LESS DATA, SMALLER BATCHES: WHEN IMPLICIT REGULARIZATION HELPS INFORMED META-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Informed meta-learning predicts on a new task using its few observed examples and task-specific knowledge. When the training pool contains only a few tasks, a model may learn spurious correlations between that knowledge and the observations that fail under distribution shift. Small example batches can discourage reliance on such shortcuts in supervised learning; we ask whether small *task* batches help when entire tasks are scarce. A finite-pool analysis shows that, for vanilla SGD, smaller task batches increase a local penalty on disagreement among task gradients. We analyze why Adam does not inherit the same scalar result, then evaluate both optimizers. With Adam, the small-versus-full-batch predictive log-likelihood benefit is much larger with 25 training tasks than with 200, on both controlled sinusoids and a prospective benchmark constructed from measured weather. This supports a scarcity-dependent optimization effect; the experiments do not identify a particular shortcut. Separately, pre-materializing tasks and overlapping data transfer yield an approximately $15\times$ end-to-end pipeline speedup on one GPU. Code: `github.com/thirdperson11111/iclr_38898.git`.

1 INTRODUCTION

Meta-learning offers a natural framework for settings where each task carries little data but the environment of related tasks is rich enough to learn an inductive bias (Finn et al., 2017; Garnelo et al., 2018a;b). Such settings are common in medicine, where the meta-training pool may consist of only a few dozen related cohorts, as in rare-disease prediction or small-population clinical studies, while the supervision available within each cohort is severely limited. In these regimes, the engineer typically has access to substantial task-specific expert knowledge, including clinical descriptions, diagnostic guidelines, and domain-specific causal structure. Informed meta-learning, recently introduced through the Informed Neural Process (INP) of Kobalczyk & van der Schaar (2025), exploits exactly this asymmetry: it conditions an amortized meta-learner on a knowledge representation $\mathcal{K}$ alongside the in-context data, so that prior information can instead be supplied at test time as a flexible additional input.

A central practical challenge for such meta-learners is that test tasks rarely originate from the same distribution as the meta-training pool. Kobalczyk & van der Schaar (2025) stress-test the INP on a trending-sinusoid benchmark in which training tasks have $b \sim \mathcal{N}(2, 1)$ for the oscillation parameter while test tasks have $b \sim \mathcal{N}(3, 1)$. We frame this shift through the lens of robustness against spurious correlations (Geirhos et al., 2020; Sagawa et al., 2020): with a narrow training pool, an amortized learner may rely on accidental associations between knowledge and context observations that do not hold for new tasks. This is a task-level hypothesis, analogous to reliance on background or color cues in supervised learning (Arjovsky et al., 2019); predictive likelihood alone does not identify which association a model uses.

Recent work in supervised learning has identified a possible lever for group robustness: the implicit regularization of stochastic gradient descent (SGD). Finite-step SGD can approximately follow gradient flow on a modified objective whose correction depends on variation among minibatch gradients (Barrett & Dherin, 2021; Smith et al., 2021). Mirzaie et al. (2026) analyze suppression of a spurious feature in a controlled model; Mirzaie et al. (2025) study small batches on supervised ro-

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.

bustness benchmarks. Their batches contain examples. An informed meta-learner instead averages gradients over tasks, each with its own data and knowledge. *Few tasks in the pool* and *few context observations within a task* are distinct constraints. We ask how changing the first level of batching affects prediction as the pool grows from 25 to 200 tasks.

We hold the INP architecture and training loss fixed and connect task batching to a finite-population gradient covariance. We then test the effect on predictive distributions with matched optimizer updates and independently sampled nested task pools. To move beyond analytic functions, we construct a prospective prediction benchmark from measured weather: context and knowledge end before the target half-day. The available ready-to-use benchmarks did not jointly give us both pool sizes, prediction-time knowledge, and a controlled shift; this construction makes that comparison possible while crediting the underlying measurements and the INP weather precedent (Kobalczyk & van der Schaar, 2025). The mathematical SGD result is local; Adam requires a separate analysis, and both optimizers are evaluated.

Contributions. Our contributions are:

- A **finite-pool task-batching analysis** that explains how small batches change the implicit objective of vanilla SGD: disagreement among task gradients contributes an explicit covariance term. We state when this can disfavor a shortcut-dependent solution and delimit the result for Adam.
- Evidence that the small-versus-full task-batch **predictive-likelihood benefit is larger under task scarcity**, across independently replicated sinusoid pools and measured-weather tasks. Complementary error and coverage metrics show where that benefit does and does not extend.
- A prospective measured-weather benchmark with a clear prediction boundary and nested training pools. It tests a seasonal shift with temporal dependence and measurement variability unavailable in the controlled sinusoid family.
- An approximately $15\times$ **end-to-end pipeline acceleration** through cached task data and overlapped transfer, enabling the original synthetic sweep on a single GPU independently of the predictive and mechanistic findings.

2 Related Work

2.1 Informed and knowledge-aware meta-learning

Meta-learning treats the learner as embedded in a distribution of related tasks and learns a prior over functions that allows rapid adaptation to a new task with limited data. Gradient-based meta-learners such as MAML learn an initialization for task-specific optimization (Finn et al., 2017); amortized Conditional Neural Processes and Neural Processes map context datasets directly to predictions (Garnelo et al., 2018a;b). The INP adds a task-specific knowledge representation to the latter map (Kobalczyk & van der Schaar, 2025). Its Appendix A.8.1–A.8.2 already examines task scarcity and overfitting to knowledge. Our question concerns the controlled interaction between training-pool size and stochastic task-batch optimization, holding the model and objective fixed.

2.2 Implicit regularization of SGD

Modified-equation analyses describe how finite optimization steps differ from gradient flow (Barrett & Dherin, 2021; Smith et al., 2021). Empirical work associates batch size and learning rate with generalization (Keskar et al., 2017; Smith et al., 2020); gradient descent has its own implicit bias on separable data (Soudry et al., 2018). Recent supervised work links small batches to spurious-feature and worst-group behavior (Mirzaie et al., 2026; 2025; Barsbey et al., 2025; Idrissi et al., 2022). In contrast, Adam carries first- and second-moment state across updates; its finite-step bias depends on hyperparameters and training history (Cattaneo et al., 2024). We give an exact finite-pool result for SGD and examine Adam separately.

2.3 GROUP ROBUSTNESS AND SPURIOUS CORRELATIONS

Shortcut learning exploits non-causal cues that correlate with labels in training (Geirhos et al., 2020; Arjovsky et al., 2019). Group DRO targets explicitly annotated groups, including Waterbirds and CelebA (Sagawa et al., 2020; Liu et al., 2015); WILDS includes shifts arising in practice (Koh et al., 2021). Just Train Twice, Deep Feature Reweighting, and Automatic Feature Reweighting reduce reliance on such groups through retraining or reweighting (Liu et al., 2021; Kirichenko et al., 2023; Qiu et al., 2023); Rame et al. (2022) penalize gradient variance across domains. Here task gradients are aggregated by the existing optimizer, without group labels or an auxiliary model. Predictive log likelihood scores the full predictive distribution; error, empirical interval coverage, and width are needed to interpret it (Gneiting & Raftery, 2007; Gneiting et al., 2007).

3 PROBLEM FORMULATION

Tasks and scarcity. Following Kobalczyk & van der Schaar (2025), a task $\tau_i = (\mathcal{D}_{C,i}, \mathcal{D}_{T,i}, \mathcal{K}_i)$ has context input-output pairs $\mathcal{D}_C$, target pairs $\mathcal{D}_T$, and knowledge $\mathcal{K}$. For a fixed meta-training pool $\mathcal{T}_{\text{train}} = \{\tau_i\}_{i=1}^N$, N counts tasks; n_C and n_T count observations *within* each task. One may model their common latent function f by

$$p(\tau) = \int p(\mathcal{D}_C, \mathcal{D}_T \mid f)\, p(\mathcal{K} \mid f)\, p(f)\, df. \tag{1}$$

This is a modeling factorization, conditional on f, rather than evidence that finite-pool knowledge–data associations are invariant under shift.

Training and prediction. The encoder parameters θ_e turn context/target sets into representations $r_{C,i}, r_{T,i}$ and knowledge $\mathcal{K}_i$ into an embedding k_i. The latent posteriors are $q_{T,i} = q_{\theta_e}(z_i \mid r_{T,i}, k_i)$ and $q_{C,i} = q_{\theta_e}(z_i \mid r_{C,i}, k_i)$; decoder $p_{\theta_d}(y \mid x, z_i)$ has parameters θ_d, and $\theta = (\theta_e, \theta_d)$. The implemented training loss is the negative target-set evidence lower bound (ELBO), averaged over tasks:

$$R_i(\theta) = -\mathbb{E}_{q_{T,i}}\left[\sum_{(x,y)\in\mathcal{D}_{T,i}} \log p_{\theta_d}(y \mid x, z_i)\right], \tag{2}$$

$$\ell_i(\theta) = R_i(\theta) + \beta D_{\mathrm{KL}}(q_{T,i} \| q_{C,i}), \qquad C(\theta) = N^{-1}\sum_{i=1}^{N} \ell_i(\theta).$$

The KL direction matters: the target-conditioned posterior is compared with the context-conditioned one. The reconstruction term sums log densities across targets before averaging over latent draws and then tasks; it is not divided by n_T. In the available synthetic code $\beta = 1$. At prediction, targets are unavailable and the context posterior supplies L latent draws. The evaluator uses $\log[L^{-1}\sum_{s=1}^{L} \exp\{\sum_{j=1}^{n_T} \log p_{\theta_d}(y_j \mid x_j, z^{(s)})\}]$, with $L = 16$ in the released synthetic code. Thus predictive log likelihood (LL) is a context-conditioned score on complete fixed evaluation tasks, distinct from the target-conditioned training ELBO. Root mean squared error (RMSE), mean absolute error (MAE), empirical 90% interval coverage, and interval width provide complementary views.

Task versus sample batch. At an update, B distinct tasks contribute gradients $g_i = \nabla\ell_i$, $g_S = B^{-1}\sum_{i\in S} g_i$, and $\bar{g} = N^{-1}\sum_i g_i$. An example minibatch changes the observations *inside* a task; a task batch changes which entire task losses are averaged. We study random reshuffling without replacement, retaining a smaller final batch. Training and validation use the source distribution and evaluation includes shifted tasks. A scarcity interaction compares the small-minus-full predictive LL gain at $N = 25$ with that gain at $N = 200$ using paired nested pools; it does not assume a trend at intermediate N.

4 ANALYSIS

We keep the original model and ELBO, transferring the sample-batch motivation of Barrett & Dherin (2021); Smith et al. (2021); Mirzaie et al. (2026) to task gradients. The task pool is fixed in the result below; expectations refer to sampled batches, not newly drawn tasks or latent Monte Carlo draws.

4.1 FINITE-POOL TASK-GRADIENT COVARIANCE

The covariance of expected task gradients measures how strongly tasks disagree in parameter space:

$$\widehat{\Sigma}_\tau = \frac{1}{N-1}\sum_{i=1}^{N}(g_i - \bar{g})(g_i - \bar{g})^\top. \tag{3}$$

Its trace is the sum of coordinate variances. It is a property of the pool and current θ, not a direct shortcut score. The following identity makes the effect of batching exact.

Lemma 1 (Finite-population task batches)**.** *For a uniform subset S of B distinct tasks, $\mathbb{E}_S\|g_S\|^2 = \|\bar{g}\|^2 + \frac{N-B}{BN}\operatorname{Tr}(\widehat{\Sigma}_\tau)$.*

The finite-population factor vanishes at $B = N$. The centered-gradient proof, including the indicator cross terms, is in Appendix A.

4.2 LOCAL MODIFIED OBJECTIVE FOR SGD

For smooth deterministic expected losses, vanilla SGD with fixed step ϵ, $B \mid N$, $m = N/B$ batches, and a uniformly random nonoverlapping partition P, the first-order modified objective is

$$\widetilde{C}_P^{(1)} = C + \frac{\epsilon}{4m}\sum_{S\in P}\|g_S\|^2, \qquad \mathbb{E}_P\widetilde{C}_P^{(1)} = C + \frac{\epsilon}{4}\|\bar{g}\|^2 + \frac{\epsilon}{4}\frac{N-B}{BN}\operatorname{Tr}(\widehat{\Sigma}_\tau). \tag{4}$$

The batch-averaged SGD endpoint, conditional on P and averaged over batch order, is locally represented by gradient flow on $\widetilde{C}_P^{(1)}$ over time $m\epsilon$ with endpoint error $O(m^3\epsilon^3)$ for sufficiently small $m\epsilon$ (Barrett & Dherin, 2021; Smith et al., 2021). This is a one-pass approximation, not a convergence theorem. The first correction also appears with full-batch gradient descent; the second is the task-covariance contribution. The latter vanishes at full batch even though within-task sampling and latent noise may remain. Appendix A gives the averaging argument and a retained-remainder extension.

For interpretation, suppose a shortcut-dependent candidate has a covariance trace exceeding that of an invariant candidate by $\Delta_\Sigma > 0$. Decreasing batch size from B_2 to $B_1 < B_2$ increases its *relative covariance contribution* by $(\epsilon/4)(1/B_1 - 1/B_2)\Delta_\Sigma$. This conditional comparison does not order the total losses. In particular the coefficient alone does not predict a larger benefit at 25 than at 200 tasks: that scarcity interaction is an empirical question.

Why the finite pool matters. A batch of B tasks sampled from a fixed pool differs from B fresh samples from an unlimited meta-distribution. The chance that a fixed task contributes to a given update is B/N, and distinct batch indicators are negatively dependent. This dependence supplies the finite-population correction in Lemma 1. In a full task batch, every task's expected gradient is averaged at once; variation across the selected *tasks* no longer contributes to the batch-mean gradient's squared norm. The model can still see different context subsamples and latent draws, and stochasticity does not disappear from the practical training loop. If a practitioner instead sampled tasks independently with replacement, the exact $(N-B)/(BN)$ factor would no longer describe that process. The distinction matters at $N = 25$, where a single task can be a substantial fraction of an update.

Interpreting the surrogate. The objective in (4) is an analytical device for a local optimizer trajectory, not a regularizer explicitly added to the code. The mean-gradient term depends on how far the current parameters are from a stationary point. The covariance term measures disagreement in the direction and size of task gradients. If an incidental association fits only some tasks, it can produce such disagreement, giving a reason to test smaller batches. But informative task-specific knowledge can also make gradients heterogeneous. Neither the sign of a learned shortcut nor the ordering of its covariance against an invariant solution follows from the model definition. The conditional comparison above states the additional premise openly; the paired experiments then test a prediction about performance under scarcity that the mathematics alone does not settle.

4.3 ADAM: STATE ACROSS TASK BATCHES

An intuitive way to see the optimizer distinction is to compare two schedules containing the same tasks. With SGD, summing their gradients to first order leaves the same pool gradient, while the second-order correction averages squared batch gradients and exposes covariance. Adam's denominator is computed after squaring *each successive batch gradient*; a large gradient in a singleton remainder updates the second moment differently from four moderate gradients averaged before a single update. The first-moment numerator also remembers where that singleton appeared relative to the preceding blocks. Batch cardinality therefore changes both the gradient and the preconditioner used on later updates. This observation motivates measuring Adam directly rather than substituting η/B for its history-dependent effective step.

The experiments principally use Adam, which sees the same task gradient $g_{S,t}$ but transforms it using persistent moments. With decay rates β_1, β_2 and stabilizer δ, its update is

$$u_t = \beta_1 u_{t-1} + (1-\beta_1) g_{S,t}, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_{S,t}^{\odot 2}, \tag{5}$$
$$\widehat{u}_t = u_t/(1-\beta_1^t), \qquad \widehat{v}_t = v_t/(1-\beta_2^t), \quad \theta_{t+1} = \theta_t - \eta \widehat{u}_t/(\sqrt{\widehat{v}_t} + \delta).$$

Products, roots, and division act coordinatewise. Bias correction depends on update number; the moment state contains preceding reshuffled batches and persists across passes. Consequently two partitions with the same mean and covariance can generate different parameter steps, especially when 25 tasks at $B = 8$ yield three eight-task updates followed by a singleton. Finite-step analyses of Adam give history- and hyperparameter-dependent modified dynamics, sometimes with qualitatively different implicit bias (Cattaneo et al., 2024). We therefore do not attach the scalar SGD covariance penalty in (4) to Adam. The precise conditional SGD result motivates a test; Adam's magnitude and scarcity dependence must be measured. A matched SGD contrast tests whether the predicted direction survives with its covered optimizer.

Our motivating hypothesis is that a scarce task pool makes incidental knowledge–data associations tempting, while disagreements between task gradients can discourage their use. LL improvement measures predictive density, not shortcut reliance. The experiments now test the optimization interaction and its predictive consequences without treating this hypothesis as an identified mechanism.

5 EXPERIMENTS AND RESULTS

5.1 EXPERIMENTAL SETUP

Trending sinusoids. We use the distribution-shifted regression family of Kobalczyk & van der Schaar (2025), $f(x) = ax + \sin(bx) + c$, with $a, c \sim \mathcal{U}[-1, 1]$, 100 inputs on $[-2, 2]$, and Gaussian observation noise with standard deviation 0.2. Throughout, $\mathcal{N}(\mu, 1)$ denotes mean μ and *unit variance*. Training and validation have $b \sim \mathcal{N}(2, 1)$ and the main OOD test $b \sim \mathcal{N}(3, 1)$. Knowledge reveals one uniformly chosen parameter among a, b, c. Context counts range from zero to ten; 100 target points are scored per task. The original 1,000/500/500 training/validation/test split belongs to the submitted baseline; corrected results use eight independently drawn 200-task pools and nested 25-task subsets, two initializations per pool, and 4,000 updates per condition. At 25 tasks, $B \in \{5, 8, 25\}$; at 200, $B \in \{8, 64, 200\}$, so the small-versus-full comparison uses 8 versus 25 and 8 versus 200. The 25-task $B = 8$ schedule includes a singleton (8,8,8,1). The released synthetic code selects checkpoints by validation predictive NLL, evaluates complete fixed splits with 16 latent draws, and aggregates initializations within each independent pool. Appendix C details the schedules and evaluator.

Prospective measured weather. To test whether the effect survives beyond analytic functions, we construct daily tasks from measured weather, following the use of U.S. Climate Reference Network temperature in Kobalczyk & van der Schaar (2025). Twelve measurements in the first half of a day form the context; the model predicts all 144 five-minute measurements in the second half. Its knowledge contains calendar information, preceding-day information, and features available by the context boundary. In particular, no second-half measurement or summary enters context or knowledge (Appendix B). Seasonal shift, correlated days, sensor noise, and varied trajectories make this a different predictive setting from shifted sinusoid frequency. Five independently sampled nested

Table 1: Principal matched-budget scarcity comparisons. Each 25-task pool is nested within its paired 200-task pool; two initializations are averaged within each independent pool. The interaction is computed from the two paired gains rather than treating four means as independent.

Benchmark	Pools	Updates	Small/full at $N = 25$	Small/full at $N = 200$
Sinusoid	8	4,000	$8/25$	$8/200$
Prospective weather	5	2,000	$5/25$	$5/200$

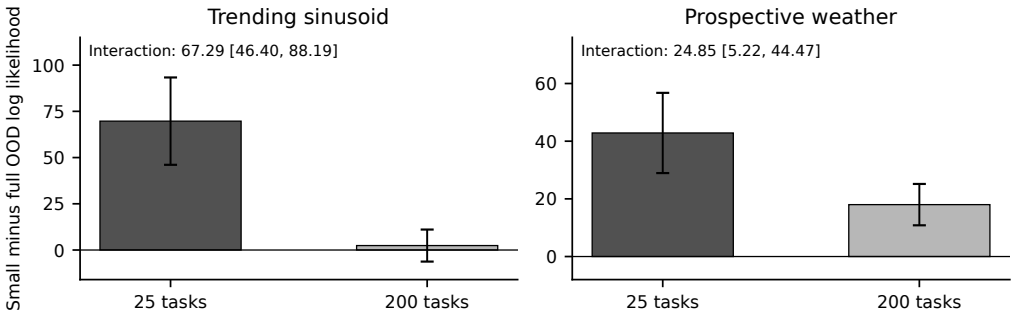

Figure 1: Small-minus-full OOD predictive LL across paired pools (mean and reported 95% CI). Sinusoid: $B = 8$ versus N, eight pools and 4,000 Adam updates; weather: $B = 5$ versus N, five pools and 2,000 updates. An interaction is the paired 25-task gain minus the 200-task gain, separately estimated across pools; the displayed annotation is not an interval formed by subtracting endpoints. Positive means higher small-batch LL.

25/200-task pool pairs, two initializations each, validation-selected learning rates, and 2,000 updates per setting are reported. At both sizes, small batch is $B = 5$ and full batch is $B = N$. The record establishes these task boundaries but does not specify the prospective station, dates, filtering, or raw feature transformations; we do not import those from the earlier controlled-temperature experiment.

Evaluation design and uncertainty. Each small and full batch comparison uses the same independently drawn task pool and two model initializations; at $N = 25$ the tasks themselves form a subset of the paired 200-task pool. This pairing reduces variation in the *difference of differences*. Confidence intervals use the independent pool as the unit: averaging the initializations first avoids counting two runs trained on the same task set as two independent replications. The reported interaction is calculated directly from paired pool-level contrasts, not by subtracting rounded condition means or confidence limits. Validation, rather than shifted test performance, selects checkpoints and learning rates. In the synthetic release, context and knowledge evaluation fixtures and latent-sampling seed are fixed across the compared runs, while task order is reshuffled without replacement during training. These choices make differences easier to attribute to a changed task batch, although they cannot make Adam's internal state identical across batch schedules.

The corrected full-split evaluator matters as much as matching updates. The submitted implementation compared training scores conditioned on targets with context-conditioned test prediction and capped evaluation by a number of batches. That gap did not compare like with like. Here both train and test predictive LL use the context posterior over complete fixed task sets. A log score rewards density assigned to the observed target trajectory, whereas RMSE summarizes a point prediction and interval coverage tests whether uncertainty is calibrated. We report them together because a model can gain LL by widening a too-narrow predictive distribution even as its mean prediction gets worse. The calendar-day task in the prospective benchmark also changes the scientific question: its knowledge is restricted to the time the prediction is made, unlike the earlier whole-trajectory-informed temperature regression.

Why prospective construction is necessary. The original INP temperature study is a valuable precedent for using measured USCRN trajectories, but one of its knowledge settings summarizes an

Table 2: Absolute OOD prediction at validation-selected checkpoints. LL sums target log densities per task; RMSE and MAE are in each benchmark's response units. Coverage is empirical coverage of nominal 90% intervals, and width must be read with it. Different target sets make raw LL values incomparable between benchmark blocks. Reported LL/RMSE uncertainty for sinusoid is across eight pools (95% CI in Appendix D); no absolute weather CIs are available.

Benchmark	N, B	OOD LL	RMSE	MAE	Coverage	Width
Sinusoid	25,5	-125.0	0.770	0.626	0.739	1.741
	25,8	-117.1	0.777	0.632	0.762	1.849
	25,25	-186.8	0.732	0.594	0.634	1.284
Weather	25,5	-316.93	2.157	1.767	0.837	4.833
	25,25	-359.77	2.569	2.086	0.730	5.014
	200,5	-340.52	2.506	2.018	0.787	5.195
	200,200	-358.52	2.689	2.183	0.746	5.386

entire day while predicting that same day's observations. Such information is useful for controlled knowledge integration; it is unavailable when forecasting the second half of a day from its first half. The prospective construction moves the boundary between input and target to a real prediction time, retains a substantial future trajectory as the target, and supplies a preceding day and calendar cues without seeing the future. It also supplies enough eligible daily tasks to compare nested training pools at both 25 and 200 tasks. The response documents this design and its results; it does not document a specific station or date range for that later construction. A station name copied from the earlier experiment would make the new benchmark appear more reproducible than the evidence permits. The achieved contribution is a controlled real-measurement evaluation of the interaction, with its precise source provenance limited to what the available record supports.

5.2 Scarcity dependence in predictive likelihood

Figure 1 answers the central question. On the sinusoid benchmark, the small-minus-full OOD LL gain is 69.70 [46.08, 93.32] at $N = 25$ and 2.41 [-6.26, 11.07] at $N = 200$. Their *paired* difference is 67.29 [46.40, 88.19], positive in all eight pools. On prospective weather the analogous gains are 42.85 [28.94, 56.76] and 18.00 [10.83, 25.17], with interaction 24.85 [5.22, 44.47], positive in all five pools. Hence the predictive-density benefit is larger with scarce tasks on both benchmarks. This does not establish a reversal or monotonicity across untested pool sizes; the weather small batch still helps at 200 tasks.

Table 2 separates predictive density from point accuracy and interval quality. On scarce sinusoids, $B = 8$ improves LL and nominal coverage error by 0.128 [0.102, 0.153] against full batch but worsens RMSE by 0.046 [0.029, 0.063] and MAE; its wider intervals help coverage. The train predictive LLs are $-71.7, -72.6, -62.1$ at $B = 5, 8, 25$; corresponding train-minus-test predictive gaps are 53.3 [40.5,66.0], 44.5 [34.5,54.5], and 124.7 [98.9,150.5]. This gap is secondary because its test term already contains the outcome of interest. On weather, small batch improves both likelihood and point accuracy at each pool size. The difference *between the RMSE improvements* at the two pool sizes is 0.229 [0.083,0.375]; the analogous coverage-error interaction is 0.066 [0.021,0.111]. These are scarcity interactions, not absolute improvements at $N = 25$.

What the absolute results add. The scarce-sinusoid condition with $B = 25$ assigns relatively concentrated 90% intervals but covers only 63.4% of observations. The $B = 8$ intervals are wider and cover 76.2%, still short of their nominal rate; the improved LL and coverage do not erase the small point-error disadvantage. At $B = 5$, the OOD LL is -125.0 rather than -117.1 for $B = 8$, so the experiment does not support a monotone "smaller is better" rule even within the 25-task pool. On prospective weather, the 25-task small/full comparison changes LL from -359.77 to -316.93, RMSE from 2.569 to 2.157, and coverage from 0.730 to 0.837. The same metrics favor $B = 5$ at 200 tasks, although the likelihood contrast is smaller. This is why the central claim concerns the *size of the small-batch benefit under scarcity*, rather than a supposed optimum that reverses once more tasks become available.

Table 3: Matched scarce-sinusoid optimizer comparison: $N = 25$, $B = 5$ minus $B = 25$, 4,000 updates, best validation checkpoint; eight independent pools. Intervals are paired 95% CIs. Positive RMSE difference means worse point error for the small batch.

Optimizer	OOD LL difference	RMSE difference
Adam	61.8 [41.1,82.5]	+0.039 [0.020,0.057]
SGD ($\eta = 10^{-3}$)	1.333 [0.681,1.985]	−0.004 [−0.009,0.001]

5.3 Optimizers, budgets, and shift families

The SGD surrogate concerns vanilla SGD, whereas the large LL effect uses Adam. Table 3 compares the same divisible 25-task $B = 5$ versus full-25 task contrast for both optimizers. At step size 10^{-3}, the stable SGD sweep completed 144 runs across configurations; it yields LL gain 1.333 [0.681,1.985], positive in 8/8 pools, and coverage-error improvement 0.0103 [0.0028,0.0178]. These runs are not 144 independent pools. Adam's much larger 61.8 [41.1,82.5] gain cannot be calculated from the SGD covariance coefficient. The Adam $B = 8$ contrast is 69.7 [46.1,93.3], but it includes retained singleton updates. The different magnitudes justify the optimizer distinction in Section 4.3.

A distinct equal-exposure arm fixes 25,000 task exposures and gives 5,000, 4,000, and 1,000 updates at $B = 5, 8, 25$. At the *final* checkpoint, $B = 5$ reverses to -29.1 [-45.8,-12.3] LL against full, while $B = 8$ retains $+51.2$ [22.9,79.6]. Thus the batch choice and budget matter. Gradient accumulation of microbatches of eight to effective full batch 25 gives -2.6 [-14.4,9.2] versus native full batch: its interval spans zero, compatible with similar performance but not a formal equivalence result. A specific Adam learning-rate scaling (Appendix C) gives $+85.0$ [61.2,108.8] and $+83.9$ [60.0,107.7] LL at $B = 5, 8$ versus 25. These controls address update count, nominal scaling, and microbatch execution as sufficient explanations, without isolating one causal mechanism.

Across seven exploratory sinusoid test families, $B = 8$ has positive LL contrasts over full batch: 22.9 [8.5,37.3] on an *in-distribution* Gaussian, 31.0 [19.7,42.2] with opposite mean, 60.2 [38.5,82.0] with positive mean, 32.1 [17.8,46.4] for a bimodal mixture, 241.2 [116.5,365.9] for simultaneous positive shift, 23.6 [9.4,37.9] for skewed gamma, and 80.5 [63.5,97.4] for wide variance. The breadth of LL improvement, including ID, argues against an OOD-exclusive interpretation. Appendix D plots these contrasts and documents a distinct earlier measured-temperature protocol whose paired LL result was inconclusive.

5.4 Pipeline and interpretation of the controls

A synthetic task is small, but loading it from the original tables was expensive: item access repeatedly traversed a curve table, searched for knowledge, and rebuilt the shared input grid. Prematerializing contiguous task and knowledge arrays turns these operations into indexing. Persistent loader workers, pinned host memory, and nonblocking transfer overlap the next batch with GPU computation. On the original single RTX 3090 workstation with three concurrent jobs, the optimized historical INP $B = 8$ run averaged 26.5 minutes and the $B = 128$ run 3.8 minutes. Multiplying these *measured optimized* times by the reported approximately 15-fold improvement gives 397.5 and 57.0 minutes as *estimated* previous-pipeline times; Appendix C includes NP timings too. The comparison is setup-specific and predates the corrected 4,000-update runs. It explains how the task-pool analysis became tractable, without acting as evidence that a particular optimizer mechanism improved prediction.

The controls delimit what a batch-size intervention actually changes. Equal update counts hold the number of parameter steps fixed but allocate more task exposures to large batches; equal task exposures fix the number of tasks processed but change parameter steps. Neither can force Adam to follow the same trajectory after a change in B. Accumulation asks a narrower question: if the same microbatches are summed before the optimizer update, does nominal execution at eight tasks recreate the eight-task effect? Its LL interval spans zero relative to native full batch, supporting the role of *effective optimizer batch size* in this comparison. The scaled-rate arm checks one specified learning-rate rule, but Adam's moment normalization prevents that rule from exactly matching an SGD-style η/B noise scale. Finally the stable SGD arm establishes a gain in the direction suggested

by the conditional theory, much smaller than Adam's; the mechanisms governing their magnitudes remain distinct.

The exploratory test families sharpen the interpretation of this result. Positive LL contrasts on shifted Gaussian, mixture, skewed, wide-variance, and simultaneous parameter changes show that the effect is not limited to one sinusoid test distribution. A positive contrast also appears on the in-distribution Gaussian family. It would be incorrect to label all seven rows OOD or to claim a benefit specific to distribution shift. The measured claim instead concerns predictive density under the evaluated task families and the *difference between scarce and larger training pools*. The prospective forecast provides an additional dataset-level comparison in which improved density comes with improved point error, addressing one important limitation of the scarce synthetic result while remaining a single measured-data construction.

6 Conclusion, Discussion, and Future Work

Conclusion. We have studied task-batch size as an optimization choice in informed meta-learning. A finite-pool analysis gives a local SGD modified objective with an explicit task-gradient covariance contribution; it connects batch size to disagreement across tasks under stated conditions. With Adam, corrected matched-update experiments show a larger small-versus-full-batch predictive-likelihood gain at 25 than at 200 training tasks on both trending sinusoids and prospective measured weather. The weather benchmark tests prediction with knowledge available before the target half-day. On sinusoids the LL gain accompanies worse point error; on weather, point accuracy and coverage move in the same direction. An approximately $15\times$ pipeline acceleration made the original synthetic analysis practical on one GPU.

Discussion. Our task-level spurious-correlation account motivates the analysis, but these experiments do not identify a shortcut or establish a universal ordering of solutions. The formal surrogate applies locally to smooth expected task losses under vanilla SGD and divisible, randomly partitioned batches; Adam has persistent adaptive state, and its empirical gain is much larger than the SGD gain. The synthetic LL–RMSE trade-off and budget-dependent $B = 5$ reversal qualify practical recommendations. A separate earlier temperature regression experiment, with whole-day knowledge, had an inconclusive LL contrast; the prospective benchmark instead enforces a prediction-time information boundary. Neither experiment evaluates clinical deployment.

Future work. We would like to understand which knowledge–data associations are learned under scarcity, derive an optimizer-matched account of Adam with reshuffling and retained remainder batches, and test the interaction in additional task families, including classification and settings with natural-language knowledge. These questions concern the scope and mechanism of the observed predictive effect; they do not change the matched comparisons reported here.

AI USE STATEMENT

We used large language models (LLMs) to assist with editing tasks, including improving clarity and grammar, as well as with non-core code maintenance tasks such as formatting, adding docstrings, and renaming. All LLM-assisted modifications were reviewed by the authors, who retain full responsibility for the final content.

REPRODUCIBILITY STATEMENT

The fixed-pool derivations and remainder case appear in Appendix A; dataset construction and the prediction boundary in Appendix B; configurations, evaluator, schedules, and pipeline in Appendix C; and corrected controls and diagnostics in Appendices D–E. The accessible code repository contains the synthetic experiment code and data, but not the prospective weather processing or raw replication results.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. URL https://arxiv.org/abs/1907.02893.

David G.T. Barrett and Benoit Dherin. Implicit gradient regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Melih Barsbey, Lucas Prieto, Stefanos Zafeiriou, and Tolga Birdal. Large learning rates simultaneously achieve robustness to spurious correlations and compressibility. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Matias D. Cattaneo, Jason Matthew Klusowski, and Boris Shigida. On the implicit bias of adam. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5862–5906. PMLR, 2024. URL https://proceedings.mlr.press/v235/cattaneo24a.html.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes, 2018b.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B*, 69(2):243–268, 2007. doi: 10.1111/j.1467-9868.2007.00587.x.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 336–351. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/idrissi22a.html.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Zb6c8A-Fghk.

Katarzyna Kobalczyk and Mihaela van der Schaar. Towards automated knowledge integration from human-interpretable representations. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=AbpQzx79T9. arXiv:2402.16105.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/koh21a.html.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015. URL https://arxiv.org/abs/1411.7766.

Nahal Mirzaie, Mahdi Ghaznavi, Hosna Oyarhoseini, Alireza Alipanah, Erfan Sobhaei, Ali Abbasi, Amirmahdi Farzaneh, Hossein Jafarinia, Parsa Sharifi Sadeh, Arefeh Boushehrian, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. The silent helper: How implicit regularization enhances group robustness. In *3rd Workshop on High-dimensional Learning Dynamics (HiLD), ICML*, 2025.

Nahal Mirzaie, Alireza Alipanah, Ali Abbasi, Amirmahdi Farzane, Hossein Jafarinia, Erfan Sobhaei, Mahdi Ghaznavi, Amir Najafi, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. Implicit regularization of SGD reduces shortcut learning. In *International Conference on Learning Representations (ICLR)*, 2026. Code: https://github.com/mirzanahal/sgd-implicit-regularization-shortcuts.

Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and fast group robustness by automatic feature reweighting. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2306.11074.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18347–18377. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/rame22a.html.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations (ICLR) 2020*, Addis Ababa, Ethiopia, April 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9058–9067. PMLR, Jul 2020.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018. URL https://www.jmlr.org/papers/v19/18-188.html.

A THEORY

Centered-gradient expansion. Fix θ and let $\delta_i = g_i - \bar{g}$, so $\sum_i \delta_i = 0$. For subset indicators $I_i = \mathbf{1}\{i \in S\}$, uniform size-$B$ sampling without replacement gives $\mathbb{E}I_i = B/N$ and $\mathbb{E}I_iI_j = B(B-1)/[N(N-1)]$ for $i \neq j$. The cross term in $\|g_S\|^2 = \|\bar{g} + B^{-1}\sum_i I_i\delta_i\|^2$ disappears since $\sum_i \delta_i = 0$. Further,

$$\begin{aligned}\mathbb{E}_S\left\|B^{-1}\sum_i I_i\delta_i\right\|^2 &= B^{-2}\left\{\frac{B}{N}\sum_i\|\delta_i\|^2 + \frac{B(B-1)}{N(N-1)}\sum_{i\neq j}\delta_i^\top\delta_j\right\}\\ &= B^{-2}\left\{\frac{B}{N} - \frac{B(B-1)}{N(N-1)}\right\}\sum_i\|\delta_i\|^2\\ &= \frac{N-B}{BN(N-1)}\sum_i\|\delta_i\|^2 = \frac{N-B}{BN}\operatorname{Tr}(\widehat{\Sigma}_\tau).\end{aligned}$$

The middle step uses $\sum_{i\neq j}\delta_i^\top\delta_j = -\sum_i\|\delta_i\|^2$. This proves Lemma 1; using a $1/N$ covariance would require a different coefficient.

Partition averaging and scope. In the modified-equation one-pass approximation of Barrett & Dherin (2021); Smith et al. (2021), smooth expected losses give the first-order surrogate $\widetilde{C}_P^{(1)}$ in (4). Conditional on a fixed partition P, averaging the ordering of its m batches cancels the order-specific cross terms at the relevant order; the resulting gradient flow over time $m\epsilon$ represents the local mean endpoint with $O(m^3\epsilon^3)$ retained endpoint error when $m\epsilon$ is sufficiently small. Each block in a uniformly random partition is marginally a uniform subset of size B, so linearity of expectation and Lemma 1 yield

$$\mathbb{E}_P\frac{1}{m}\sum_{S\in P}\|g_S\|^2 = \|\bar{g}\|^2 + \frac{N-B}{BN}\operatorname{Tr}(\widehat{\Sigma}_\tau).$$

The formula concerns one local pass at a fixed point, not a statement about repeated-pass convergence. The full-batch endpoint removes only the task covariance term. Latent sampling, randomly redrawn contexts, and observation-level noise have been integrated into the expected ℓ_i and can generate separate optimization randomness in practice.

Conditional comparison. Holding the two candidate parameter points and their mean-gradient contributions distinct, let $\Delta_\Sigma = \operatorname{Tr}(\widehat{\Sigma}_\tau^{\text{shortcut}}) - \operatorname{Tr}(\widehat{\Sigma}_\tau^{\text{invariant}}) > 0$. The change in their *covariance-part difference* between B_2 and $B_1 < B_2$ is $\frac{\epsilon}{4}(1/B_1 - 1/B_2)\Delta_\Sigma > 0$. No universal ranking of the full surrogate or of OOD prediction follows.

Retained remainder. If $N = mB + r$ with $0 < r < B$, there are $m+1$ updates, one of size r. Applying the lemma to each batch size gives the average task-covariance coefficient

$$\alpha_{N,B,r}^{\text{rem}} = \frac{1}{m+1}\left[m\frac{N-B}{BN} + \frac{N-r}{rN}\right]. \tag{6}$$

For $N = 25, B = 8$ the schedule is 8,8,8,1: the singleton differs sharply from an ordinary size-eight step. $B = 5$ versus 25 is a divisible comparison. A configured $B > N$ is effective full batch $B_{\text{eff}} = N$. Equation (6) is a coefficient for unequal sampled batch sizes; it does not turn the retained-batch Adam process into the divisible SGD proposition.

Optimizer detail. In (5) the second moment is of elementwise squared *batch* gradients, not a fixed covariance trace; the recursive state includes gradient signs, order, scale, and bias-correction index. Since an update is nonlinear in $\widehat{u}_t, \widehat{v}_t$, averaging g_S before this map differs from averaging mapped updates. Random reshuffling and retained singleton batches alter subsequent moments. Cattaneo et al. (2024) analyzes finite-step implicit bias for Adam and RMSProp under specific conditions; it does not supply a scalar finite-population surrogate for the exact practical schedule here.

B DATASETS

Trending-sinusoid tasks. The original INP construction uses $f(x) = ax + \sin(bx) + c$ with $a, c \sim \mathcal{U}[-1, 1]$, a 100-point shared grid on $[-2, 2]$, additive observation noise $\mathcal{N}(0, 0.2^2)$, and one knowledge item revealing a uniformly chosen member of $\{a, b, c\}$. Training/validation frequency $b \sim \mathcal{N}(2, 1)$ and principal shifted test $b \sim \mathcal{N}(3, 1)$ specify variance in the second Gaussian argument. Original data were split 1,000/500/500; the corrected nested 25/200 pools are a separate campaign. The accessible synthetic code contains CSVs for the original distribution and mean-shift 2.5, 3.0, and 3.5 variants, but not a generator or raw data/configurations for all seven later exploratory families. Their precise distributional parameters beyond the reported family names cannot be authenticated from the provided record, so we report only the verified contrasts in Figure 2.

Prospective measurement boundary. The underlying weather precedent uses U.S. Climate Reference Network (USCRN) temperature measurements (Kobalczyk & van der Schaar, 2025). The prospective response defines a distinct forecasting task and does not document its station/date selection. Table 4 records only confirmed information availability; source processing, missingness rules, eligibility criteria, station/date, calendar encodings, prior-day summary definitions, and train-only preprocessing fits are unavailable in the accessible materials. Those facts cannot be recovered by transplanting the original INP's 2021–2022 Aleknagik setup into this later experiment.

Table 4: Prospective daily task and prediction-time information. Five nested pool pairs compare 25 and 200 tasks; knowledge is restricted to data available at or before the context boundary.

Source	Availability for prediction
Calendar information	Available before the target half-day.
Preceding-day information	From the previous day; particular reductions not specified.
Current-day context	12 observed first-half-day measurements.
Target	All 144 second-half-day measurements, withheld from inputs and knowledge.

Earlier controlled temperature regression. This distinct response used two years (2021–2022) of USCRN Aleknagik, Alaska sub-hourly temperature data, one calendar day per task with 288 five-minute measurements. Five independently sampled 25-day warm-season training pools and two initializations were evaluated on held-out cold-season days after 4,000 updates. Knowledge included the *whole trajectory's* minimum and maximum, so this task is controlled informed regression and cannot be described as prospective prediction. These temperature targets and the prospective second-half targets differ in count and units of LL aggregation.

C IMPLEMENTATION

Architecture and training. The original latent INP has permutation-invariant context and knowledge encoders, sum aggregation into a latent variable, and a feed-forward decoder (Kobalczyk & van der Schaar, 2025). The original source describes hidden dimension 128, three GELU decoder layers, and latent dimension 128; synthetic code in the accessible anonymous repository exposes its specific model and loss. The corrected synthetic runs use task-pool seeds 1001–1008, initialization seeds 0 and 1, Adam learning rate 10^{-3} except designated controls, 4,000 updates, and random reshuffling without replacement; incomplete batches remain. Validation is evaluated every 500 updates and selects minimum full-split predictive NLL. Test tasks do not choose the checkpoint. The prospective campaign instead uses 2,000 updates and validation-selected learning rates; its optimizer parameters and construction files are not included in the accessible repository, so the original synthetic defaults cannot certify those settings.

Evaluator and uncertainty. The synthetic evaluator fixes within-task context and knowledge fixtures across configurations, uses 16 context-conditioned latent draws in evaluation mode, sums target log densities per draw, and applies log-mean-exp across draws. It then averages LL over complete tasks. Pool-mean contrasts first average the two initializations within each independent pool; 95%

two-sided t intervals and interactions are computed from paired pool differences. The published intervals are transcribed, not recomputed from absent raw run records. Target noise, latent sampling, and context fixtures are different sources of variation from which tasks enter a batch. These definitions apply directly to the released synthetic evaluator; the response describes a common complete-split context-conditioned evaluator for the weather campaign but does not give its latent-draw count or data preprocessing implementation.

Budget schedules and learning-rate control. At 25 tasks $B = 5, 8, 25$ gives each reshuffled pass 5, 4, and 1 optimizer updates, respectively, with cardinalities 5×5; 8,8,8,1; and 25. At 200 tasks $B = 8, 64, 200$ gives 25, 4, and 1 updates per pass, with the 64 schedule 64,64,64,8. All corrected sinusoid primary comparisons fix 4,000 updates. The equal-exposure control instead fixes 25,000 tasks seen, giving 5,000, 4,000, and 1,000 updates for $B = 5, 8, 25$; its table uses final, not validation-selected, checkpoints. The specific LR-scaling control sets Adam rates $0.0003125, 0.0005, 0.0015625$ at $B = 5, 8, 25$ respectively. This is a tested scaling rule, not exact control of Adam's effective noise. Accumulation processes microbatches of eight and steps only after effective size 25. Stable SGD uses learning rate 10^{-3}; higher attempted rates had numerical instability according to the released synthetic summary.

Single-GPU pipeline. The submitted implementation used a single RTX 3090 (24 GB) and Intel Core i9, running three concurrent jobs; timing therefore reflects contended throughput. Original item access repeatedly indexed a pandas curve table, scanned a knowledge table, and allocated the shared grid. The improved pipeline cached contiguous NumPy arrays, a common grid, and knowledge keyed by task, while preserving tensor values. Persistent loader workers, prefetching, pinned host memory, and nonblocking transfers overlapped data work with GPU computation. Table 5 reports historical optimized wall time and a 15-fold *estimate* of original-pipeline time; it is separate from the corrected predictive campaigns. Neither an idle-machine timing comparison nor component ablation was reported.

Table 5: Historical three-seed average wall time per run on one contended RTX 3090; previous-pipeline column is estimated as $15\times$ the measured optimized time. These timing runs are separate from corrected 4,000-update predictive results.

Model	B	Optimized measured (min)	Previous pipeline estimated (min)
NP	8	10.6	159.0
NP	128	3.0	45.0
INP	8	26.5	397.5
INP	128	3.8	57.0

D RESULTS AND CONTROLS

Table 6: Corrected scarce-sinusoid absolute metrics at best validation; 95% CIs across eight independent task pools when reported. Train LL and interval metrics were supplied as point estimates.

B	Train LL	OOD LL [95% CI]	RMSE [95% CI]	MAE [95% CI]	Gap [95% CI]
5	−71.7	−125.0 [−139.6,−110.5]	.770 [.743,.798]	.626 [.602,.650]	53.3 [40.5,66.0]
8	−72.6	−117.1 [−125.9,−108.4]	.777 [.750,.805]	.632 [.608,.655]	44.5 [34.5,54.5]
25	−62.1	−186.8 [−216.0,−157.7]	.732 [.714,.749]	.594 [.579,.609]	124.7 [98.9,150.5]

Pool replication and NP. Every corrected 25-task synthetic pool is nested in a corresponding independently drawn 200-task pool; the two initializations do not create independent pool observations. The variance of pool means / conditional initialization variance / estimated pool component is 304.1/66.1/271.0 at $B = 5$, 109.4/45.1/86.9 at $B = 8$, and 1216.5/183.6/1124.7 at full 25. The last component subtracts half the conditional initialization variance from the pool-mean variance because there are two initializations per pool. These quantities are *variances*, not standard deviations. A released synthetic summary gives INP-minus-NP small-batch interaction -1.8 [-10.3,6.6],

Table 7: Reported paired scarce-sinusoid controls ($N = 25$); LL and RMSE differences are left minus native full $B = 25$ with 95% pool-level CIs. Fixed-update rows select by validation; equal-exposure rows use final checkpoints.

Comparison	OOD LL difference	RMSE difference
Adam $B = 5$, 4,000 updates	61.8 [41.1,82.5]	.039 [.020,.057]
Adam $B = 8$, 4,000 updates	69.7 [46.1,93.3]	.046 [.029,.063]
Equal exposures final, $B = 5$	-29.1 [-45.8,-12.3]	$-.009$ [$-.021$,.002]
Equal exposures final, $B = 8$	51.2 [22.9,79.6]	.008 [$-.003$,.020]
Accumulated micro 8, effective 25	-2.6 [-14.4,9.2]	.004 [$-.005$,.013]
LR-scaled Adam, $B = 5$	85.0 [61.2,108.8]	.027 [.012,.042]
LR-scaled Adam, $B = 8$	83.9 [60.0,107.7]	.041 [.023,.059]
Stable SGD $B = 5$	1.333 [.681,1.985]	$-.004$ [$-.009$,.001]

with no detectable knowledge-specific amplification. The response reports a separate prospective NP scarcity interaction of 28.42 [16.93,39.90], but supplies no underlying NP condition table; it is therefore an interaction-only comparison.

Shift families. Figure 2 plots reported $B = 8$ minus full-25 test LL for seven families using the same eight pools. Full generating parameters and raw pool observations were not provided for this exploratory campaign; plotting uses only the published summary endpoints.

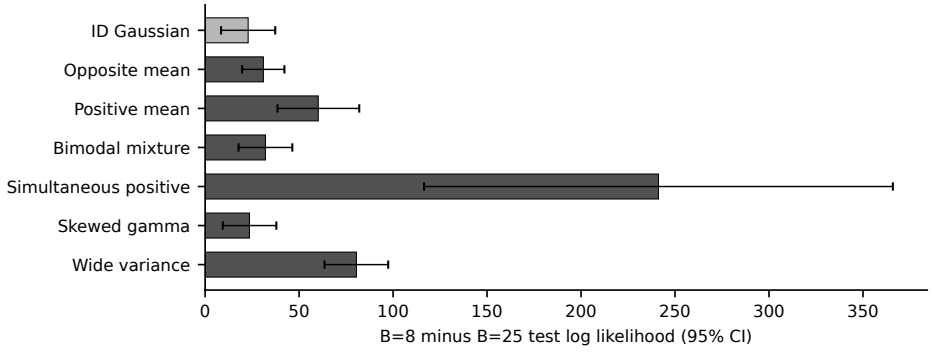

Figure 2: Exploratory sinusoid-family LL contrasts, $B = 8$ minus 25, eight pools with 95% CIs. ID Gaussian is in-distribution; the other six are shifted families. Values and interval endpoints come from the review response; no observations are simulated.

Separate temperature protocol. On whole-trajectory-informed USCRN Aleknagik daily regression, $B = 5$ OOD LL was -4330.7 [-4932.3,-3729.2], and full $B = 25$ was -4270.0 [-5990.3,-2549.7]. Paired LL difference -60.7 [-1608.6,1487.2] favored small batch in 2/5 pools: inconclusive. RMSE was 11.969 [11.685,12.253] versus 12.411 [10.851,13.971], MAE 11.672 [11.384,11.959] versus 12.134 [10.549,13.718], and 90% coverage 0.198 [0.182,0.214] versus 0.176 [0.147,0.206]. Both were severely under-covered. Its whole-day targets mean raw LL scale must not be compared to the prospective half-day forecast.

E DIAGNOSTICS

A fixed set of 32 tasks served as the diagnostic probe. At each checkpoint, trace of task-gradient covariance and the squared norm of the mean gradient were compared with OOD LL over $n = 48$ configurations/runs, *not* independent task pools. A normalized-noise diagnostic was also reported; its exact denominator is not specified in the accessible record and is not inferred here. Table 8 gives the correlations. The weak normalized-noise association and shared variation in model fit prevent causal identification of implicit regularization.

Table 8: Correlations with OOD LL on the fixed 32-task diagnostic set; each row uses 48 configurations/runs. Values are associations, not independent-pool inferential tests.

Checkpoint	Diagnostic	Pearson	Spearman
Best validation	Covariance trace	$-.584$	$-.455$
	Mean-gradient squared norm	$-.468$	$-.373$
	Normalized noise	$-.026$	$-.205$
Final	Covariance trace	$-.834$	$-.919$
	Mean-gradient squared norm	$-.762$	$-.882$
	Normalized noise	$.159$	$-.013$