	$+4.03 \pm 0.08$	-0.66 ± 0.10
iTransformer	ETTh1, 10^{-4}	-2.17 ± 0.25	-0.13 ± 0.20	-2.32 ± 0.25
iTransformer	ETTm2, $3 \cdot 10^{-4}$ (type1)	-2.51 ± 0.43	-3.57 ± 0.56	-2.25 ± 0.44

Residual matching refines the current-teacher target. With EMA fixed, bounded residual matching lowers mean error relative to output-MSE distillation in 12 of 16 ETT configurations, by 0.07 pp on average (Table 4). It preserves sensitivity to small residuals while bounding larger discrepancies, and the corresponding five-configuration $H = 96$ contrasts are shown in Table 5.

Together, the controls connect answer content to shared weights: the constant-slot reference fixes the deployed class, descriptor replacements isolate example-specific information, and teacher-loss interventions expose its route into the shared body. EMA and residual matching then refine the targets along this same training path, while the host matrix below maps its architectural reach and the folding check establishes deployment equivalence.

4.4 Cross-architecture evaluation and deployment

A separate host-specific study maps architectural reach on DLinear, FACT, iTransformer, and MixLinear (Wang et al., 2026; Ma et al., 2026): six datasets and four horizons yield 96 cells and 480 seed-level paired comparisons. DLinear, FACT, and iTransformer each improve five-seed mean MSE in 20/24 cells. Across all four hosts, macro paired changes are -2.34% MSE and -1.62% MAE, with improvements in 68 and 74 cells, respectively. The matrix also includes MixLinear as a matched horizon-wise correction case: it improves in 8/24 MSE cells and has $+0.78\%$ MSE and $+1.03\%$ MAE macro changes. Table 6 gives the host–dataset results; Appendix E gives this separate protocol.

Table 6: Five-seed paired MSE change (%) by host and dataset; (x/4) counts improving horizons.

Dataset	DLinear	FACT	iTransformer	MixLinear
ETTh1	-2.53 (4/4)	-3.57 (4/4)	-2.77 (4/4)	$+1.85$ (1/4)
ETTh2	-19.90 (4/4)	-1.48 (4/4)	-1.31 (4/4)	$+0.22$ (2/4)
ETTm1	$+0.58$ (0/4)	-5.14 (4/4)	-5.29 (4/4)	-1.18 (3/4)
ETTm2	-10.14 (4/4)	-1.73 (4/4)	-2.34 (4/4)	$+0.17$ (2/4)
Weather	-2.80 (4/4)	-2.03 (4/4)	-0.70 (3/4)	$+3.26$ (0/4)
Traffic	-0.70 (4/4)	$+0.93$ (0/4)	$+0.12$ (1/4)	$+0.35$ (0/4)

Across foldable hosts, all 36 ETTh1, ETTh2, and ETTm2 cells improve among the three bias-compatible hosts, along with 11/12 Weather and 8/12 ETTm1 cells; Traffic contributes 5/12. These counts locate where answer-conditioned training helps, while exact bias folding guarantees equivalent deployment for every compatible host.

Folding preserves the learned student in compatible hosts. Proposition 2 removes the correction branch when a compatible native bias exists. On DLinear/iTransformer at ETTm2, $H = 96$, the five-seed float32 maximum difference between affine-correction and precomputed-constant paths is 4.77×10^{-7} (Appendix B). Paired accuracy therefore compares predictors within the same deployed function class; MixLinear retains the correction in both arms.

Training overhead is reported in Appendix F; folding removes the training scaffold without changing inference operations or parameter shapes.

5 Conclusion

FoldOPSD turns observed futures into a training-time scaffold: answer- conditioned supervision reshapes residual geometry and gradient weighting in the shared forecasting body, while the constant-slot student remains history-only. Equation (7), the controls, and gradient evidence identify this update as the transfer path; compatible bias folding removes the teacher and auxiliary modules without changing inference. Across four datasets and five seeds, FoldOPSD improves all 24 settings. The broader principle is that privileged future context can improve optimization inside a fixed history-only function class, then be discarded once the update is learned.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring disclosure. We used generative AI tools only to edit the manuscript for readability and language polishing, which is a recommended-disclosure task. We reviewed all AI-assisted text and take responsibility for the final content of this work, including its text, claims, and artifacts.

REFERENCES

Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. RepVGG: Making VGG-style ConvNets great again. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1607–1616. PMLR, 2018. URL https://proceedings.mlr.press/v80/furlanello18a.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint*, 2015. Presented at the NIPS 2014 Deep Learning Workshop.

Yuki Ichihara, Naoto Iwase, Mohammad Atif Quamar, and Junpei Komiyama. Privileged solutions or context-induced teacher behavior? dissecting on-policy self-distillation. *arXiv preprint arXiv:2608.09228*, 2026. doi: 10.48550/arXiv.2608.09228.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI 2018)*, pages 876–885. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/313.pdf.

Rickard K. A. Karlsson, Martin Willbo, Zeshan M. Hussain, Rahul G. Krishnan, David Sontag, and Fredrik Johansson. Using time-series privileged information for provably efficient learning of prediction models. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 5459–5484. PMLR, 2022. URL https://proceedings.mlr.press/v151/k-a-karlsson22a.html.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations (ICLR)*, 2022.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Learning using privileged information: SVM+ and weighted SVM. *Neural Networks*, 53:95–108, 2014. doi: 10.1016/j.neunet.2014.02.002. URL https://arxiv.org/abs/1306.3161.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30211–30226. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24n.html.

Chenxi Liu, Hao Miao, Qianxiong Xu, Shaowen Zhou, Cheng Long, Zhao Yan, Ziyue Li, and Rui Zhao. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. In *IEEE International Conference on Data Engineering (ICDE)*, 2025. URL https://arxiv.org/abs/2505.02138.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

Aitian Ma, Dongsheng Luo, and Mo Sha. MixLinear: Extreme low-resource multivariate time series forecasting with 0.1k parameters. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2410.02081.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3351–3361. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/2288f691b58edecadcc9a8691762b4fd-Paper.pdf.

Juntong Ni, Zewen Liu, Shiyu Wang, Ming Jin, and Wei Jin. TimeDistill: Efficient long-term time series forecasting with MLP via cross-architecture distillation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pages 1113–1124, 2026. doi: 10.1145/3770854.3780218. URL https://dl.acm.org/doi/10.1145/3770854.3780218.

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

Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5142–5151. PMLR, 2019. URL https://proceedings.mlr.press/v97/phuong19a.html.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16, 2015.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009.

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. MobileOne: An improved one millisecond mobile backbone. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7907–7917, 2023.

Huiqiang Wang, Jieming Shi, and Qing Li. FACT: Fine-grained across-variable convolution for multivariate time series forecasting. In *International Conference on Learning Representations*, pages 4967–4996, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/090d636f951ac0bb18fd5610c95c6a76-Paper-Conference.pdf.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2b187165e28fdfdc0ffb34d1bfff2b0c-Paper-Conference.pdf.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations (ICLR)*, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3713–3722, October 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Zhang_Be_Your_Own_Teacher_ICCV_2019_paper.html.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4328, June 2018. URL

https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Deep_Mutual_Learning_CVPR_2018_paper.html.
Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. doi: 10.48550/arXiv.2601.18734.
Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

A Implementation and evaluation protocol

The observed future serves as the target for both views and as the source of the teacher's answer hints. This section specifies the data protocol, hint construction, and matched comparisons. For the compatible hosts in the main study, the trained student's correction folds into the native bias without changing the inference architecture.

A.1 Data and host implementations

All tasks are multivariate: the model predicts every observed channel. ETTh1 and ETTm2 (Zhou et al., 2021) contain seven transformer load-and-temperature variables, sampled hourly and every 15 minutes, respectively. Weather (Wu et al., 2021) has 21 variables sampled every 10 minutes; Traffic (Wu et al., 2021) has 862 hourly channels. The benchmark data loader uses chronological splits and fits channel-wise standardisation on the training split. The ETT split uses 12 months for training, four for validation, and four for testing; the custom datasets use 70%, 10%, and 20%. The history of a validation or test example may include observations before its split boundary, but its forecast target lies within the designated split. MSE and mean absolute error (MAE) are measured in the dataset-standardised units, after reversing any additional per-window normalisation inside the host.

The DLinear body uses a length-25 moving average with replicated edge padding. It sums linear projections of the moving-average trend and the remaining seasonal component, with weights shared across channels. Projection weights start at $1/L$ and biases at zero.

The iTransformer body embeds each channel's L observations into one token. It uses multi-head channel attention, residual connections, layer normalisation, and a two-layer feed-forward block with a rectified linear unit (ReLU) between the layers. ReLU maps a scalar z to $\max(0, z)$. The implementation's feed-forward width is twice its token width, and it does not embed calendar covariates. On ETT it uses width 128 and two blocks. On Weather it uses width 512 and three blocks; on Traffic it uses width 512 and four blocks. All use eight heads and dropout probability 0.1. The reference and every FoldOPSD variant in a comparison share one host implementation and one set of these settings throughout.

Training settings. The optimizer is Adam, with gradient norms clipped at 5. DLinear and iTransformer train for ten epochs; their batch size is 32 except for iTransformer on Traffic, where it is 16. Each arm restores the checkpoint with minimum student-view validation MSE within its epoch budget. We use either a fixed learning rate (*none*) or the *type1* schedule, which multiplies the initial rate by 0.5^e at zero-based epoch e. Both schedules retain the same initialisation and matched training random states across arms. Unless an arm is named as an intervention, FoldOPSD uses the default weights of Section 3.3 ($\alpha = 0.5$, $\rho = 0.999$, $\beta = 1$, $\lambda = 0.04$, $\tau = 10$), at the configuration chosen by the selection rule in (8).

The reference is the constant-slot student trained with $\ell(\hat{\mathbf{Y}}_s, \mathbf{Y})$, the student loss alone. At the zero initialisation, W_c and $\mathbf{s}_\varnothing$ have zero gradients under this objective and remain zero, but b_c receives a gradient. Keeping b_c trainable therefore gives the reference the same additive bias parameterisation as FoldOPSD. On the main-study hosts, this constant can be absorbed into the existing host bias. The comparison holds both training parameterisation and deployed function class fixed, and changes only the training objective.

Table 7: Candidate learning rates and schedules for the ETT main table. An unmarked rate uses a constant schedule; a superscript h denotes *type1*, which halves the rate each epoch. Every listed candidate has five seeds for the reference, FoldOPSD, noise, and ds-mean arms. The rates shown in the main table minimise mean reference validation MSE.

Host / dataset	H	Candidate configurations
DLinear / ETTh1	96	10^{-3}, $3 \cdot 10^{-3}$, $(3 \cdot 10^{-3})^h$
	192, 720	10^{-3}, $3 \cdot 10^{-3}$
	336	10^{-3}, $(10^{-3})^h$, $3 \cdot 10^{-3}$
DLinear / ETTm2	96	$3 \cdot 10^{-4}$, 10^{-3}, $(10^{-3})^h$
	192, 336	10^{-3}
	720	10^{-3}, $3 \cdot 10^{-3}$
iTransformer / ETTh1	96	10^{-4}, $(3 \cdot 10^{-4})^h$, 10^{-3}
	192	10^{-4}, $(10^{-3})^h$, $3 \cdot 10^{-3}$
	336	$3 \cdot 10^{-5}$, 10^{-4}, $3 \cdot 10^{-3}$
	720	$3 \cdot 10^{-5}$, 10^{-4}, $(10^{-4})^h$
iTransformer / ETTm2	96	$3 \cdot 10^{-4}$, $(3 \cdot 10^{-4})^h$, 10^{-3}
	192, 336, 720	10^{-5}, $3 \cdot 10^{-5}$, 10^{-3}

A.2 Multiseed configuration selection

For candidate configuration c, let $v_{i,c}$ be the reference's minimum validation MSE over epochs for seed i. We summarise each candidate by its mean and sample variance over the selection seeds,

$$\bar{v}_c = \frac{1}{n_v} \sum_{i=1}^{n_v} v_{i,c}, \qquad s^2_{v,c} = \frac{1}{n_v - 1} \sum_{i=1}^{n_v} (v_{i,c} - \bar{v}_c)^2, \qquad c^\star = \arg\min_c \bar{v}_c. \tag{8}$$

Thus the mean determines the configuration and the sample variance describes seed variability. All configurations in a comparison use the same selection seed set, and the selected configuration is then used by every arm in that row. Exact ties in mean validation MSE are resolved by configuration label. For ETT, $n_v = 5$ with seeds 0–4. Table 7 lists the candidate rates and schedules.

For the transfer study, calibration uses seeds 0 and 1 and three rates: $0.3r$, r, and $3r$, without a schedule. The centre rate r is 10^{-4} for DLinear on the reported transfer datasets. For iTransformer it is 10^{-4} on Weather and 10^{-3} on Traffic. Every comparison arm then uses the selected rate and seeds 0–4. Table 8 reports the selection statistics and variance of the paired test changes.

For test changes Δ_i, we compute $s^2_\Delta = \sum_i (\Delta_i - \bar{\Delta})^2/(n-1)$ and display $\bar{\Delta} \pm \sqrt{s^2_\Delta/n}$ in the main tables, so those entries are standard errors over five paired seeds, while Table 8 reports sample variances and the extended host tables report sample standard deviations. With five seeds, the two-sided 0.05 critical value is $|t| = 2.776$. At least four seeds must also agree on the direction of change to satisfy the sign criterion.

A.3 Exact construction of the answer hints

The answer-hint descriptor is a vector of numerical hints, not a replacement trajectory for student supervision. Each patch contributes local level, spread, and trend in the order in which that patch occurs in the future. For lookback channel j, define its mean μ_j and population standard deviation σ_j. First set $q_j = \sigma_j + 10^{-5}$, then use

$$\tilde{\sigma}_j = \max\left(q_j,\, 0.01 \max\left(D^{-1} \sum_{k=1}^{D} q_k, 10^{-6}\right)\right).$$

This limits amplification on nearly constant channels. For patch m, let $a_{m,j}$ and $u_{m,j}$ be its mean and population standard deviation. With within-patch indices $t = 0, \ldots, P-1$ and $\bar{t} = (P-1)/2$, its least-squares slope over that patch is

$$b_{m,j} = \frac{\sum_{t=0}^{P-1} (t - \bar{t}) Y_{mP+t,j}}{\sum_{t=0}^{P-1} (t - \bar{t})^2 + 10^{-12}}.$$

Here patch and time indices are zero-based. The descriptor concatenates

$$\left[\frac{a_{m,j} - \mu_j}{\tilde{\sigma}_j}, \quad \frac{u_{m,j}}{\tilde{\sigma}_j}, \quad \frac{b_{m,j}}{\tilde{\sigma}_j}\right]_{m=0}^{M-1}$$

Table 8: Multiseed validation statistics for the selected configurations. n_v is the number of seeds used for configuration selection: five on ETT and two in the transfer calibration. $\bar{v}$ and s_v^2 are the mean and unbiased sample variance of the reference's best validation MSE. The final column gives the sample variance of the paired test changes Δ in squared percentage points; all test results use five seeds. Configuration selection minimises $\bar{v}$; the reported variances are descriptive.

Host	Dataset	H	n_v	$\bar{v}$	s_v^2	s_Δ^2 (pp^2)
DLinear	ETTh1	96	5	0.646564	1.43×10^{-5}	1.5417
DLinear	ETTh1	192	5	0.875882	3.25×10^{-4}	15.2214
DLinear	ETTh1	336	5	1.081832	1.06×10^{-4}	0.9909
DLinear	ETTh1	720	5	1.225032	2.19×10^{-4}	3.4108
DLinear	ETTm2	96	5	0.128440	3.09×10^{-6}	3.9323
DLinear	ETTm2	192	5	0.171764	1.08×10^{-6}	0.7829
DLinear	ETTm2	336	5	0.222326	4.48×10^{-6}	1.6847
DLinear	ETTm2	720	5	0.305376	8.26×10^{-5}	30.8592
DLinear	Traffic	96	2	0.547940	2.31×10^{-7}	0.0045
DLinear	Traffic	720	2	0.545090	5.00×10^{-9}	0.0013
DLinear	Weather	96	2	0.461265	8.45×10^{-9}	1.0303
DLinear	Weather	720	2	0.652550	1.25×10^{-7}	0.1840
iTransformer	ETTh1	96	5	0.687214	5.81×10^{-6}	0.0490
iTransformer	ETTh1	192	5	0.995652	7.17×10^{-6}	0.2460
iTransformer	ETTh1	336	5	1.291730	9.87×10^{-6}	0.8407
iTransformer	ETTh1	720	5	1.571142	2.50×10^{-5}	1.0414
iTransformer	ETTm2	96	5	0.133400	4.94×10^{-6}	0.9325
iTransformer	ETTm2	192	5	0.178414	3.40×10^{-7}	0.1220
iTransformer	ETTm2	336	5	0.224998	3.36×10^{-7}	0.0400
iTransformer	ETTm2	720	5	0.295016	9.89×10^{-8}	0.0666
iTransformer	Traffic	96	2	0.369520	5.62×10^{-7}	0.1116
iTransformer	Traffic	720	2	0.422705	1.23×10^{-6}	0.0627
iTransformer	Weather	96	2	0.423275	2.96×10^{-7}	3.5074
iTransformer	Weather	720	2	0.710060	4.32×10^{-6}	0.0183

and clips each entry to $[-10, 10]$. The slope measures change per time step, scaled by lookback variability. At $P = 24$, the descriptor dimensions for $H = 96, 192, 336, 720$ are $12, 24, 42, 90$; at $P = 48$, they are $6, 12, 21, 45$. No future-patch statistic uses samples from another future patch; the normalisation above also uses lookback statistics.

A.4 Matched summary and objective comparisons

We first compare answer hints with replacement descriptors under a fixed two-view objective, then examine the student target and distillation distance. Let q denote the teacher descriptor and $\hat{\mathbf{Y}}_t(q)$ its forecast. Basic OPSD and both summary controls use

$$\mathcal{L}_{\text{basic}}(q) = \ell(\hat{\mathbf{Y}}_s, \mathbf{Y}) + \beta\ell(\hat{\mathbf{Y}}_t(q), \mathbf{Y}) + \lambda\ell(\hat{\mathbf{Y}}_s, \text{sg}[\hat{\mathbf{Y}}_t(q)]). \tag{9}$$

Basic OPSD supplies the true descriptor $\mathbf{s}(\mathbf{Y}; \mathbf{X})$. Noise and ds-mean replace that descriptor while retaining the objective, loss weights, and correction parameterisation. Their differences compare summary inputs at the same point in the method, before either an EMA student target or a bounded residual-matching loss enters the overall training objective.

For noise, $q = \hat{\sigma}_{\text{batch}}\epsilon$, with independent standard Gaussian entries in ϵ and $\hat{\sigma}_{\text{batch}}$ equal to the standard deviation over all entries of the current batch's true descriptors. This retains the batch's overall descriptor scale while removing the alignment between each entry and its future patch. For ds-mean, the implementation averages descriptors over examples in each training batch, then averages those batch means. The resulting per-channel vector retains a dataset-level descriptor while removing any variation across the training examples themselves.

Two further arms retain the true descriptor and connect basic OPSD to the complete objective. The EMA variant replaces the first term of (9) with T_1, keeping the teacher loss and output-MSE distillation unchanged. Full FoldOPSD then replaces output MSE with T_3. Summary replacement, EMA, and distillation distance thus each have a comparison in which the other objective choices are held fixed. The controls remain at the basic-OPSD stage to test summary content there; the later arms measure target smoothing and residual matching.

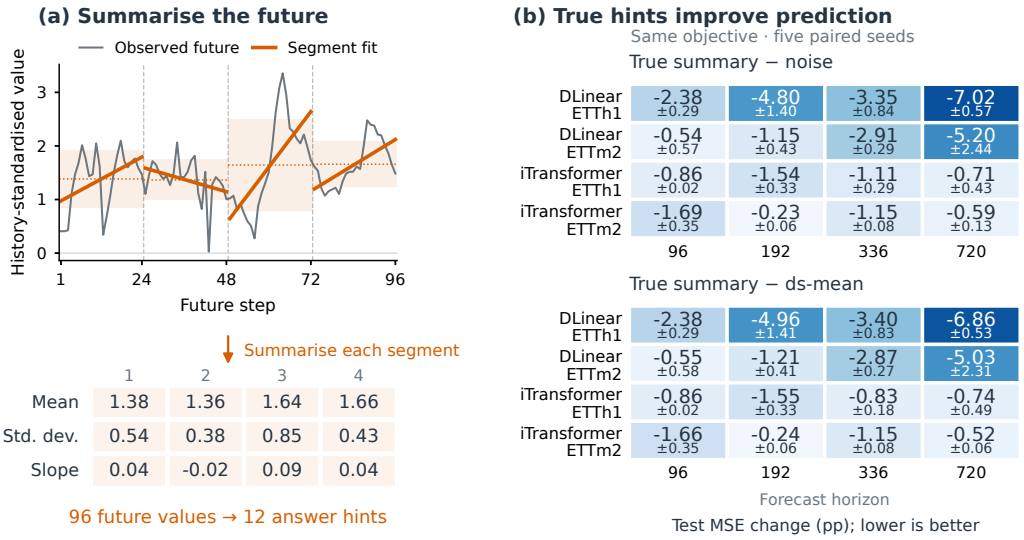

Figure 3: Structured answer hints from future segments. **(a)** Four 24-step ETTh1 segments yield 12 statistics: mean, standard deviation, and slope. Orange lines are least-squares fits and bands are mean $\pm$ standard deviation of observed data, not forecasts. **(b)** True versus replacement summaries under the same basic-OPSD objective. Values are five-seed mean $\pm$ standard error of $100(m_i^{\text{basic}} - m_i^{\text{control}})/m_i^0$, with m_i^0 the paired reference test MSE. All 16 mean differences favour true hints. Configurations match Table 2; noise preserves batch descriptor scale and ds-mean supplies a dataset-level summary. Appendix A.4 specifies the matched objectives.

Table 9: Matched comparisons at the 16 validation-selected ETT configurations in Table 2. The first two contrasts change the teacher descriptor under basic OPSD. The next two add EMA and replace output MSE with bounded residual matching, respectively. Each entry is the mean $\pm$ standard error of $100(m_i^a - m_i^b)/m_i^0$ over five paired seeds, in percentage points; negative values favour the first arm. All contrasts use the same supervised reference denominator.

Host	Dataset	H	Basic−noise	Basic−ds-mean	EMA−basic	Full−EMA
DLinear	ETTh1	96	-2.38 ± 0.29	-2.38 ± 0.29	-0.54 ± 0.33	-0.15 ± 0.04
DLinear	ETTh1	192	-4.80 ± 1.40	-4.96 ± 1.41	-0.88 ± 0.35	-0.18 ± 0.06
DLinear	ETTh1	336	-3.35 ± 0.84	-3.40 ± 0.83	-1.62 ± 0.50	-0.04 ± 0.05
DLinear	ETTh1	720	-7.02 ± 0.57	-6.86 ± 0.53	-3.88 ± 0.52	-0.05 ± 0.18
DLinear	ETTm2	96	-0.54 ± 0.57	-0.55 ± 0.58	$+0.21 \pm 0.46$	-0.25 ± 0.10
DLinear	ETTm2	192	-1.15 ± 0.43	-1.21 ± 0.41	$+0.35 \pm 0.33$	-0.09 ± 0.17
DLinear	ETTm2	336	-2.91 ± 0.29	-2.87 ± 0.27	-0.25 ± 0.33	$+0.32 \pm 0.05$
DLinear	ETTm2	720	-5.20 ± 2.44	-5.03 ± 2.31	-0.08 ± 0.80	-0.25 ± 0.24
iTransformer	ETTh1	96	-0.86 ± 0.02	-0.86 ± 0.02	$+0.18 \pm 0.12$	$+0.02 \pm 0.01$
iTransformer	ETTh1	192	-1.54 ± 0.33	-1.55 ± 0.33	-0.30 ± 0.19	-0.06 ± 0.02
iTransformer	ETTh1	336	-1.11 ± 0.29	-0.83 ± 0.18	-0.79 ± 0.18	-0.18 ± 0.01
iTransformer	ETTh1	720	-0.71 ± 0.43	-0.74 ± 0.49	-2.59 ± 0.31	$+0.10 \pm 0.21$
iTransformer	ETTm2	96	-1.69 ± 0.35	-1.66 ± 0.35	-0.68 ± 0.21	-0.18 ± 0.04
iTransformer	ETTm2	192	-0.23 ± 0.06	-0.24 ± 0.06	-0.62 ± 0.21	$+0.03 \pm 0.00$
iTransformer	ETTm2	336	-1.15 ± 0.08	-1.15 ± 0.08	-0.55 ± 0.06	-0.12 ± 0.01
iTransformer	ETTm2	720	-0.59 ± 0.13	-0.52 ± 0.06	-0.06 ± 0.07	-0.08 ± 0.04

Table 9 gives these comparisons at all 16 ETT configurations selected for the main table. For two arms a and b, the contrast is $100(m_i^a - m_i^b)/m_i^0$, using the same reference denominator m_i^0. This expresses the changes in comparable percentage-point units. Basic OPSD improves in mean over both replacement summaries in all 16 configurations. The EMA and distance columns then show how the complete method differs from that matched content comparison, so each entry is a conditional addition at a specified point in the objective.

A.5 Sample alignment in a learned-slot interface

The matched replacement controls above establish that true summaries carry useful example-dependent content. A complementary permutation test asks whether that content must stay attached to the future that generated it. This study uses a related interface whose student generates a slot from history through a one-hidden-layer ReLU network of width 32, rather than FoldOPSD's learned constant. The teacher uses the same affine future- summary interface. Its objective combines student MSE, teacher MSE with $\beta = 1$, and output-MSE distillation with $\lambda = 0.04$, without EMA or bounded residual matching. The generated slot remains input-dependent at inference, so this study isolates sample alignment in the teaching signal.

The four configurations are DLinear/ETTm2 at $H = 96$ and 720, iTransformer/ETTm2 at $H = 96$, and DLinear/Traffic at $H = 96$, with rates 10^{-3}, $3 \cdot 10^{-3}$, $3 \cdot 10^{-4}$, and $3 \cdot 10^{-4}$, respectively; only iTransformer uses the type1 schedule. All use lookback 96, batch size 32, ten epochs, and seeds 0–4. For batches larger than one, an independent random stream draws a nonzero cyclic shift of the complete descriptors, assigning each real summary to a different example while preserving the batch's descriptor values. The aligned and permuted arms otherwise share their configurations.

Table 10: Sample-alignment control in the related history-generated-slot interface. The permuted arm cyclically shifts real summaries within each batch. Entries are five-seed mean ± standard error of $100(m_i^{\text{full}} - m_i^{\text{control}})/m_i^{\text{host}}$ in pp; negative values favour aligned summaries. KD-off removes output distillation; auxiliary directly supervises the generated summary at its validation-selected weight. Signs count seeds favouring aligned over permuted summaries.

Host	Dataset	H	Full−permuted	Full−KD-off	Full−auxiliary	Signs
DLinear	ETTm2	96	-0.79 ± 0.54	$+0.03 \pm 0.03$	$+0.26 \pm 0.23$	4/5
DLinear	ETTm2	720	-4.64 ± 2.32	-0.02 ± 0.18	$+0.91 \pm 0.57$	4/5
iTransformer	ETTm2	96	-2.89 ± 0.48	$+0.10 \pm 0.02$	-0.01 ± 0.04	5/5
DLinear	Traffic	96	-1.46 ± 0.08	$+0.06 \pm 0.00$	$+0.34 \pm 0.12$	5/5

Across all four settings, the aligned arm has lower five-seed mean test MSE than its permuted counterpart. The paired change ranges from -0.79 to -4.64 percentage points, with 4/5 or 5/5 seeds favouring alignment. This identifies sample alignment as part of the answer-conditioned teaching signal and complements the constant-slot ablation in Table 9.

B Bias folding and numerical checks

For hosts with a compatible native output bias, deployment retains the student's learned forecast without computing the correction branch. For a trained student, compute $v = W_c \mathbf{s}_\varnothing + b_c$ once. In DLinear, add v to the seasonal projection bias `lin_s.bias`; updating the trend projection alone is equivalent. In iTransformer, update `proj.bias`. Remove the correction map and constant slot after this update. The trained affine parameters and lookback statistics of reversible normalisation remain unchanged during this bias substitution.

The numerical check evaluates

$$N_{\mathbf{X}}\big(f_\theta(\mathbf{X}) + (W_c \mathbf{s}_\varnothing + b_c)\mathbf{1}_D^\top\big) \quad \text{against} \quad N_{\mathbf{X}}\big(f_\theta(\mathbf{X}) + v\mathbf{1}_D^\top\big)\,.$$

Substituting $b + v$ for the host bias gives the deployment expression in Proposition 2. Table 11 uses five seeds per host from the output-correction implementation at $H = 96$ on ETTm2. The largest absolute prediction difference in these runs is 4.77×10^{-7}. Zero sample variance in some rows means that the measured maxima coincide across seeds. The one-time bias substitution removes the external correction addition: inference needs neither a descriptor nor a per-prediction folding step.

C Training-objective analysis

C.1 Ideal conditional-gradient reduction and checkpoint criterion

The ideal calculation identifies when answer-aware supervision can reduce the noise in a history-only learning signal without changing its conditional mean. Let $S = \phi(\mathbf{X}, \mathbf{Y})$ be any answer summary, and define $q^* = \mathbb{E}[\varepsilon \mid \mathbf{X}, S]$ and $\eta = \varepsilon - q^*$ for $\mathbf{Y} = \mu(\mathbf{X}) + \varepsilon$, where $\mathbb{E}[\varepsilon \mid \mathbf{X}] = 0$. Then $\mathbb{E}[q^* \mid \mathbf{X}] = 0$ and $\mathbb{E}[\eta \mid \mathbf{X}, S] = 0$. In particular, $\mathbb{E}[\eta (q^*)^\top \mid \mathbf{X}] = 0$ by iterated expectation.

Table 11: Multiseed check of constant-correction evaluation on ETTm2, $H = 96$, using the main FoldOPSD arm at a configuration selected by mean reference validation MSE. For each seed, e_i is the largest absolute prediction difference between the affine-correction student and the constant-evaluation path. The table reports the mean, sample variance, and maximum of e_i. Both paths evaluate the same trained student, the second precomputing its constant correction.

Host	Seeds	Mean e_i	Variance e_i	Max e_i
DLinear	5	2.98×10^{-7}	3.20×10^{-14}	4.77×10^{-7}
iTransformer	5	4.77×10^{-7}	0	4.77×10^{-7}

Proposition 3 (Ideal conditional-gradient reduction)**.** *Fix a history-only student forecast $\hat{\mathbf{Y}}_s$ and let the teacher forecast be $\hat{\mathbf{Y}}_s + q^*$. For half-squared losses and any $\beta \geq 0$, write $\delta = \hat{\mathbf{Y}}_s - \mu(\mathbf{X})$. The student output gradient u_s and the normalized student-plus-teacher output gradient u_β have the same conditional mean, and*

$$\mathbb{E}[u_s \mid \mathbf{X}] = \mathbb{E}[u_\beta \mid \mathbf{X}] = \delta,$$
$$\mathrm{Cov}(u_s \mid \mathbf{X}) - \mathrm{Cov}(u_\beta \mid \mathbf{X}) = \left[1 - (1+\beta)^{-2}\right] \mathrm{Cov}(q^* \mid \mathbf{X}) \succeq 0. \tag{10}$$

For a fixed shared-body Jacobian $J_\theta(\mathbf{X}) = \partial \hat{\mathbf{Y}}_s / \partial \theta$, the same positive-semidefinite covariance ordering holds for the parameter gradients.

Proof. Since $\mathbf{Y} = \mu(\mathbf{X}) + q^* + \eta$, the two output gradients are

$$u_s = \hat{\mathbf{Y}}_s - \mathbf{Y} = \delta - q^* - \eta,$$
$$u_\beta = \frac{(\hat{\mathbf{Y}}_s - \mathbf{Y}) + \beta(\hat{\mathbf{Y}}_s + q^* - \mathbf{Y})}{1+\beta} = \delta - \eta - \frac{q^*}{1+\beta}.$$

Both q^* and η have conditional mean zero given $\mathbf{X}$. Their conditional cross-covariance is zero because $\mathbb{E}[\eta \mid \mathbf{X}, S] = 0$. The displayed covariance difference therefore follows by expanding each covariance. For fixed $\mathbf{X}$, $J_\theta(\mathbf{X})$ is deterministic, and transforming an output gradient u into $J_\theta(\mathbf{X})^\mathsf{T} u$ transforms its covariance by $J_\theta(\mathbf{X})^\mathsf{T}(\cdot)J_\theta(\mathbf{X})$, preserving the positive-semidefinite ordering. □

The preceding proposition uses the oracle gap. The implemented model has a more general gap, and its relationship to the oracle can be stated exactly. The following result connects the proof to the output-space affine slot rather than treating that slot as an oracle.

Proposition 4 (Centered approximation for the implemented gap)**.** *Let $Q_{\mathrm{out}} = a_{\mathbf{X}} \odot W_{\mathrm{c}}(S - \mathbf{s}_\varnothing)$ be the actual teacher–student output gap, with the scale factor omitted when the host is not normalised. Define*

$$m_Q(\mathbf{X}) = \mathbb{E}[Q_{\mathrm{out}} \mid \mathbf{X}], \qquad Q_\circ = Q_{\mathrm{out}} - m_Q(\mathbf{X}), \qquad e = Q_\circ - q^*,$$

and set $\kappa = \beta/(1+\beta)$. For the fixed history-only forecast in Proposition 3, the normalised label-supervised output gradient produced by the actual gap is

$$u_Q = \delta - \eta - (1-\kappa)q^* + \kappa e + \kappa m_Q(\mathbf{X}), \tag{11}$$

and hence

$$\mathbb{E}[u_Q \mid \mathbf{X}] = \delta + \kappa m_Q(\mathbf{X}). \tag{12}$$

Writing $C_{a,b} = \mathrm{Cov}(a, b \mid \mathbf{X})$, its conditional covariance satisfies

$$\begin{aligned}\mathrm{Cov}(u_s \mid \mathbf{X}) - \mathrm{Cov}(u_Q \mid \mathbf{X}) = {} & [1-(1-\kappa)^2]C_{q^*,q^*} - \kappa^2 C_{e,e} \\ & + \kappa(1-\kappa)(C_{q^*,e} + C_{e,q^*}) + \kappa(C_{\eta,e} + C_{e,\eta}).\end{aligned} \tag{13}$$

For a fixed shared-body Jacobian, the same identities hold after left and right multiplication by that Jacobian. In particular, if $m_Q(\mathbf{X}) = 0$ and $e = 0$, then $Q_{\mathrm{out}} = q^*$ and (13) *reduces to* (10)*. More generally, the oracle covariance reduction is retained precisely when the error terms on the second line do not overwhelm the first term, while m_Q accounts separately for a mean shift.*

Proof. The joint output gradient is

$$u_Q = \frac{u_s + \beta(u_s + Q_{\mathrm{out}})}{1+\beta} = u_s + \kappa Q_{\mathrm{out}}.$$

Using $u_s = \delta - q^* - \eta$ and $Q_{\text{out}} = q^* + e + m_Q(\mathbf{X})$ gives (11). Since q^*, η, and e are conditionally centred, its conditional mean is (12). Moreover, $\mathbb{E}[\eta(q^*)^\mathsf{T} \mid \mathbf{X}] = 0$ by the definition of q^*. Expanding the covariance of $-\eta - (1-\kappa)q^* + \kappa e$ and subtracting the covariance of $-\eta - q^*$ yields (13). A fixed history-conditioned Jacobian is deterministic under the conditioning, so the parameter-gradient statement follows by congruence transformation. □

The centered-approximation proposition therefore characterizes the learned constant slot through two quantities: the realizable target component q^* and the approximation error e. Conditional centring controls the mean update, while e controls the residual covariance, making the connection between the implemented gap and the ideal gradient criterion explicit.

At one fixed checkpoint, let $g_s = \nabla_\theta T_{1,\text{label}}$ and $g_t = \nabla_\theta T_2$ be the shared-body gradients from the same minibatch, with $\alpha = \lambda = 0$. Set $c = g_t - g_s$ and $g_{\text{joint}} = g_s + \kappa c$. Across the diagnostic minibatch distribution,

$$\operatorname{trVar}(g_{\text{joint}}) - \operatorname{trVar}(g_s) = 2\kappa \operatorname{trCov}(g_s, c) + \kappa^2 \operatorname{trVar}(c). \tag{14}$$

This identity separates the covariance benefit from the variance cost of the teacher correction. To account for a shift in the mean, let g_{ref} be a fixed history-only gradient computed on a large block disjoint from the diagnostic batches, and define $\text{MSE}_{\text{ref}}(g) = \mathbb{E}\|g - g_{\text{ref}}\|_2^2$. Then

$$\begin{aligned}
\Delta \,\text{MSE}_{\text{ref}} &= \text{MSE}_{\text{ref}}(g_{\text{joint}}) - \text{MSE}_{\text{ref}}(g_s) \\
&= \underbrace{\operatorname{trVar}(g_{\text{joint}}) - \operatorname{trVar}(g_s)}_{\text{variance change}} + \underbrace{\|\mathbb{E}[g_s] + \kappa\mathbb{E}[c] - g_{\text{ref}}\|_2^2 - \|\mathbb{E}[g_s] - g_{\text{ref}}\|_2^2}_{\text{mean-error change}} \\
&= 2\kappa\mathbb{E}\langle g_s - g_{\text{ref}}, c\rangle + \kappa^2\mathbb{E}\|c\|_2^2.
\end{aligned} \tag{15}$$

A negative value indicates that the joint gradient is closer to the empirical history-only update represented by this reference.

C.2 Fixed-checkpoint gradient probe

The paired probe measures the learned correction's effect on empirical gradient error at a history-only predictor. We use the validation-selected DLinear/ETTm2 checkpoint at $H = 96$, seed 0, with a learned constant student slot. The checkpoint was trained with $\alpha = 0.5$, $\beta = 1$, $\lambda = 0.04$, and EMA decay 0.999; the probe holds its parameters fixed and isolates the two label-supervised paths by setting $\alpha = \lambda = 0$ only in the gradient calculation.

Each split supplies eight temporally separated paired batches of 32 windows. The fixed history-only reference averages student-label gradients over 31,439 training or 8,495 validation windows whose raw observations do not overlap the diagnostic batches in that split. All gradients use the 18,624 parameters of the shared DLinear body, and the joint gradient is divided by $1 + \beta$ before comparison. Table 12 reports mean squared distance to a same-split reference on train and validation windows. The probe uses these training and validation windows, while held-out test windows provide the final forecasting evaluation; the percentage change is $100(\text{Joint}/\text{Student} - 1)$.

Table 12: Fixed-checkpoint shared-body gradient error on DLinear/ETTm2 ($H = 96$). Each probe split is compared with a disjoint large-block history-only reference from the same split; lower MSE is better.

Probe batches	Reference	Student MSE	Joint MSE	Δ (%)
Train	Train	0.24529	0.17597	−28.3
Validation	Validation	0.04791	0.03113	−35.0

The learned teacher correction reduces empirical between-batch variance in both splits. Equation (15) decomposes reference error into variance and mean-alignment terms. For training batches, $\operatorname{trCov}(g_s, c) = -0.05079$ and $\operatorname{trVar}(c) = 0.02319$; for validation batches the terms are -0.01983 and 0.02360. With $\kappa = 0.5$, these values explain the observed variance-trace changes through (14). The same-split references make the probe report both within-split stabilization and mean alignment, complementing the conditional-gradient result in Proposition 3.

C.3 Proof of the answer-conditioned residual weighting

Proposition 1 characterises how a shared-weight, answer-conditioned teacher changes the fixed student's residual criterion. This direction-wise view complements privileged-feature weighting (Lapin et al., 2014), analyses of soft-label training geometry (Phuong and Lampert, 2019), and basis-function analyses of self-distillation (Mobahi et al., 2020). Consider $\alpha = \lambda = 0$, $N_{\mathbf{X}}$ equal to the identity, $\beta > 0$, and a fixed student slot $\mathbf{s}_\varnothing$. These assumptions isolate the two supervised losses and allow us to solve for the privileged correction at a fixed folded student.

Stack all training example–channel pairs as n rows. Let $R \in \mathbb{R}^{n\times H}$ be the folded student's residual matrix, and let $S \in \mathbb{R}^{n\times d}$ have rows $(\mathbf{s}_j(\mathbf{Y};\mathbf{X}) - \mathbf{s}_\varnothing)^\mathsf{T}$. Writing $A = W_c^\mathsf{T}$, the teacher residual matrix is $R + SA$. Deployment keeps the folded student, so we hold R fixed and profile over A: adjusting b_c to offset the change in $W_c\mathbf{s}_\varnothing$ keeps the student's constant correction fixed. This algebra does not require a native host bias; bias folding is a separate condition for branch-free deployment. Ignoring the common MSE factor $1/(nH)$, the supervised objective is

$$\mathcal{J}(R, A) = \|R\|_F^2 + \beta\|R + SA\|_F^2,$$

where $\|\cdot\|_F^2$ sums the squared matrix entries.

Let P_S be the orthogonal projector onto the column space of S, the subspace of $\mathbb{R}^n$ spanned by its descriptor columns, and let I be the $n \times n$ identity matrix.

Proof. Decompose each residual column into its projection $P_S R$ and its orthogonal remainder $(I - P_S)R$. Every column of SA lies in the first subspace, so

$$\|R + SA\|_F^2 = \|P_S R + SA\|_F^2 + \|(I - P_S)R\|_F^2.$$

The first term is a squared Frobenius norm, so $\|R + SA\|_F^2 \geq \|(I - P_S)R\|_F^2$ for every A. By definition of the column space, some A satisfies $SA = -P_S R$, making the first term zero and attaining the lower bound. Adding $\|R\|_F^2 = \|P_S R\|_F^2 + \|(I - P_S)R\|_F^2$ proves (7). The argument also covers linearly dependent descriptor columns and nonunique minimisers. □

Equation (7) assigns unit weight to components the summary can linearly explain and an additional teacher weight β to components outside that subspace. If $S = 0$, this reduces to $(1 + \beta)\|R\|_F^2$, which changes the loss scale but not its minimisers. A nonzero summary can instead change the relative weighting of errors and hence which student is preferred within the same function class.

The descriptor's column space therefore determines the relative weights in the profiled criterion, even though its correction is absent at deployment. The following interventions connect this role to the learning paths of the jointly trained method.

C.4 Loss-component and direct-supervision comparisons

Table 5 in the main text, Figure 4, and Figure 5 below study the same five configurations at $H = 96$: DLinear on ETTh1 at 10^{-3} and ETTm2 at $3 \cdot 10^{-4}$, both without a schedule; iTransformer on ETTh1 at $3 \cdot 10^{-4}$ with *type1* and at 10^{-4} without a schedule; and iTransformer on ETTm2 at $3 \cdot 10^{-4}$ with *type1*. These are four distinct model–dataset–horizon cells, with one cell studied at two rates. Every configuration is run over five paired seeds against its supervised reference, and each contrast below is normalised by that reference test MSE.

For seed i, let m_i^0, m_i^E, m_i^{ET}, and m_i^F be test MSEs for the supervised reference, EMA-only student, FoldOPSD with $\lambda = 0$, and full FoldOPSD, respectively. The three sequential contrasts are

$$\begin{aligned} d_i^E &= 100(m_i^E - m_i^0)/m_i^0, \\ d_i^T &= 100(m_i^{ET} - m_i^E)/m_i^0, \\ d_i^K &= 100(m_i^F - m_i^{ET})/m_i^0. \end{aligned} \tag{16}$$

They sum to $100(m_i^F - m_i^0)/m_i^0$ by cancellation, and each is reported as a five-seed mean with its standard error; negative values favour the added component. Each contrast measures an addition at a specified point in the objective, with every arm trained from its matched initialisation: the ordering asks what teacher supervision contributes after EMA is present, and what residual matching contributes once both supervised objectives are included.

Teacher supervision carries the hinted view's direct supervised signal into the shared body. Once EMA is present, this addition lowers test MSE in four of the five configurations, while residual matching provides the final target refinement. Figure 4 shows the paired contrasts and the effect of varying the teacher-loss weight.

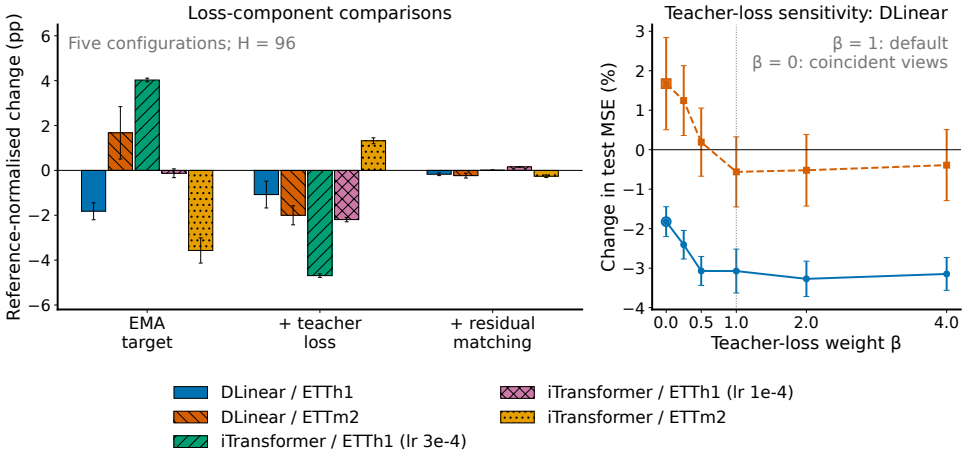

Figure 4: Training-objective interventions at $H = 96$. Left: successive additions of EMA, teacher supervision, and residual distillation, under the same paired reference denominator. Right: teacher-loss-weight β sensitivity on DLinear. Error bars show standard errors over five seeds. iTransformer/ETTh1 contributes two learning-rate settings to the five configurations.

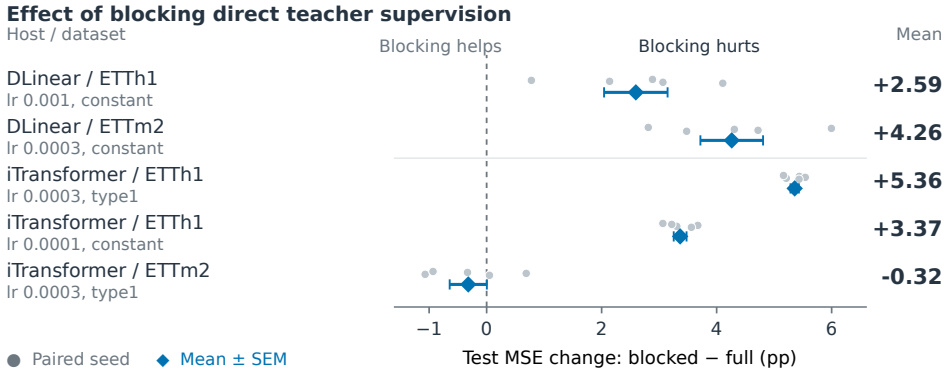

Figure 5: Removing the direct teacher-supervision path increases error in four of five configurations. Blocking the teacher loss's gradient to the shared forecasting body retains the correction's training and the residual distillation target, but removes both the label and hint-dependent body gradients; positive values therefore mean higher student test MSE. Each value is $100(m_i^{\text{blocked}} - m_i^{\text{full}})/m_i^0$ with the same paired reference denominator m_i^0. Circles show five paired seeds, and diamonds and bars the mean and standard error. All rows use $H = 96$ at the learning rates shown, with type1 halving the rate each epoch.

Teacher-body detachment. The body-detachment intervention asks what the complete objective loses when the teacher's direct body update is blocked. Its protocol is the complete objective above, at the same five configurations, matched initialisations, and paired seeds; only the teacher's body output is detached, so the correction still trains and still supplies T_3's teacher target, while the direct body update from T_2 is removed together with its ordinary label-supervision component. Thus the intervention removes both label and hint-dependent gradients from teacher supervision.

Blocking that path costs 2.59–5.36 pp in four of the five configurations, several times the largest five-seed standard error in the same comparison (Figure 5). The increase appears in four configurations covering both hosts and the two rates of iTransformer/ETTh1, each trained from the same objective, data, and paired seeds. The block removes both signals that T_2 sends to the shared body: the ordinary label-supervision component and the hint-dependent component of the affine rescaling $r_t = r_s + a_{\mathbf{x}} \odot Q$ in (6), proportional to $a_{\mathbf{x}}^2 \odot Q$, with $Q = [W_c(\mathbf{s}_j - \mathbf{s}_\varnothing)]_{j=1}^D$ zero at initialisation. The intervention measures the complete shared-supervision path, with the two body-gradient components removed together.

Teacher-loss weight. The two DLinear configurations additionally vary $\beta \in \{0, 0.25, 0.5, 1, 2, 4\}$. At $\beta = 0$, the zero-initialised W_c and $\mathbf{s}_\varnothing$ make the two views coincide under matched stochastic states. Their residual-matching loss is then zero, so these parameters receive no gradient that separates the views. The correction bias can still be updated through T_1. Consequently, $\beta = 0$ removes the mechanism that activates dependence on the privileged descriptor, in addition to removing the direct teacher supervision that every other configuration in the panel retains.

D HORIZON DEPENDENCE

Figure 6 holds the learning rate fixed and lets the horizon vary on ETTh1: mean error falls at every horizon from 96 to 720 steps, both hosts obtain their largest gain at 720 steps.

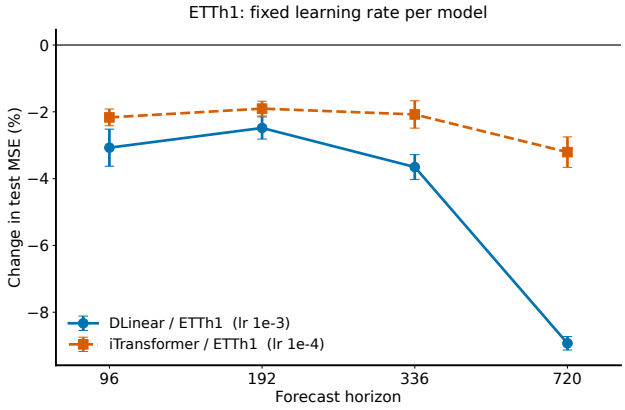

Figure 6: ETTh1 with a fixed rate across horizons: 10^{-3} for DLinear and 10^{-4} for iTransformer, without a schedule. Error bars are five-seed paired standard errors. All eight changes meet the per-cell criterion; the reductions at $H = 720$ are 8.93% and 3.21%, respectively.

E EXTENDED FOUR-HOST FORECASTING STUDY

To examine how the teaching interface transfers across hosts, we evaluate DLinear, iTransformer, FACT (Wang et al., 2026), and MixLinear (Ma et al., 2026) on ETTh1, ETTh2, ETTm1, ETTm2, Weather, and Traffic, at $H \in \{96, 192, 336, 720\}$ with seeds 0–4. We report this host-specific protocol as an extension of the main study. Across 96 complete five-seed cells, macro paired changes are -2.34% MSE and -1.62% MAE, and 68 cells improve in MSE while 74 improve in MAE. Table 13 gives the per-host macro changes and win counts. All hosts use the FoldOPSD objective in Equations (2)–(5), with the same teaching terms and weights.

Configuration protocol. Recipes retain host defaults except for declared grids selected by three-seed reference-validation means. For grouped horizons, the $H = 96$ selection applies to 96/192 and the $H = 336$ selection to 336/720. Cell-level recipes and selection rules are provided in `tables/extended_configurations.csv`. Paired arms share the configuration and training random stream, independently restore their lowest student-validation-MSE checkpoints, and use those checkpoints for the final test evaluation of each arm.

Lookback is 720 for MixLinear and 96 for the other hosts. DLinear uses learning rate 10^{-4}, batch size 32, ten epochs, patience 3, and the type1 schedule throughout. iTransformer uses ten epochs, FACT fifteen, and MixLinear fifteen or thirty. MixLinear uses type3; the others use type1. These settings differ from the reference-rate-selected main study.

Matched deployment interface. Both arms include the same trainable constant correction during training. The three compatible hosts export it into an existing native bias through the bias folding of Appendix B; for FACT, this is `inner.projection.bias`. The numerical check in that appendix covers DLinear and iTransformer. MixLinear has no compatible arbitrary horizon-wise output bias, so both arms retain an H-element correction and its output addition; this matched fixed-correction export is the one host that differs from native bias folding.

Table 13: Four-host evaluation under host-specific recipes. Wins count negative five-seed mean paired changes, so a negative macro value and a win count above 12/24 favour FoldOPSD. Macro values average all 24 cells per host.

Host	MSE wins	Macro Δ MSE (%)	MAE wins	Macro Δ MAE (%)
DLinear	20/24	−5.91	24/24	−4.69
FACT	20/24	−2.17	20/24	−1.37
iTransformer	20/24	−2.05	22/24	−1.43
MixLinear	8/24	+0.78	8/24	+1.03

Table 6 localises these changes by host and dataset. FACT and iTransformer improve on all four ETT datasets at every horizon; DLinear has its largest reductions on ETTh2 and ETTm2. Supplementary `tables/extended_reporting.csv` gives per-horizon means, sample standard deviations, and variances for all 96 cells.

F Training cost and computational complexity

F.1 Computational complexity

On-policy self-teaching trades training computation for an improved student within a fixed deployment architecture. We account for the current teacher, historical student, and answer interface separately. Let B be batch size, K the number of host parameters, and F and G the costs of one host forward pass and its backward pass on a batch. Let A denote the host activations retained for backward propagation. These quantities include the host's dependence on lookback L, horizon H, and channel count D. The descriptor dimension is $d = 3\lfloor H/P \rfloor$.

Computing lookback normalisation and the patch means, standard deviations, and slopes takes $O(BD(L+H))$ time. Applying the shared affine correction and differentiating it takes $O(BDHd)$ time; this bound includes evaluating the student constant in the batched implementation. Bounded residual matching takes $O(BDH)$ time. Updating the EMA parameters takes $O(K + Hd + H + d)$ time. The full update evaluates two current-model views with gradients and one historical view without gradients, giving

$$T_{\text{update}} = 3F + 2G + O(BD(L + H + Hd) + K + Hd)\,. \tag{17}$$

The supervised reference uses one host forward and one backward pass, plus the matched constant correction and optimizer update. For E epochs over N training windows, the training cost is $E\lfloor N/B \rfloor T_{\text{update}}$ when dropping incomplete training batches, as in our implementation, plus validation and checkpoint-selection costs. Since $d = O(H/P)$, the dense hint-to-horizon map contributes $O(BDH^2/P)$; its horizon dependence is not constant when the patch size is fixed.

The correction introduces $Hd + H + d$ trainable parameters. Training also stores an EMA copy and the two current computation graphs; a corresponding memory bound, including inputs and forecast-sized working tensors, is

$$O(K + Hd + 2A + BD(L + H + d))\,. \tag{18}$$

Optimizer states and the EMA copy contribute to parameter storage; measured peak memory and latency are reported separately below.

At export, computing $v = W_{\text{c}}\mathbf{s}_{\varnothing} + b_{\text{c}}$ and updating a compatible bias costs $O(Hd + H)$ once. The hint extractor, correction map, and EMA model are then absent from inference. MixLinear retains the same H-element constant in both compared arms, as specified in Appendix E.

F.2 Measured training overhead

We measure DLinear and TSLib iTransformer on Weather and Traffic at $L = H = 96$, with ten epochs per arm. Both arms use Adam, gradient clipping at 5, float32, and seed 0. Batch size is 32 except for iTransformer on Traffic, where it is 16. The TSLib implementation uses token and feed-forward widths of 512, eight heads, GELU, dropout 0.1, and calendar covariate tokens; it has three blocks on Weather and four on Traffic. The forecasting study instead uses the local iTransformer implementation without calendar covariate tokens, specified in Appendix A. Learning rates are fixed at $3 \cdot 10^{-4}$ except for iTransformer/Weather (10^{-4}). The reference uses supervised student MSE with the same constant-slot parameterisation; FoldOPSD uses all three terms of the objective.

Each arm runs in a fresh process, with the two arms of a pair executing sequentially on the same NVIDIA A100-SXM4-40GB. The environment uses PyTorch 2.7.1, CUDA 12.6, and six CPU threads per process. CUDA events measure 100 updates after 20 warm-up updates in each of the first three epochs, giving 300 measured updates per arm. The interval starts after a batch is retrieved and includes host-to-device transfer, hint extraction, EMA work, all forward and backward passes, gradient clipping, and the optimizer update. Training-epoch wall time includes data loading. Full-fit wall time includes all ten training and validation epochs, optimizer creation, and selected-checkpoint restoration, excluding initial dataset loading and final evaluation. Peak memory is the maximum CUDA-allocated memory over the full fit.

Both arms complete ten epochs and use their lowest-validation-MSE checkpoint. Forecasting performance is evaluated separately in the five-seed studies.

Table 14: Measured training cost at $L = H = 96$ on an A100 (seed 0). Update is the median of 300 post-loader CUDA-event measurements. Train and fit are wall times for all ten epochs; fit additionally includes validation and checkpoint selection. Peak is CUDA-allocated memory over the full fit.

Host	Dataset	Arm	Update (ms)	Train (min)	Fit (min)	Peak (GiB)
DLinear	Weather	Reference	3.50	0.90	1.04	0.020
		FoldOPSD	5.99	1.31	1.46	0.022
DLinear	Traffic	Reference	15.76	1.12	1.33	0.142
		FoldOPSD	16.51	1.20	1.41	0.211
iTransformer	Weather	Reference	13.27	3.16	3.35	0.182
		FoldOPSD	30.98	6.43	6.69	0.258
iTransformer	Traffic	Reference	96.13	12.26	12.94	4.988
		FoldOPSD	213.50	27.12	27.78	10.243

Table 14 compares four paired workloads. On Weather and Traffic, respectively, full-fit time is $1.40\times$ and $1.06\times$ the reference for DLinear, and $2.00\times$ and $2.15\times$ for the iTransformer timing adapter. The corresponding peak-memory ratios are $1.08\times$, $1.49\times$, $1.42\times$, and $2.05\times$. Median-update ratios range from $1.05\times$ to $2.33\times$ across these workloads. The pass-count analysis identifies the work required by self-teaching, and the measurements quantify its implementation cost. Compatible-host bias folding preserves the host's inference operations, whereas MixLinear retains a matched horizon-wise output addition in both arms after training.