Improving Neural Processes in the Low-Data Regime via Context-Subset Training and Self-Distillation

Hyungi Lee
Kookmin University
lhk2708@kookmin.ac.kr

Jangho Kim*
Kookmin University
jangho.kim@kookmin.ac.kr

Abstract

Neural processes (NPs) are flexible uncertainty-aware meta-learners, but often fail when meta-training tasks are scarce, losing generalization and collapsing into task memorization. We propose Context-Subset Self-Distillation Neural Processes (CSSDNP), a training framework that trains a student on random context subsets while distilling the full-context predictive distribution of an EMA teacher from the same task. This context-asymmetric objective combines reduced-context predictive learning with forward-KL distillation, aligning predictive distributions across different context views without architectural or test-time changes. Theoretically, we show that context-subset training learns the reduced-context prediction problem while providing multiple context views of each task, and that calibrated full-context distillation preserves the asymptotic Bayes target while improving finite-sample estimation through richer teacher supervision. Empirically, CSSDNP improves robustness and generalization across NP variants and tasks, especially in low-meta-data regimes.

1 Introduction

Neural processes [NPs; 11, 12] provide a probabilistic meta-learning framework [25, 24] for modeling distributions over functions. By conditioning on a context set and amortizing posterior prediction at target inputs, they combine the flexibility of stochastic processes with the efficiency of neural networks, making them useful for uncertainty-aware regression, sequential prediction, Bayesian optimization, and partially observed structured prediction [22, 26, 16, 9]. However, NPs learn their task prior from a finite set of meta-training tasks. When these tasks do not cover test-time variations in function shape, input coverage, noise level, context sparsity, or task dimensionality, the learned prior can become overly narrow. This low-meta-data regime [13] is common in scientific and real-world domains, where tasks may correspond to costly experiments, rare trajectories, personalized systems, or limited temporal snapshots [5, 41]. Consequently, standard NP training can yield overconfident predictions and poor generalization to unseen tasks.

In this work, we study how NPs fail in the low-meta-data regime. We find that standard training exhibits a characteristic *generalization collapse*: with limited meta-training tasks, training likelihood continues to improve while target likelihood on unseen tasks sharply deteriorates. This reveals *task-level memorization*, where the model specializes to the finite set of observed task functions rather than learning a predictive mechanism that transfers to the broader task distribution. This phenomenon is closely related to memorization in meta-learning [46] and motivates meta-augmentation approaches [37]. Addressing this failure requires extracting more conditional training signals from each available task while preserving the predictive structure needed for uncertainty-aware generalization. Our key perspective is therefore to regularize the conditional predictive distributions induced by different context views of the same task. Unlike standard consistency regularization, where the teacher

*Corresponding author.

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

and student receive the same input under different augmentations, context reduction in NPs changes the Bayesian posterior itself, making the teacher–student gap a principled information-asymmetric inference problem rather than mere noise injection.

To this end, we propose *Context-Subset Self-Distillation Neural Processes* (CSSDNP), a training framework that improves low-meta-data robustness without changing the NP architecture or adding test-time cost. CSSDNP trains a student on random subsets of the original context, so that each task induces multiple reduced-context prediction problems within a context-target training episode. Unlike standard context-target resampling in NP training, CSSDNP couples these reduced-context predictions with a full-context teacher prediction. To compensate for the information removed by context reduction, an Exponential Moving Average [EMA; 33] teacher predicts from the full context of the same task and provides a self-distillation target [47]. This creates a context-asymmetric learning signal: the student learns under partial context while being guided by a stronger full-context predictive distribution. Since changing the context set changes the conditional predictive distribution in NPs, CSSDNP aligns reduced-context predictions with the richer full-context prediction of the same underlying task. Theoretically, we show that context-subset training learns the reduced-context prediction problem while providing multiple context views, and that full-context self-distillation yields an asymptotically unbiased target under a calibrated teacher and improves finite-sample estimation through teacher-guided shrinkage. Empirically, CSSDNP improves generalization across multiple NP variants and tasks, especially in the low-meta-data regime.

The contributions of this work are as follows:

- We identify and characterize generalization collapse in NPs under the low-meta-data regime.
- We propose CSSDNP, a context-asymmetric self-distillation framework that regularizes a reduced-context student toward a full-context EMA teacher from the same task.
- We theoretically show that context-subset training induces multiple reduced-context views of each task, and that full-context self-distillation provides a principled reduced-context target with no asymptotic target bias under a calibrated teacher, while improving finite-sample estimation.
- We empirically demonstrate that CSSDNP improves robustness and generalization across diverse neural process models and tasks.

2 Preliminaries

2.1 Problem setup

Meta-learning [20] aims to learn a shared predictive mechanism from a collection of related tasks, so that a model can adapt efficiently to new tasks. We consider a meta-dataset $\mathcal{D}_{\text{meta}} = \{D_m\}_{m=1}^{M}$, where each task D_m is sampled i.i.d. from an underlying task distribution p_τ, and M denotes the number of available meta-training tasks. Following recent general neural process settings [25, 24], we allow tasks to have different input and output dimensionalities. Task m has input space $\mathcal{X}_{d_x(m)} \subseteq \mathbb{R}^{d_x(m)}$ and output space $\mathcal{Y}_{d_y(m)} \subseteq \mathbb{R}^{d_y(m)}$, where $d_x, d_y : \mathbb{N} \to \mathbb{N}$. Its dataset is

$$D_m = \{(x_{m,i}, y_{m,i})\}_{i=1}^{n_m}, \qquad (x_{m,i}, y_{m,i}) \in \mathcal{X}_{d_x(m)} \times \mathcal{Y}_{d_y(m)}.$$

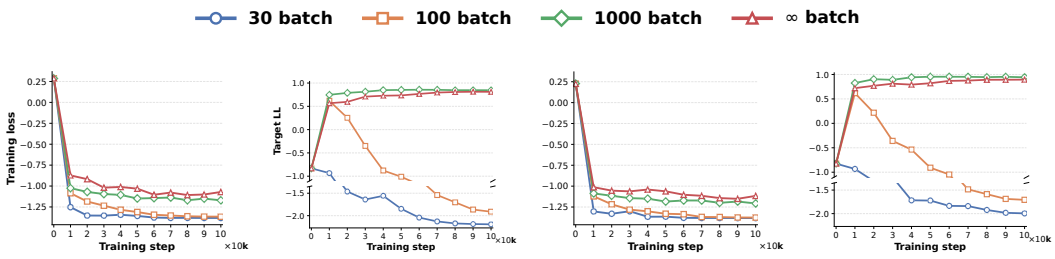
Each task is split into a context set and a target set,

$$D_m = D_m^{(c)} \cup D_m^{(t)}, \qquad D_m^{(c)} = \{(x_{m,i}^{(c)}, y_{m,i}^{(c)})\}_{i=1}^{n_m^c}, \quad D_m^{(t)} = \{(x_{m,j}^{(t)}, y_{m,j}^{(t)})\}_{j=1}^{n_m^t},$$

with $n_m = n_m^c + n_m^t$. The context set provides conditioning information, and the target set is used for prediction and training. The goal is to learn a predictive rule that, given the context set of a new task, produces accurate and uncertainty-aware predictions on its target points.

2.2 Neural processes

Neural processes [NPs; 11, 12] are probabilistic meta-learning models that define predictive distributions over target outputs conditioned on context sets. Given $D_m^{(c)}$, an NP predicts $q_\theta\left(y_m^{(t)} \mid x_m^{(t)}, D_m^{(c)}\right)$,



(a) CANP: Training loss (b) CANP: Target LL (c) TNP: Training loss (d) TNP: Target LL

Figure 1: Loss and target log-likelihood in the low-meta-data regime on RBF GP regression with CANP and TNP.

or more generally, a joint predictive distribution $q_\theta\left(y^{(t)}_{m,1:n^t_m} \mid x^{(t)}_{m,1:n^t_m}, D^{(c)}_m\right)$, where θ denotes the model parameters.

A useful probabilistic view is that each task is generated by an underlying task-specific function f_m. Under this view [23, 25, 24], predictive distribution can be written as follows:

$$p(Y_m \mid X_m, D^{(c)}_m) = \int \left[\prod_{i=1}^{n_m} p(y_{m,i} \mid f_m, x_{m,i})\right] p(f_m \mid D^{(c)}_m)\, df_m,$$

where X_m and Y_m denote the collections of inputs and outputs for task m. In practice, NPs approximate this conditional distribution with neural encoders and decoders. Many NP variants introduce a task-level latent variable $z_m \in \mathbb{R}^{d_z}$ to summarize the task function [12, 22, 9]:

$$q_\theta\left(y^{(t)}_m \mid x^{(t)}_m, D^{(c)}_m\right) = \int q_\theta\left(y^{(t)}_m \mid x^{(t)}_m, z_m\right) q_\theta\left(z_m \mid D^{(c)}_m\right) dz_m.$$

NP variants mainly differ in how they encode the context set $D^{(c)}_m$ into z_m. Conditional neural processes [CNPs; 11, 22, 16, 35] use deterministic encoders to produce a point representation of the task, whereas latent neural processes [(latent) NPs; 12, 22, 9] place a distribution over z_m to model functional uncertainty more explicitly. In both cases, a decoder maps the target input and task representation to a predictive distribution. CNPs are typically trained by maximizing predictive log-likelihood over meta-training tasks. Latent NPs are commonly trained by maximizing either predictive log-likelihood or an evidence lower bound (ELBO), such as

$$\mathbb{E}_{D_m}\left[\sum_{j=1}^{n_m} \mathbb{E}_{q_\theta(z_m \mid D_m)}\left[\log p_\theta(y_{m,j} \mid x_{m,j}, z_m)\right] - \mathrm{KL}\left(q_\theta(z_m \mid D_m) \,\middle\|\, q_\theta(z_m \mid D^{(c)}_m)\right)\right].$$

These objectives form the basis of NP training, but they can become fragile when the number of meta-training tasks is limited. In the next section, we show that likelihood maximization or ELBO alone can lead to overfitting to observed tasks and degraded generalization. Our framework addresses this failure while remaining applicable to both deterministic and latent NP variants.

3 Observation: Generalization collapse in the low-meta-data regime

We now describe our empirical observation that motivates our method. We call a setting *low-meta-data* when the number of meta-training tasks M is small enough that task diversity is severely limited. In our experiments, this typically corresponds to M on the order of 10^2 to 10^3, depending on the benchmark and task family.

Our central observation is that NPs exhibit a characteristic failure pattern in this regime. When M is sufficiently large, e.g., on the order of 10^4 or more, standard NP training behaves as expected: optimization improves performance and the learned predictive process generalizes well to unseen tasks. As M decreases, however, test performance may initially improve but eventually deteriorates sharply. In the most data-scarce settings, generalization can even begin to degrade almost immediately after training starts, indicating that the model rapidly specializes to the limited set of observed tasks

rather than learning transferable structure. This failure is not merely sample-level overfitting within each task, but rather *task-level memorization*. When only a small number of tasks is available, the empirical task distribution becomes a poor approximation to the underlying task distribution. Consequently, the learned predictive process specializes to the finite set of observed task functions instead of capturing transferable predictive structure across the broader task family.

Figure 1 illustrates this phenomenon on Gaussian process (GP) regression with the RBF kernel using Conditional Attentive Neural Processes [CANP; 22] and Transformer Neural Processes [TNP; 35]. Each training batch contains 16 tasks, and we vary the total number of training batches from 30 to 1000. The ∞ setting denotes online training, where each batch is freshly sampled from the data generator and used only once. With sufficient meta-training data, both models improve steadily and generalize well. In contrast, when the number of tasks is small, training loss keeps decreasing while target log-likelihood on unseen tasks deteriorates substantially. Thus, in the low-meta-data regime, optimizing the standard NP objective can improve training fit while simultaneously harming generalization. These observations motivate a training strategy beyond stronger generic regularization. In particular, we seek a method that improves robustness under limited context information while preserving the richer predictive structure available from the full observed context.

4 Context-Subset Self-Distillation Neural Processes

4.1 Overview

As shown in Section 3, standard NP training can memorize observed meta-training tasks in the low-meta-data regime, degrading generalization to unseen tasks. In order to address this issue, we introduce *Context-Subset Self-Distillation Neural Processes* (CSSDNP), a training framework that stabilizes NP learning and improves predictive distributions without changing the architecture or adding test-time cost. CSSDNP consists of two components. First, the student is trained on random subsets of the original context, inducing multiple reduced-context views of each task and thereby augmenting the meta-training signal. Second, a self-distillation regularizer guides the student toward a more informative prediction from an EMA teacher conditioned on the full context of the same task, enforcing conditional predictive consistency without changing the NP architecture or test-time procedure. Together, these components encourage the student to learn robust predictive structure under partial context while remaining consistent with the richer full-context prediction.

4.2 Context-subset training

When the number of meta-training tasks M is small, neural processes can overfit the finite set of observed tasks instead of learning the underlying predictive process. A useful training strategy should therefore increase the effective meta-training signal while keeping the prediction problem sufficiently challenging to discourage memorization.

CSSDNP achieves this by exploiting the context–target structure of each task. For each meta-training task D_m, we construct an augmented context set called *refined context* $D_m^{(r)} \subseteq D_m^{(c)}$ by first sampling its size n_m^r uniformly from $\{a, \dots, b\}$, where a and b are hyperparameters, and then sampling n_m^r points from $D_m^{(c)}$ without replacement. Since $D_m^{(r)}$, $D_m^{(c)}$, and $D_m^{(t)}$ are generated from the same task-specific function, the model can still learn a valid reduced-context predictive distribution,

$$p_{\mathrm{std}}(x_m; D_m^{(r)}) = q_\theta\Big(\cdot \mid x_m, D_m^{(r)}\Big).$$

At each training step, we sample a refined context $D_m^{(r)}$ and train the model to predict the task data conditioned only on this reduced context. Different subset samples induce different conditional views of the same task, effectively augmenting the meta-training signal. At the same time, because the model must predict under less conditioning information, the task becomes harder, which discourages direct task memorization and promotes a more transferable predictive mechanism. Unlike standard context-target resampling in NP training, CSSDNP further couples the reduced-context student with a full-context teacher within the same training episode, as described in the next subsection.

For CNPs, the context-subset objective is

$$\mathcal{L}_{\text{sub}}(\theta; D_m^{(r)}) = -\sum_{j=1}^{n_m} \log q_\theta(y_{m,j} \mid x_{m,j}, D_m^{(r)}).$$

For latent NP variants, we use the corresponding ELBO under the refined context:

$$\mathcal{L}_{\text{sub}}(\theta; D_m^{(r)}) = -\sum_{j=1}^{n_m} \mathbb{E}_{q_\theta(z_m \mid D_m)} \left[\log p_\theta(y_{m,j} \mid x_{m,j}, z_m)\right] + \text{KL}\Big(q_\theta(z_m \mid D_m) \,\Big\|\, q_\theta(z_m \mid D_m^{(r)})\Big).$$

This training strategy has two theoretical interpretations. First, training on random context subsets corresponds to learning the Bayes-optimal predictor for the reduced-context prediction problem induced by the subset distribution. Second, because each task can generate many refined contexts, context-subset training induces an augmented task distribution over multiple conditional views of the same underlying task. However, since refined contexts discard information from the full context, the induced prediction problem is generally harder than full-context prediction. We summarize these properties below and defer formal statements and proofs to Section B.1.

Proposition 1 (Informal)**.** *Training on random context subsets minimizes the population predictive log-loss under the refined context, thereby learning the Bayes-optimal reduced-context predictor.*

Proposition 2 (Informal)**.** *Context-subset training can be viewed as learning over an augmented task distribution and reducing the context generally makes the induced prediction problem harder than full-context prediction.*

4.3 Full-context self-distillation

Context-subset training increases the effective meta-training signal by inducing multiple reduced-context views, but it also removes information from the original context and can make prediction unnecessarily difficult. To remedy this issue, we introduce a full-context self-distillation objective that trains the student under reduced context while guiding it toward a stronger predictive target.

Let θ and ϕ denote the student and teacher parameters, respectively. Then, the teacher is updated as an EMA of the student during training as follows:

$$\phi \leftarrow \tau\phi + (1-\tau)\theta,$$

where $\tau \in (0, 1)$ is a momentum coefficient close to 1. The EMA teacher provides a temporally smoothed and more stable predictive distribution during training. For each task, the student conditions on the refined context $D_m^{(r)} \subseteq D_m^{(c)}$, whereas the teacher conditions on the full context $D_m^{(f)} := D_m^{(c)}$. Thus, for an input x, the teacher prediction is $q_\phi(\cdot \mid x, D_m^{(f)})$, while the corresponding student prediction is $q_\theta(\cdot \mid x, D_m^{(r)})$. This creates a deliberate information asymmetry: the student must predict from a partial context, but is regularized toward the richer predictive distribution available from the full context of the same task. This resembles JEPA-style [2, 3] predictive self-supervision in spirit, but here the target is a full-context predictive distribution, which is different from previous JEPA [2, 3] or self-distillation [17] methods. Thus, the distillation objective is formulated as follows:

$$\mathcal{L}_{\text{dist}} = \mathbb{E}_{D_m \sim \mathcal{D}_{\text{meta}}} \left[\sum_{j=1}^{n_m} \text{KL}\Big(q_\phi(\cdot \mid x_{m,j}, D_m^{(f)}) \,\Big\|\, q_\theta(\cdot \mid x_{m,j}, D_m^{(r)})\Big)\right].$$

We use the *forward* Kullback-Liebler (KL) divergence from the full-context teacher to the reduced-context student. This direction is important: since the teacher observes more context, it provides a stronger posterior target, and minimizing the forward KL encourages the student to cover the support of the teacher predictive distribution rather than collapsing onto a single high-density region. More importantly, forward KL induces an arithmetic mixture of full-context teacher predictions over the missing context, which recovers the Bayes-optimal reduced-context predictor under a calibrated teacher. This is particularly desirable for neural processes, where uncertainty modeling is central. For latent NP variants, whose predictive distributions can be mixture-like or multimodal after marginalizing latent variables, forward KL promotes mode coverage and helps preserve predictive uncertainty.

4.3.1 MC approximation for latent neural processes

For latent NPs, the predictive distributions of the teacher and the students are obtained as follows:

$$q_\phi(y_m \mid x_m, D_m^{(f)}) = \int q_\phi(y_m \mid x_m, z) q_\phi(z \mid D_m^{(f)})\, dz,$$
$$q_\theta(y_m \mid x_m, D_m^{(r)}) = \int q_\theta(y_m \mid x_m, z) q_\theta(z \mid D_m^{(r)})\, dz.$$

Since the KL between these marginal predictive distributions is generally not available in closed form, we approximate the forward KL by Monte Carlo (MC) samples from the teacher predictive distribution. Let

$$\tilde{y}_m^{(1)}, \ldots, \tilde{y}_m^{(K)} \sim q_\phi(\cdot \mid x_m, D_m^{(f)}).$$

Then, up to an additive constant independent of θ,

$$\mathrm{KL}\Big(q_\phi(\cdot \mid x_m, D_m^{(f)}) \,\Big\|\, q_\theta(\cdot \mid x_m, D_m^{(r)})\Big) \approx -\frac{1}{K}\sum_{k=1}^{K} \log q_\theta(\tilde{y}_m^{(k)} \mid x_m, D_m^{(r)}).$$

Thus, the student is trained to assign high likelihood to samples from the full-context teacher. When the teacher predictive distribution contains multiple plausible modes, this sampling-based forward-KL objective encourages the student to cover them rather than explain only a single representative mode, leading to more faithful uncertainty transfer and more stable variance prediction.

Theoretically, full-context self-distillation has three key properties. First, under forward KL, it induces a principled reduced-context target given by the conditional average of the full-context teacher over the missing context. Second, if the teacher is calibrated, this target coincides with the Bayes-optimal reduced-context predictor. Third, in the finite-meta-data regime, it acts as a teacher-guided shrinkage mechanism [21, 32] that can reduce estimation variance. This can also be interpreted as providing richer soft supervision than standard target-sample likelihood training, potentially reducing gradient variance under a calibrated teacher. We summarize these properties below and provide formal statements and proofs in Section B.2.

Proposition 3 (Informal)**.** *Under suitable assumptions, full-context self-distillation induces a principled reduced-context target; if the teacher is calibrated, this target coincides with the Bayes-optimal reduced-context predictor.*

Proposition 4 (Informal)**.** *Under a local strongly convex approximation, full-context self-distillation acts as a teacher-guided shrinkage mechanism, reducing finite-sample estimation variance.*

The final CSSDNP objective is

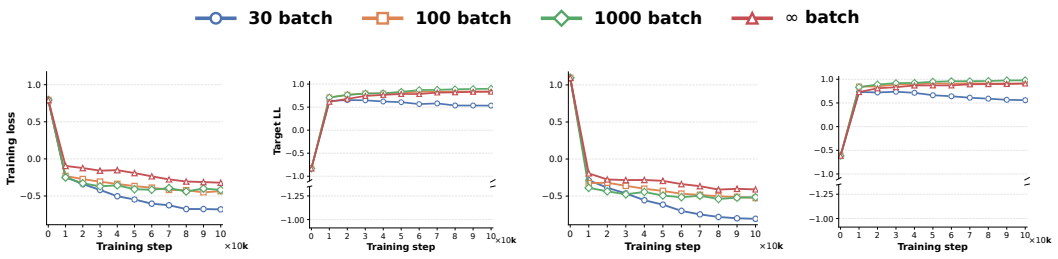
$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{sub}} + \lambda_{\mathrm{dist}} \mathcal{L}_{\mathrm{dist}},$$

where $\lambda_{\mathrm{dist}} \geq 0$ controls the strength of distillation. For latent NP variants, $\mathcal{L}_{\mathrm{dist}}$ is implemented using the MC approximation above. For the overall training and inference refer to Appendix C.2.

5 Experiments

5.1 Settings

We empirically evaluate CSSDNP in diverse low-meta-data regimes, focusing on regression tasks. We consider representative deterministic and latent NP variants: CANP [22] and TNP [35] as CNP-style models, and Attentive Neural Processes [ANP; 22] and Dimension Agnostic Neural Processes [DANP; 25] as latent NP models. Since CSSDNP is a training framework rather than a new architecture, we denote its combination with a base model A by CSSDA; for example, CSSDCANP denotes CANP trained with CSSDNP. Unless otherwise specified, all latent NP methods use 50 latent-variable samples for fair comparison. For every method, we report the test performance of the **checkpoint selected by the best validation target log-likelihood**, ensuring that the gains are not due to delayed overfitting or different early-stopping protocols. In all tables, the best metric is shown in **bold**, and results are averaged over five random seeds with 1-sigma error bars. Dataset and architecture details are provided in Section C.



(a) CSSDCANP: loss (b) CSSDCANP: target LL (c) CSSDTNP: loss (d) CSSDTNP: target LL

Figure 2: Training dynamics under CSSDNP on the same RBF GP regression as Figure 1. Unlike standard training, CSSDNP maintains stable target log-likelihood even with low-meta-data regime.

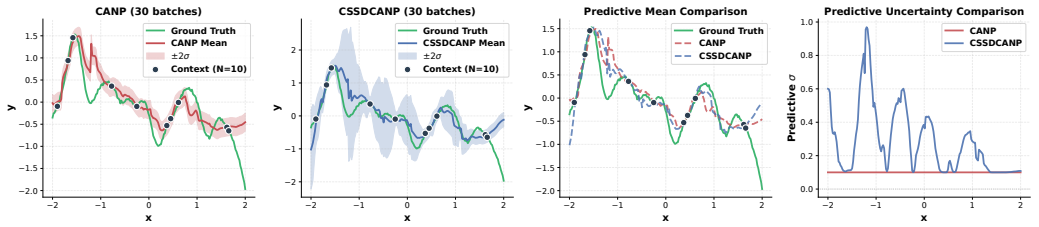


Figure 3: (a,b) Predictive mean and variance of CANP and CSSDCANP in the low-meta-data regime. (c,d) Predictive mean and uncertainty comparison.

5.2 Observation revisiting

We revisit the observation study in Section 3, changing only the training method to CSSDNP. As shown in Figure 2, CSSDCANP and CSSDTNP optimize stably across all meta-training sizes, indicating that random context subsetting and self-distillation do not destabilize training. More importantly, CSSDNP substantially remedies the overfitting observed under standard training. While the standard training already collapses in the 100-batch setting, CSSDNP maintains stable predictive performance even with only 30 batches. With 100 batches, CSSDNP already reaches performance comparable to training with 1000 batches, which would require substantially more meta-training data under standard training.

Figure 3 further shows that CSSDNP improves the learned predictive distribution, not only the mean prediction. Under standard training, CANP's predictive variance collapses toward the minimum value imposed by the model parameterization, 0.1, producing overconfident predictions across the input domain. In contrast, CSSDCANP assigns small variance near dense context regions and larger variance where context information is sparse, yielding predictive intervals that better cover the ground-truth function. This suggests that the degradation of standard training is largely driven by uncertainty collapse: although its mean can still capture the rough function shape, it severely underestimates variance and suffers from poor log-likelihood. CSSDNP preserves meaningful uncertainty and thereby maintains robust predictive performance in the low-meta-data regime. Refer to Appendix D.1 to see the additional results for the other CSSDNP architectures.

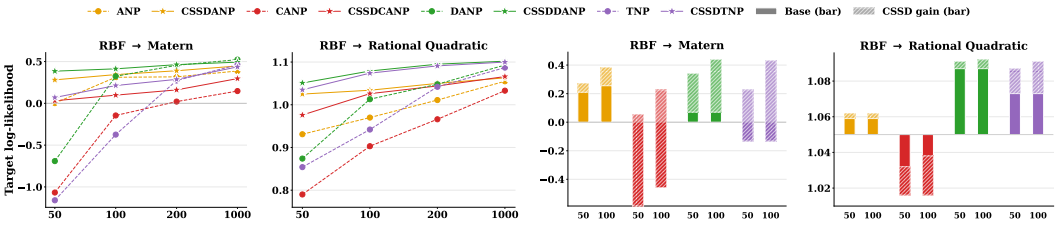
5.3 Gaussian Processes regression

In this section, we evaluate CSSDNP on three GP regression settings: (i) **various kernels**, where we compare NP models under enough- and low-meta-training data; (ii) **domain shift**, where models trained on RBF kernels are evaluated zero-shot on other kernels; and (iii) **few-shot finetuning**, where an RBF-pretrained model is adapted to a new kernel using only a small number of finetuning batches.

Various kernels. Table 1 reports target log-likelihoods across four NP models and three GP kernels. When meta-training data is sufficient ($N = 1000$), CSSDNP performs comparably to or better than the base models, showing that it does not compromise performance in the data-rich regime. In the low-meta-train setting ($N = 50$), CSSDNP consistently outperforms all base models across architectures

Table 1: Target log-likelihoods on 1D GP regression across kernels and meta-training sizes.

Model	Enough-meta-train ($N_b = 1000$)			Low-meta-train ($N_b = 50$)		
	RBF	Matern	Rat.Quad.	RBF	Matern	Rat.Quad.
CANP	0.871±0.003	0.162±0.005	1.032±0.005	0.503±0.010	-0.888±0.021	0.723±0.018
CSSDCANP	**0.901** ±0.002	**0.241**±0.003	**1.060**±0.003	**0.747** ±0.008	**-0.017** ±0.013	**0.962**±0.011
TNP	0.963 ±0.002	**0.458**±0.004	1.088±0.004	0.401 ±0.009	-0.569±0.017	0.666±0.017
CSSDTNP	**0.972**±0.003	0.448±0.003	**1.099**±0.003	**0.849**±0.011	**0.124**±0.020	**1.029**±0.019
ANP	0.908 ±0.005	0.365±0.007	1.055±0.003	0.629 ±0.011	0.027 ±0.025	0.896±0.021
CSSDANP	**0.919**±0.004	**0.373**±0.005	**1.063**±0.005	**0.827** ±0.010	**0.208** ±0.022	**0.997**±0.018
DANP	0.972 ±0.003	0.514±0.003	1.094±0.003	0.548±0.009	-0.518 ±0.0018	0.814±0.018
CSSDDANP	**0.987**±0.004	**0.516** ±0.003	**1.109**±0.004	**0.868**±0.010	**0.200**±0.015	**1.035**±0.015



(a) DS: RBF → Matern (b) DS: RBF → RQ (c) FT: RBF → Matern (d) FT: RBF → RQ

Figure 4: Domain shift and few-shot finetuning on 1D GP regression. (a,b) Zero-shot transfer from RBF-trained models ($N_b \in \{50, 100, 200, 1000\}$) to Matern and RQ kernels. (c,d) Few-shot finetuning from an RBF-pretrained model to target kernels using $N_b \in \{50, 100\}$ adaptation batches. Solid bars denote base performance, and hatched extensions denote the additional gain from CSSDNP.

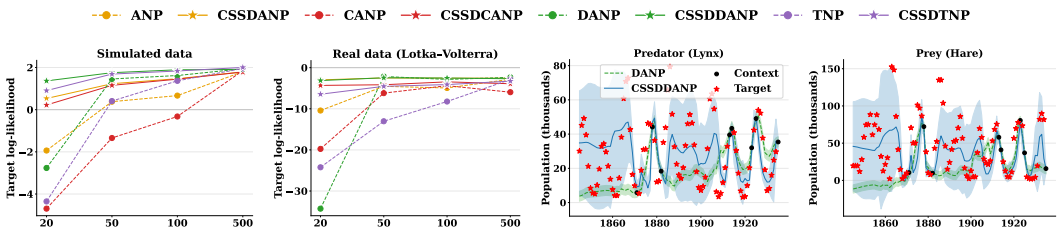
and kernels, with especially large gains on the Matern kernel where standard training suffers severe collapse. These results show that CSSDNP effectively mitigates task-level overfitting in the low-meta-data regime. Refer to Appendix D.2 for comparison with additional **generic regularization and consistency baselines** (e.g., Dropout [18], R-drop [44]).

Domain shift. To assess out-of-distribution robustness, we train models on RBF processes and evaluate them zero-shot on Matérn and Rational Quadratic (RQ) kernels across $N_b \in \{50, 100, 200, 1000\}$. As shown in Figure 4(a), base models can fail catastrophically under kernel mismatch in the low-data regime; for example, CANP and TNP obtain strongly negative target log-likelihoods at $N_b = 50$. CSSDNP consistently prevents this failure across all four architectures, with every CSSD variant achieving positive likelihoods even at $N = 50$. On the closer RQ shift in Figure 4(b), CSSD variants also uniformly outperform their base counterparts across all N, indicating that the benefit is not merely a rescue effect in the most extreme regime but a consistent improvement in transfer robustness.

Few-shot finetuning. We next test whether CSSDNP provides a stronger adaptation procedure under domain shift. Starting from an RBF-pretrained model ($N_b = 1000$), we finetune on the target kernel using only $N_b \in \{50, 100\}$ batches. In Figure 4(c,d), hatched regions show the additional gain from CSSDNP finetuning over standard finetuning. The improvement is most pronounced on Matérn: standard finetuning of CANP and TNP still fails to achieve positive log-likelihood even with $N = 100$, whereas their CSSDNP counterparts significantly recover performance. On RQ, CSSDNP also yields consistent gains across models. Overall, CSSDNP improves both zero-shot transfer and few-shot adaptation, with the largest benefits appearing when the base training procedure struggles most. Refer to Appendices D.4 and D.10 for additional **uncertainty metrics** (e.g., CRPS, Cov@95, RMSE) and **multidimensional tasks**, which show the same trend and confirm that the gains are not specific to log-likelihood or 1D regression.

5.4 Sim-to-Real Transfer

We evaluate Sim-to-Real generalization on the Lotka–Volterra predator–prey system [28], where models are meta-trained only on ODE-generated trajectories and tested on real Hudson Bay population records. Figure 5(a,b) shows a clear gap between CSSD variants and their base models. On simulated



(a) Simulation LL (b) Real LL (c) Prediction: Predator (d) Prediction: Prey

Figure 5: Sim-to-Real transfer on the Lotka–Volterra predator–prey system. **(a,b)** Target log-likelihood on simulated and real data as a function of meta-training size $N_b \in \{20, 50, 100, 500\}$. **(c,d)** Predictive distributions at N_b=20 on real data: DANP (green, dashed) produces overconfident narrow bands, whereas CSSDDANP (blue, solid) is better calibrated and covers more observations.

Table 2: Ablation study on 1D GP regression (RBF kernel, N=100).

(a) Component ablation

Model	r	λ	Target log-lik. ↑
CANP (base)	1.0	0	0.604 ± 0.008
+ Subsamp	0.5	0	0.802 ± 0.010
+ Distill	1.0	0.5	0.754 ± 0.008
+ CSSD	0.5	0.5	$\mathbf{0.844} \pm 0.006$

(b) Distill weight λ (r=0.5)

λ	Target log-lik. ↑
0.05	0.824 ± 0.004
0.1	0.825 ± 0.003
0.5	$\mathbf{0.844} \pm 0.006$
1.0	0.831 ± 0.004
3.0	0.769 ± 0.010

(c) Context ratio r (λ=0.5)

r	Target log-lik. ↑
0.3	0.830 ± 0.005
0.5	$\mathbf{0.844} \pm 0.006$
0.7	0.824 ± 0.004
0.9	0.815 ± 0.005
1.0	0.754 ± 0.009

data, both perform similarly when N is large, but on real data CSSD yields substantial gains across all meta-training sizes. In particular, base models collapse severely at small N_b, often producing log-likelihoods far below zero, whereas CSSD variants remain much more stable. Although the base models gradually recover as N increases, CSSD consistently retains a clear advantage, indicating that it learns more transferable representations rather than overfitting to simulated trajectory patterns.

Figure 5(c,d) provides a qualitative example at $N_b = 20$. Given the same context points, CSSDDANP tracks the true population dynamics more accurately than DANP and yields better-calibrated uncertainty. While DANP produces overconfident narrow intervals that miss many observations, CSSDDANP outputs wider, better-calibrated predictive bands that more consistently cover the target points. This suggests that CSSD encourages appropriate uncertainty under distribution shift, which is crucial for deploying simulation-trained models in real-world settings. Refer to Appendix D.11 and D.12 to see further experiments on **image completion** and **climate prediction tasks**. The overall trends are the same for the other tasks.

5.5 Ablation

We ablate CSSD on 1D GP regression with the RBF kernel. Here, r denotes the student context ratio, i.e., the fraction of the original context given to the student, and λ denotes the distillation weight.

Component analysis. Table 2(a) verifies that both components of CSSD are useful. Subsampling alone (r=0.5, λ=0) and distillation alone (r=1.0, λ=0.5) both improve over the CANP baseline, while their combination achieves the best performance. This confirms that the two components play complementary roles: context subsampling creates an information-asymmetric prediction problem that discourages memorization, whereas distillation transfers the richer full-context teacher prediction to the reduced-context student.

Effect of distillation weight. Table 2(b) studies the effect of λ with r=0.5. CSSD remains strong across a broad range of distillation weights and degrades mainly when the teacher signal becomes too weak or too dominant. This indicates that CSSD is not highly sensitive to careful tuning of λ, making it practical in low-meta-data settings.

Table 3: Additional ablations on 1D GP regression. (a) Complementarity with fresh context-target resampling at $N_b = 50$. (b) Comparison of distillation divergences.

(a) Fresh context-target resampling			(b) Distillation divergence			
Method	RBF	Matern	Model	Forward KL	Reverse KL	Symmetric KL
Base	0.693	-0.167	CANP	**0.747**	0.539	0.645
Subset	0.805	-0.069	TNPD	**0.849**	0.678	0.766
Distill	0.775	-0.115	ANP	**0.827**	0.579	0.714
Both (CSSD)	**0.835**	**0.059**	DANP	**0.868**	0.744	0.762

Effect of student context ratio. Table 2(c) varies the context ratio r with λ=0.5. CSSD performs best at an intermediate ratio and remains effective over a wide range of reduced contexts, supporting the importance of a non-trivial but learnable information asymmetry between student and teacher. Performance drops when r approaches 1.0, where the student and teacher observe nearly the same context and distillation provides less useful supervision.

Complementarity with context-target resampling. Standard NP training commonly resamples context-target partitions across training episodes. To examine whether CSSDNP provides benefits beyond this practice, we evaluate its individual components under fresh context-target resampling on RBF and Matérn GP regression at $N_b = 50$. As shown in Table 3(a), standard resampling improves the baseline performance, while CSSDNP provides further gains of 0.142 and 0.226 on RBF and Matérn, respectively. Moreover, CSSDNP outperforms both context subsetting and distillation alone. These results demonstrate that CSSDNP complements standard episode-level augmentation by introducing nested context-asymmetric predictive supervision rather than relying on fixed context-target partitions.

Effect of distillation divergence. We further compare forward KL with reverse and symmetric KL under the same RBF-GP training protocol across four NP architectures. As shown in Table 3(b), forward KL consistently achieves the best performance across all architectures, outperforming reverse KL by 0.124–0.248 and symmetric KL by 0.083–0.113 in target log-likelihood. This supports our theoretical motivation that forward KL induces an arithmetic mixture of full-context teacher predictions, recovering the Bayes-optimal reduced-context predictor under a calibrated teacher, whereas reverse KL favors mode-seeking aggregation. The consistent improvements across deterministic and latent NP variants further demonstrate that the choice of forward KL is not backbone-specific. Additional ablations on the **effect of EMA**, **EMA momentum**, and the **number of distillation Monte Carlo samples** are provided in Appendix D.8.

6 Conclusion

We studied generalization collapse in neural processes under scarce meta-training tasks, where standard objectives induce task-level memorization and degrade performance on unseen tasks. To address this, we proposed *Context-Subset Self-Distillation Neural Processes* (CSSDNP), a training framework that combines context-subset training with self-distillation from a full-context EMA teacher. Context-subset training exposes the model to multiple reduced-context views of each task, increasing effective meta-training diversity and discouraging memorization. Full-context self-distillation compensates for the resulting information loss by guiding the student toward a stronger predictive target; theoretically, it preserves the asymptotic Bayes-optimal reduced-context predictor under a calibrated teacher and improves finite-sample estimation through teacher-guided shrinkage. CSSDNP is architecture-agnostic and adds no inference cost over the base model. Empirically, CSSDNP achieves stable training and robust performance across diverse models and settings, with the largest gains in the low-meta-data regime where standard training fails most severely. As a limitation, Appendix D.3 shows that under extremely scarce settings, such as one-batch training, CSSDNP still improves over standard training but cannot fully prevent generalization failure. Extending CSSDNP to such extreme meta-data regimes is an important direction for future work.

Acknowldegments This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25480825), the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Leading Generative AI Human Resources Development program (IITP-2024-RS-2024-00397085) funded by the Korea government (MSIT), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02219317, AI Star Fellowship (Kookmin University)), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-23524783).

References

[1] Matthew Ashman, Cristiana Diaconu, Junhyuck Kim, Lakee Sivaraya, Stratis Markou, James Requeima, Wessel P Bruinsma, and Richard E Turner. Translation equivariant transformer neural processes. *arXiv preprint arXiv:2406.12409*, 2024.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[4] Wessel P Bruinsma, James Requeima, Andrew YK Foong, Jonathan Gordon, and Richard E Turner. The gaussian neural process. *arXiv preprint arXiv:2101.03606*, 2021.

[5] Wessel P Bruinsma, Stratis Markou, James Requiema, Andrew YK Foong, Tom R Andersson, Anna Vaughan, Anthony Buonomo, J Scott Hosking, and Richard E Turner. Autoregressive conditional neural processes. *arXiv preprint arXiv:2303.14468*, 2023.

[6] Leo Feng, Hossein Hajimirsadeghi, Yoshua Bengio, and Mohamed Osama Ahmed. Latent bottlenecked attentive neural processes. *arXiv preprint arXiv:2211.08458*, 2022.

[7] Leo Feng, Frederick Tung, Hossein Hajimirsadeghi, Yoshua Bengio, and Mohamed Osama Ahmed. Memory efficient neural processes via constant memory attention block. *arXiv preprint arXiv:2305.14567*, 2023.

[8] Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *arXiv preprint arXiv:2103.15671*, 2021.

[9] A. Y. K. Foong, W. P. Bruinsma, J. Gordon, Y. Dubois, J. Requeima, and R. E. Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. In *Conference on Neural Information Processing Systems*, 2020.

[10] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018.

[11] M. Garnelo, D. Rosenbaum, C. J. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. M. A. Eslami. Conditional neural processes. In *International Conference on Machine Learning*, 2018.

[12] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Y. W. Teh. Neural processes. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[13] Xiaoqing Geng, Xiwen Chen, Kenny Q Zhu, Libin Shen, and Yinggong Zhao. Mick: A meta-learning framework for few-shot relation classification with small training data. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 415–424, 2020.

[14] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*, 81(25):2340–2361, 1977.

[15] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[16] J. Gordon, W. P. Bruinsma, A. Y. K. Foong, J. Requeima, Y. Dubois, and R. E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[19] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

[20] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5149–5169, 2021.

[21] William James, Charles Stein, et al. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379. University of California Press, 1961.

[22] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, S. M. A. Eslami, D. Rosenbaum, and V. Oriol. Attentive neural processes. In *International Conference on Learning Representations*, 2018.

[23] Hyungi Lee, Eunggu Yun, Giung Nam, Edwin Fong, and Juho Lee. Martingale posterior neural processes. In *International Conference on Learning Representations*, 2023.

[24] Hyungi Lee, Moonseok Choi, Hyunsu Kim, Kyunghyun Cho, Rajesh Ranganath, and Juho Lee. Test time scaling for neural processes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[25] Hyungi Lee, Chaeyun Jang, Dongbok Lee, and Juho Lee. Dimension agnostic neural processes. *arXiv preprint arXiv:2502.20661*, 2025.

[26] J. Lee, Y. Lee, J. Kim, E. Yang, S. J. Hwang, and Y. W. Teh. Bootstrapping neural processes. In *Conference on Neural Information Processing Systems*, 2020.

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[28] Alfred James Lotka. *Elements of physical biology*. Williams & Wilkins, 1925.

[29] C. Louizos, X. Shi, K. Schutte, and M. Welling. The functional neural process. In *Conference on Neural Information Processing Systems*, 2019.

[30] Duncan Alexander MacLulich. *Fluctuations in the numbers of the varying hare (Lepus americanus)*, volume 189. University of Toronto Press Toronto, 1937.

[31] Stratis Markou, James Requeima, Wessel P Bruinsma, Anna Vaughan, and Richard E Turner. Practical conditional neural processes via tractable dependent predictions. *arXiv preprint arXiv:2203.08775*, 2022.

[32] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

[33] Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

[34] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

[35] Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, 2022.

[36] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

[37] Janarthanan Rajendran, Alexander Irpan, and Eric Jang. Meta-learning requires meta-augmentation. *Advances in Neural Information Processing Systems*, 33:5705–5715, 2020.

[38] Tommy Rochussen and Vincent Fortuin. Sparse gaussian neural processes. *arXiv preprint arXiv:2504.01650*, 2025.

[39] Tommy Rochussen and Vincent Fortuin. Amortising inference and meta-learning priors in neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KG6SSTz2GJ.

[40] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[41] Anna Vaughan, Will Tebbutt, J Scott Hosking, and Richard E Turner. Convolutional conditional neural processes for local climate downscaling. *Geoscientific Model Development Discussions*, 2021:1–25, 2021.

[42] Qi Wang and Herke Van Hoof. Learning expressive meta-representations with mixture of expert neural processes. *Advances in neural information processing systems*, 35:26242–26255, 2022.

[43] Qi Wang, Marco Federici, and Herke van Hoof. Bridge the inference gaps of neural processes via expectation maximization. In *The Eleventh International Conference on Learning Representations*, 2023.

[44] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: Regularized dropout for neural networks. *Advances in neural information processing systems*, 34:10890–10905, 2021.

[45] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[46] Mingzhang Yin, George Tucker, Mingyuan Zhou, Sergey Levine, and Chelsea Finn. Meta-learning without memorization. *arXiv preprint arXiv:1912.03820*, 2019.

[47] 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*, pages 3713–3722, 2019.

A Related Works

Neural Processes Neural Processes [NPs; 11, 12] provide a flexible framework for uncertainty-aware meta-learning by learning predictive distributions over functions from context–target pairs. The original Conditional Neural Process [CNP; 11] introduced an amortized encoder–decoder formulation for function regression, while Neural Processes [NP; 12] added a global latent variable and an ELBO objective to model functional uncertainty. Since then, the NP family has expanded mainly along three directions: stronger context encoders, richer predictive distributions, and improved inference.

A first line of work improves architectural inductive biases for encoding context sets. Attentive Neural Processes [ANP; 22] use attention to construct more expressive context representations, while Convolutional Neural Processes [Conv(C)NP; 16, 9] introduce convolutional deep-set modules to encode translation equivariance. Transformer Neural Processes [TNP; 35] recast uncertainty-aware meta-learning as sequence modeling with masked transformer layers, and Translation-Equivariant Transformer Neural Processes [TE-TNP; 1] further combine transformer-based modeling with translation-equivariant structure. Dimension-Agnostic Neural Processes [DANP; 25] extend this direction by introducing the Dimension Aggregator Block (DAB), allowing a single model to handle tasks with varying input and output dimensions.

A second line of work focuses on uncertainty modeling and dependent predictions. Functional Neural Processes [FNP; 29] introduce local latent variables to better capture function-level variation, and Bootstrapping Neural Processes [BNP; 26] use residual bootstrapping to mitigate model misspecification. Gaussian Neural Processes [GNP; 31] and Convolutional Gaussian Neural Processes [ConvGNP; 31] model dependencies among target predictions through tractable Gaussian predictive distributions, while Fully Convolutional Gaussian Neural Processes [FullConvGNP; 4] define convolutional Gaussian predictive processes with stronger dependent-prediction structure. Autoregressive Conditional Neural Processes [AR-CNP; 5] instead obtain non-Gaussian and dependent predictions by autoregressively deploying CNPs at test time. Other variants improve predictive structure or scalability through specialized designs, including Latent Bottlenecked Attentive Neural Processes [LBANP; 6], Mixture-of-Experts Neural Processes [MoE-NP; 42], Self-normalized Importance-weighted Neural Processes [SI-NP; 43], and Constant Memory Attentive Neural Processes [CMANP; 7]. Martingale Posterior Neural Processes [MPNP; 23] further connect NPs with martingale-posterior-based Bayesian inference [8] for more principled uncertainty estimation.

A third direction improves inference, especially for latent-path NP models. Test Time Scaling for Neural Processes [TTSNP; 24] addresses the gap between the variationally learned global latent distribution and the true latent posterior by refining latent samples with Sequential Monte Carlo at test time, improving prediction and calibration without retraining the base NP. In contrast to these architectural and inference-oriented approaches, our work studies a complementary failure mode: generalization collapse caused by scarce meta-training tasks. Rather than introducing a new NP architecture or modifying test-time inference, we propose a training framework that improves low-meta-data generalization by combining reduced-context learning with self-distillation from a stronger full-context teacher.

Several related approaches explicitly address learning with limited meta-training data. Prior-data Fitted Networks [PFNs; 34, 19] leverage synthetic datasets sampled from a specified task prior to learn amortized inference, although their effectiveness depends on the availability and suitability of the assumed data-generating process. Sparse Gaussian Neural Processes [SGNP; 38] incorporate sparse-GP structure, while Bayesian Neural Processes [BNNP; 39] use amortized Bayesian neural network inference with a meta-learned prior. These methods address limited meta-data through additional modeling assumptions or prior information, whereas CSSDNP modifies the training objective without requiring architectural changes or external task generators.

In contrast to these architectural and inference-oriented approaches, our work studies a complementary failure mode: generalization collapse caused by scarce meta-training tasks. Rather than introducing a new NP architecture or modifying test-time inference, we propose a training framework that improves low-meta-data generalization by combining reduced-context learning with self-distillation from a stronger full-context teacher.

Memorization and Meta-Augmentation in Meta-Learning Memorization in meta-learning has been studied as a failure mode in which models exploit task-specific information rather than learning

transferable adaptation mechanisms. Yin et al. [46] analyze this phenomenon and introduce an information-theoretic regularization approach to mitigate memorization. Rajendran et al. [37] further investigate meta-overfitting and propose task-distribution augmentation to improve generalization. Our observation of generalization collapse is closely related to these findings, but focuses on how task-level memorization manifests in the predictive distributions of NPs under scarce meta-training tasks. While standard NP training already employs context-target resampling to generate different conditional views of a task, CSSDNP complements this practice by coupling a reduced-context student with a full-context teacher within each training episode. Thus, our contribution lies in context-asymmetric predictive distillation rather than context resampling or meta-augmentation alone.

Self-Distillation and Consistency Regularization Self-distillation and consistency regularization are among the closest lines of work to CSSDNP, but their teacher-student structure typically focuses on soft-target transfer, augmentation invariance, or noisy self-training [17, 10, 47, 40, 45]. The classical knowledge-distillation framework transfers softened predictive distributions from a teacher to a student [17], while later self-distillation methods showed that students with the same or comparable capacity can still benefit from teacher predictions, even without a larger external model [10, 47]. Consistency-based methods such as Mean Teacher and Noisy Student further demonstrated that EMA teachers, stochastic perturbations, and noise injection can act as powerful regularizers by enforcing prediction stability under perturbed inputs [40, 45], although the effectiveness of such methods can depend strongly on the baseline and data regime [36]. CSSDNP is conceptually related to these approaches, but differs in the source and meaning of the teacher–student asymmetry. Rather than applying generic input perturbations or enforcing invariance to standard data augmentations, CSSDNP perturbs the conditioning information of a neural process: the teacher observes the full context of a task, whereas the student observes only a randomly reduced context. Therefore, the teacher prediction is not merely a softened class-probability target or a smoothed version of the student output, but a task-specific posterior-like predictive distribution induced by richer conditioning information. This distinction is important because the student must learn to predict under missing context while still matching a target that reflects the uncertainty structure available from the full context. In the low-meta-data regime, this conditional teacher–student asymmetry creates multiple reduced-context views of each task to discourage task-level memorization, while full-context distillation compensates for the information loss introduced by context reduction. Thus, CSSDNP can be viewed as a neural-process-specific form of context-asymmetric predictive distillation rather than a direct application of generic self-distillation. Crucially, because the NP predictive distribution is conditioned on the context set, reducing the context is not equivalent to adding input noise; it increases the Bayes-optimal expected log-loss by $I(Y; D^{(f)} \mid X, D^{(r)})$ (Proposition 6), highlighting the information asymmetry underlying our formulation.

B Theory

In this section, we provide a theoretical perspective on why CSSDNP can improve generalization in the low-meta-data regime. Our analysis follows the two key components of CSSDNP. First, we study why *context-subset training* can be beneficial. Second, we study why *full-context self-distillation* can be beneficial.

Random objects and probability law. Let $\tau \sim P_{\mathcal{T}}$ denote a random task drawn from the meta-distribution over tasks. For task τ, let $D^{(f)} \sim P_f(\cdot \mid \tau)$ denote the random full context and let $U \sim P_U$ denote the auxiliary randomness of the subset sampler, independent of $(\tau, D^{(f)})$. The refined context is defined by

$$D^{(r)} = S(D^{(f)}, U), \qquad D^{(r)} \subseteq D^{(f)}.$$

Let $(X, Y) \sim P_t(\cdot \mid \tau, D^{(f)})$ denote a random target input-output pair from the same task. Accordingly, all expectations below are taken with respect to the joint law

$$(\tau, D^{(f)}, U, D^{(r)}, X, Y),$$

where $D^{(r)} = S(D^{(f)}, U)$. Whenever the conditioning variable is $D^{(r)}$ only, the randomness in the missing part of the context is induced by the conditional law of $D^{(f)}$ given $D^{(r)}$.

Throughout, let

$$p_r(\cdot \mid x, d^{(r)}) := p(\cdot \mid x, d^{(r)})$$

denote the ***true reduced-context predictive distribution*** and

$$p_f(\cdot \mid x, d^{(f)}) := p(\cdot \mid x, d^{(f)})$$

the ***true full-context predictive distribution***. Let $q_\theta(\cdot \mid x, d^{(r)})$ be the student predictive distribution and $q_\phi(\cdot \mid x, d^{(f)})$ be the teacher predictive distribution. For latent neural process variants, these denote the predictive marginals after integrating out the latent variable.

We define the population subset-training objective by

$$\mathcal{L}_{\text{sub}}(\theta) := \mathbb{E}_{\tau, D^{(f)}, U, X, Y}\big[-\log q_\theta(Y \mid X, D^{(r)})\big],$$

and the population distillation objective by

$$\mathcal{L}_{\text{dist}}(\theta; \phi) := \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[\mathrm{KL}\big(q_\phi(\cdot \mid X, D^{(f)}) \,\|\, q_\theta(\cdot \mid X, D^{(r)})\big)\Big].$$

Accordingly, for a fixed teacher ϕ, the population CSSDNP objective is

$$\mathcal{J}_\lambda(\theta; \phi) := \mathcal{L}_{\text{sub}}(\theta) + \lambda \mathcal{L}_{\text{dist}}(\theta; \phi), \qquad \lambda \geq 0.$$

B.1 Why context-subset training helps

We first study the subset-training term. The main point is that training under $D^{(r)}$ is not an arbitrary surrogate: under log-loss, it directly targets the correct reduced-context Bayes predictor. At the same time, because $D^{(r)}$ contains less information than $D^{(f)}$, the resulting prediction problem is strictly harder in general.

Proposition 5 (Population target of context-subset training)**.** *For any predictive distribution $q(\cdot \mid x, d^{(r)})$,*

$$\mathcal{L}_{\text{sub}}(q) := \mathbb{E}_{\tau, D^{(f)}, U, X, Y}\big[-\log q(Y \mid X, D^{(r)})\big] = \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[H\big(p_r(\cdot \mid X, D^{(r)}), q(\cdot \mid X, D^{(r)})\big)\Big],$$

where $H(p, q) := -\int p(y) \log q(y)\, dy$ is the cross-entropy. Consequently, any population minimizer $q^\star_{\text{sub}}$ satisfies

$$q^\star_{\text{sub}}(\cdot \mid x, d^{(r)}) = p_r(\cdot \mid x, d^{(r)}) \qquad \textit{for almost every } (x, d^{(r)}),$$

and the excess risk decomposes as

$$\mathcal{L}_{\text{sub}}(q) - \mathcal{L}_{\text{sub}}(q^\star_{\text{sub}}) = \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[\mathrm{KL}\big(p_r(\cdot \mid X, D^{(r)}) \,\|\, q(\cdot \mid X, D^{(r)})\big)\Big].$$

Proof. By the tower property,

$$\mathcal{L}_{\text{sub}}(q) = \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[\mathbb{E}_{Y \mid \tau, D^{(f)}, U, X}\big[-\log q(Y \mid X, D^{(r)})\big]\Big].$$

Since $D^{(r)} = S(D^{(f)}, U)$ is measurable with respect to $(D^{(f)}, U)$, and $p_r(\cdot \mid X, D^{(r)})$ is the true conditional law of Y given $(X, D^{(r)})$, we may rewrite the inner expectation as

$$\mathbb{E}_{Y \mid X, D^{(r)}}\big[-\log q(Y \mid X, D^{(r)})\big] = -\int p_r(y \mid X, D^{(r)}) \log q(y \mid X, D^{(r)})\, dy.$$

Therefore,

$$\mathcal{L}_{\text{sub}}(q) = \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[H\big(p_r(\cdot \mid X, D^{(r)}), q(\cdot \mid X, D^{(r)})\big)\Big].$$

Now use the identity

$$H(p, q) = H(p) + \mathrm{KL}(p \| q).$$

Applying this pointwise with $p = p_r(\cdot \mid X, D^{(r)})$ and $q = q(\cdot \mid X, D^{(r)})$, we obtain

$$\mathcal{L}_{\text{sub}}(q) = \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[H\big(p_r(\cdot \mid X, D^{(r)})\big)\Big] + \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[\mathrm{KL}\big(p_r(\cdot \mid X, D^{(r)}) \,\|\, q(\cdot \mid X, D^{(r)})\big)\Big].$$

The first term is independent of q, so the minimum is attained exactly when

$$q(\cdot \mid x, d^{(r)}) = p_r(\cdot \mid x, d^{(r)})$$

almost surely. Subtracting the minimum value yields the excess-risk identity. □

Proposition 5 shows that context-subset training learns the Bayes-optimal predictor for the reduced-context prediction problem induced by random context reduction.

The next result quantifies how much more difficult this reduced-context problem can be.

Conditional mutual information. In the following proposition, we use $I(A; B \mid C)$ to denote the conditional mutual information, defined as

$$I(A; B \mid C) = H(A \mid C) - H(A \mid B, C).$$

Equivalently,

$$I(A; B \mid C) = \mathbb{E}_{A,B,C}\left[\log \frac{p(A, B \mid C)}{p(A \mid C)p(B \mid C)}\right].$$

In particular,

$$I(Y; D^{(f)} \mid X, D^{(r)}) = H(Y \mid X, D^{(r)}) - H(Y \mid X, D^{(f)}, D^{(r)}).$$

Proposition 6 (Information gap induced by context reduction)**.** *Let*

$$\mathcal{L}^{\star}_{\text{full}} := \inf_{\tilde{q}} \mathbb{E}_{\tau, D^{(f)}, X, Y}\big[-\log \tilde{q}(Y \mid X, D^{(f)})\big]$$

and

$$\mathcal{L}^{\star}_{\text{sub}} := \inf_{q} \mathbb{E}_{\tau, D^{(f)}, U, X, Y}\big[-\log q(Y \mid X, D^{(r)})\big].$$

Then

$$\mathcal{L}^{\star}_{\text{sub}} - \mathcal{L}^{\star}_{\text{full}} = I\Big(Y; D^{(f)} \mid X, D^{(r)}\Big) \geq 0.$$

Equivalently,

$$\mathcal{L}^{\star}_{\text{sub}} = \mathcal{L}^{\star}_{\text{full}} + I\Big(Y; D^{(f)} \mid X, D^{(r)}\Big).$$

Proof. By Proposition 5,

$$\mathcal{L}^{\star}_{\text{sub}} = \mathbb{E}_{X, D^{(r)}, Y}\big[-\log p_r(Y \mid X, D^{(r)})\big] = H(Y \mid X, D^{(r)}).$$

Similarly,

$$\mathcal{L}^{\star}_{\text{full}} = \mathbb{E}_{X, D^{(f)}, Y}\big[-\log p_f(Y \mid X, D^{(f)})\big] = H(Y \mid X, D^{(f)}).$$

Since $D^{(r)}$ is a measurable function of $D^{(f)}$, conditioning on $(X, D^{(f)}, D^{(r)})$ is equivalent to conditioning on $(X, D^{(f)})$. Hence

$$H(Y \mid X, D^{(f)}) = H(Y \mid X, D^{(f)}, D^{(r)}).$$

Therefore,

$$\mathcal{L}^{\star}_{\text{sub}} - \mathcal{L}^{\star}_{\text{full}} = H(Y \mid X, D^{(r)}) - H(Y \mid X, D^{(f)}, D^{(r)}) = I(Y; D^{(f)} \mid X, D^{(r)}),$$

which is nonnegative by definition of conditional mutual information. □

Remark 7 (Information gap)**.** *Proposition 6 shows that context-subset training induces a genuine information loss. In particular, the Bayes-optimal log-loss under the refined context $D^{(r)}$ exceeds that of full-context prediction by $I(Y; D^{(f)} \mid X, D^{(r)})$, which measures the additional target-relevant information contained in the discarded part of the context. Thus, subset conditioning generally defines a harder prediction problem in the statistical sense: even the optimal reduced-context predictor cannot match the performance of the optimal full-context predictor unless the removed context carries no additional information about Y. This explains why subset training can improve robustness while still weakening the predictive signal, and it provides a principled motivation for introducing full-context self-distillation to recover part of the lost information.*

Remark 8 (Task-level augmentation view)**.** *Because the refined context $D^{(r)}$ is randomly sampled from the full context $D^{(f)}$, a single original task induces many related reduced-context prediction problems. Thus, context-subset training can also be interpreted as a form of task-level augmentation.*

B.2 Why full-context self-distillation helps

We now study the distillation term. Our analysis proceeds in three steps. First, we identify the exact population target induced by forward-KL distillation from the full-context teacher to the reduced-context student. Second, we show that this target is asymptotically correct when the teacher is calibrated. Third, we show that the resulting penalty can improve finite-sample estimation through a teacher-guided shrinkage effect.

Proposition 9 (Population target of full-context forward-KL distillation)**.** *For each $(x, d^{(r)})$, define the conditional average of the full-context teacher by*

$$\bar{q}_\phi(\cdot \mid x, d^{(r)}) := \mathbb{E}_{D^{(f)}|X=x,\ D^{(r)}=d^{(r)}}\big[q_\phi(\cdot \mid x, D^{(f)})\big].$$

Then any conditional minimizer of

$$q \ \mapsto\ \mathbb{E}_{D^{(f)}|X=x,\ D^{(r)}=d^{(r)}}\Big[\mathrm{KL}\big(q_\phi(\cdot \mid x, D^{(f)}) \,\|\, q(\cdot \mid x, d^{(r)})\big)\Big]$$

is given by

$$q^\star_{\mathrm{dist}}(\cdot \mid x, d^{(r)}) = \bar{q}_\phi(\cdot \mid x, d^{(r)}).$$

Proof. Fix $(x, d^{(r)})$. Conditioned on $(X = x, D^{(r)} = d^{(r)})$, the objective becomes

$$\mathbb{E}_{D^{(f)}|x,d^{(r)}}\Big[\mathrm{KL}\big(q_\phi(\cdot \mid x, D^{(f)}) \,\|\, q(\cdot \mid x, d^{(r)})\big)\Big].$$

Expanding the KL divergence,

$$\mathrm{KL}(q_\phi \| q) = \int q_\phi(y) \log q_\phi(y)\, dy - \int q_\phi(y) \log q(y)\, dy.$$

Taking conditional expectation over $D^{(f)}$ given $(x, d^{(r)})$, the first term is independent of q. Hence minimizing the conditional expected KL is equivalent to minimizing

$$-\int \mathbb{E}_{D^{(f)}|x,d^{(r)}}\big[q_\phi(y \mid x, D^{(f)})\big] \log q(y \mid x, d^{(r)})\, dy.$$

By definition of $\bar{q}_\phi$, this equals

$$-\int \bar{q}_\phi(y \mid x, d^{(r)}) \log q(y \mid x, d^{(r)})\, dy = H\big(\bar{q}_\phi(\cdot \mid x, d^{(r)}), q(\cdot \mid x, d^{(r)})\big).$$

This cross-entropy is minimized exactly when

$$q(\cdot \mid x, d^{(r)}) = \bar{q}_\phi(\cdot \mid x, d^{(r)}),$$

which proves the result. □

Corollary 10 (No asymptotic target bias under a calibrated teacher)**.** *Assume that the teacher is calibrated in the sense that*

$$q_\phi(\cdot \mid x, D^{(f)}) = p_f(\cdot \mid x, D^{(f)}) \qquad \textit{almost surely}.$$

Then

$$\bar{q}_\phi(\cdot \mid x, D^{(r)}) = p_r(\cdot \mid x, D^{(r)}) \qquad \textit{almost surely}.$$

Consequently, under a calibrated teacher, the population target induced by full-context forward-KL distillation coincides with the reduced-context Bayes predictor.

Proof. Fix y. By Proposition 9 and the calibration assumption,

$$\bar{q}_\phi(y \mid x, D^{(r)}) = \mathbb{E}_{D^{(f)}|X=x,\ D^{(r)}}\big[q_\phi(y \mid x, D^{(f)})\big] = \mathbb{E}_{D^{(f)}|X=x,\ D^{(r)}}\big[p_f(y \mid x, D^{(f)})\big].$$

By the tower property of conditional expectation,

$$\mathbb{E}_{D^{(f)}|X=x,\ D^{(r)}}\big[p_f(y \mid x, D^{(f)})\big] = p_r(y \mid x, D^{(r)}).$$

Hence

$$\bar{q}_\phi(\cdot \mid x, D^{(r)}) = p_r(\cdot \mid x, D^{(r)}).$$

□

Proposition 11 (Effect of teacher mismatch)**.** *For a fixed teacher q_ϕ, the unrestricted population minimizer of the combined CSSDNP objective satisfies*

$$q_\lambda^\star(\cdot \mid x, d^{(r)}) = \frac{p_r(\cdot \mid x, d^{(r)}) + \lambda \bar{q}_\phi(\cdot \mid x, d^{(r)})}{1+\lambda}.$$

Consequently,

$$q_\lambda^\star - p_r = \frac{\lambda}{1+\lambda}(\bar{q}_\phi - p_r).$$

Thus, teacher mismatch introduces a population bias attenuated by the factor $\lambda/(1+\lambda)$.

Proof. For fixed $(x, d^{(r)})$, the population objective, up to terms independent of q, is

$$H(p_r, q) + \lambda H(\bar{q}_\phi, q).$$

By linearity of cross-entropy in its first argument, this is equivalent to

$$(1+\lambda) H\left(\frac{p_r + \lambda \bar{q}_\phi}{1+\lambda}, q\right),$$

which is minimized by the stated mixture. Subtracting p_r gives the mismatch identity. □

Remark 12 (Teacher calibration)**.** *Proposition 11 shows that perfect teacher calibration is sufficient, but not necessary, for an unbiased population target. The bias vanishes whenever $\bar{q}_\phi = p_r$, even if the teacher is not calibrated for every full context. Otherwise, the reduced-context NLL anchors the population target toward p_r and attenuates teacher mismatch by $\lambda/(1+\lambda)$. Nevertheless, substantial mismatch under extremely scarce meta-training tasks can still degrade learning.*

Rao–Blackwellized predictive supervision. Beyond preserving the population target, full-context distillation can provide a lower-variance learning signal than standard target-sample supervision. We formalize this interpretation under an ideal calibrated teacher.

Proposition 13 (Variance reduction through full-context supervision)**.** *Let*

$$G = -\nabla_\theta \log q_\theta(Y \mid X, D^{(r)})$$

denote the stochastic gradient of the reduced-context NLL, and assume that the teacher is calibrated and that differentiation and expectation can be interchanged. Define

$$G_{\mathrm{dist}} = \mathbb{E}_{Y \sim p_f(\cdot \mid X, D^{(f)})}\left[-\nabla_\theta \log q_\theta(Y \mid X, D^{(r)})\right].$$

Then,

$$\mathbb{E}[G_{\mathrm{dist}} \mid X, D^{(r)}] = \mathbb{E}[G \mid X, D^{(r)}],$$

and

$$\begin{aligned}\mathrm{Cov}(G \mid X, D^{(r)}) - \mathrm{Cov}(G_{\mathrm{dist}} \mid X, D^{(r)}) \\ = \mathbb{E}\left[\mathrm{Cov}(G \mid X, D^{(f)}, D^{(r)}) \mid X, D^{(r)}\right] \succeq 0.\end{aligned}$$

Proof. Under a calibrated teacher,

$$G_{\mathrm{dist}} = \mathbb{E}[G \mid X, D^{(f)}, D^{(r)}].$$

The equality of conditional means follows from the tower property. Applying the law of total covariance conditioned on $(X, D^{(r)})$ gives

$$\begin{aligned}\mathrm{Cov}(G \mid X, D^{(r)}) = \mathbb{E}[\mathrm{Cov}(G \mid X, D^{(f)}, D^{(r)}) \mid X, D^{(r)}] \\ + \mathrm{Cov}(G_{\mathrm{dist}} \mid X, D^{(r)}).\end{aligned}$$

The conditional covariance term is positive semidefinite, proving the result. □

Remark 14 (Finite-sample interpretation)**.** *Proposition 13 shows that, under an ideal calibrated teacher, full-context predictive supervision can be interpreted as Rao–Blackwellized soft supervision with no greater conditional gradient covariance than target-sample supervision. This provides an additional explanation for the finite-sample benefits of CSSDNP. In practice, teacher mismatch and Monte Carlo approximation may weaken this guarantee, while EMA smoothing can further stabilize the teacher.*

We next analyze a complementary finite-sample mechanism: teacher-guided shrinkage. While Proposition 13 characterizes the variance of predictive supervision, the following theorem studies how the distillation penalty can reduce parameter estimation error through increased local curvature. Define

$$R(\theta) := \mathbb{E}_{\tau, D^{(f)}, U, X, Y}\left[-\log q_\theta(Y \mid X, D^{(r)})\right]$$

and

$$D(\theta) := \mathbb{E}_{\tau, D^{(f)}, U, X}\Big[\mathrm{KL}\big(q_\phi(\cdot \mid X, D^{(f)}) \,\|\, q_\theta(\cdot \mid X, D^{(r)})\big)\Big].$$

Then

$$J_\lambda(\theta) = R(\theta) + \lambda D(\theta).$$

To isolate the finite-sample bias–variance tradeoff induced by full-context self-distillation, we analyze a local regime around a reference solution in which both the supervised risk and the distillation penalty admit a strongly convex and smooth approximation. This result should therefore be interpreted as a local mechanism theorem, rather than as a global characterization of the highly nonconvex training landscape of deep neural processes. Under this local approximation, the theorem formalizes the central effect of the distillation term: it can improve estimation by increasing effective curvature and thereby reducing variance, at the cost of a possible bias induced by teacher mismatch.

Theorem 15 (Local shrinkage effect of full-context self-distillation)**.** *Let*

$$\theta^\star \in \arg\min_\theta R(\theta), \qquad \theta^\star_\lambda \in \arg\min_\theta J_\lambda(\theta).$$

Given M i.i.d. meta-training tasks $\tau_1, \dots, \tau_M$, let

$$\widehat{J}_{\lambda,M}(\theta) = \frac{1}{M}\sum_{m=1}^{M} \ell_\lambda(\theta; \tau_m, U_m),$$

denote the empirical counterpart of $J_\lambda(\theta)$, where $\ell_\lambda(\theta; \tau_m, U_m)$ is the per-task CSSDNP loss including both the subset-training and distillation terms, and let

$$\widehat{\theta}_\lambda \in \arg\min_\theta \widehat{J}_{\lambda,M}(\theta).$$

Assume that on a neighborhood containing $\theta^\star$, $\theta^\star_\lambda$, and $\widehat{\theta}_\lambda$,

1. *R is μ-strongly convex;*
2. *D is α-strongly convex and has L_D-Lipschitz gradient;*
3. *there exists θ_ϕ such that $\nabla D(\theta_\phi) = 0$;*
4. *$\widehat{J}_{\lambda,M}$ is $(\mu + \lambda\alpha)$-strongly convex;*
5. *the empirical gradient noise at $\theta^\star_\lambda$ satisfies*

$$\mathbb{E}_{\tau_{1:M}, U_{1:M}}\left[\left\|\nabla \widehat{J}_{\lambda,M}(\theta^\star_\lambda)\right\|^2\right] \le \frac{\sigma^2}{M}.$$

Define $\rho := \|\theta^\star - \theta_\phi\|$, then

$$\mathbb{E}_{\tau_{1:M}, U_{1:M}}\left[\|\widehat{\theta}_\lambda - \theta^\star\|^2\right] \le \frac{2\sigma^2}{M(\mu + \lambda\alpha)^2} + \frac{2\lambda^2 L_D^2 \rho^2}{(\mu + \lambda\alpha)^2}.$$

Proof. We decompose

$$\widehat{\theta}_\lambda - \theta^\star = (\widehat{\theta}_\lambda - \theta^\star_\lambda) + (\theta^\star_\lambda - \theta^\star).$$

Using

$$\|a + b\|^2 \le 2\|a\|^2 + 2\|b\|^2,$$

it suffices to bound the estimation term and the bias term separately. Since $\widehat{J}_{\lambda,M}$ is $(\mu + \lambda\alpha)$-strongly convex and $\nabla \widehat{J}_{\lambda,M}(\widehat{\theta}_\lambda) = 0$, strong monotonicity of the gradient implies

$$(\mu + \lambda\alpha)\,\|\widehat{\theta}_\lambda - \theta^\star_\lambda\|^2 \le \left\langle \nabla \widehat{J}_{\lambda,M}(\theta^\star_\lambda) - \nabla \widehat{J}_{\lambda,M}(\widehat{\theta}_\lambda),\, \theta^\star_\lambda - \widehat{\theta}_\lambda \right\rangle.$$

Hence

$$(\mu + \lambda\alpha)\,\|\widehat{\theta}_\lambda - \theta^\star_\lambda\|^2 \le \left\|\nabla \widehat{J}_{\lambda,M}(\theta^\star_\lambda)\right\|\,\|\widehat{\theta}_\lambda - \theta^\star_\lambda\|,$$

which gives

$$\|\widehat{\theta}_\lambda - \theta^\star_\lambda\| \le \frac{\left\|\nabla \widehat{J}_{\lambda,M}(\theta^\star_\lambda)\right\|}{\mu + \lambda\alpha}.$$

Squaring and taking expectation with respect to $(\tau_{1:M}, U_{1:M})$,

$$\mathbb{E}_{\tau_{1:M},U_{1:M}}\left[\|\widehat{\theta}_\lambda - \theta^\star_\lambda\|^2\right] \le \frac{\sigma^2}{M(\mu + \lambda\alpha)^2}.$$

Since J_λ is $(\mu + \lambda\alpha)$-strongly convex and $\nabla J_\lambda(\theta^\star_\lambda) = 0$, we similarly have

$$(\mu + \lambda\alpha)\,\|\theta^\star_\lambda - \theta^\star\|^2 \le \langle \nabla J_\lambda(\theta^\star) - \nabla J_\lambda(\theta^\star_\lambda),\, \theta^\star - \theta^\star_\lambda\rangle = \langle \nabla J_\lambda(\theta^\star),\, \theta^\star - \theta^\star_\lambda\rangle\,.$$

Because $\theta^\star$ minimizes R, we have $\nabla R(\theta^\star) = 0$, hence

$$\nabla J_\lambda(\theta^\star) = \lambda \nabla D(\theta^\star).$$

Therefore,

$$(\mu + \lambda\alpha)\,\|\theta^\star_\lambda - \theta^\star\| \le \lambda \|\nabla D(\theta^\star)\|.$$

Using the L_D-Lipschitz continuity of ∇D and $\nabla D(\theta_\phi) = 0$,

$$\|\nabla D(\theta^\star)\| = \|\nabla D(\theta^\star) - \nabla D(\theta_\phi)\| \le L_D \|\theta^\star - \theta_\phi\| = L_D \rho.$$

Hence

$$\|\theta^\star_\lambda - \theta^\star\| \le \frac{\lambda L_D \rho}{\mu + \lambda\alpha},$$

and therefore

$$\|\theta^\star_\lambda - \theta^\star\|^2 \le \frac{\lambda^2 L_D^2 \rho^2}{(\mu + \lambda\alpha)^2}.$$

Using

$$\|\widehat{\theta}_\lambda - \theta^\star\|^2 \le 2\|\widehat{\theta}_\lambda - \theta^\star_\lambda\|^2 + 2\|\theta^\star_\lambda - \theta^\star\|^2,$$

and substituting the two bounds above yields

$$\mathbb{E}_{\tau_{1:M},U_{1:M}}\left[\|\widehat{\theta}_\lambda - \theta^\star\|^2\right] \le \frac{2\sigma^2}{M(\mu + \lambda\alpha)^2} + \frac{2\lambda^2 L_D^2 \rho^2}{(\mu + \lambda\alpha)^2}.$$

□

On Assumption 4. Assumption 4 is mainly a proof-oriented technical condition ensuring that the empirical objective retains enough local curvature for the estimation bound. We do not interpret it as a realistic global property of the full empirical objective in deep neural process training.

Remark 16 (Interpretation for CSSDNP)**.** *The results above provide a unified explanation for CSSDNP. Context-subset training is principled because it learns the correct reduced-context Bayes predictor while augmenting each task through multiple missing-context views. However, subset conditioning generally discards useful information and therefore makes prediction harder. Full-context self-distillation compensates for this by transferring information from a stronger full-context teacher. When the teacher is calibrated, this transfer does not alter the asymptotic reduced-context Bayes target, while teacher mismatch introduces a population bias attenuated by the reduced-context NLL. In the finite-meta-data regime, distillation can further improve learning through lower-variance predictive supervision and teacher-guided shrinkage.*

C Experimental Details

C.1 Implementation Details

Our experiments are based on the official code[2] of DANP [25]. All experiments were implemented in Python 3.11[3] using PyTorch 2.10.0[4] (CUDA 12.8) and NumPy 2.4.1[5]. Training was conducted on NVIDIA RTX PRO 6000 Blackwell workstation GPUs (96 GB VRAM each), using two GPUs to run experiments in parallel.

We evaluate four Neural Process base models: ANP [22], CANP (the conditional variant of ANP without a latent variable), DANP [25], and TNP [35]. For each base model, we train a corresponding CSSDA variant using our distillation framework. Following the CSSDA naming convention, the distilled variant of model X is denoted CSSDX; thus the four CSSDA models are **CSSDANP**, **CSSDCANP**, **CSSDDANP**, and **CSSDTNP**.

Architecture. Table 4 summarizes the architectural hyperparameters for each model. ANP and CANP are MLP-attention hybrids with cross-attention encoders; DANP and TNP-D are Transformer-based models. The CSSDA variants share identical architectures with their respective base models for the student (trainable) network. An EMA copy of the student is maintained as the teacher during training and discarded at inference, so inference-time parameters and computational cost are identical to the base model.

Table 4: Model architecture configurations. CSSDA variants share the same architecture as their base models (see text).

Model	**Type**	`dim_hid`	`dim_lat`	`enc_v`	`enc_qk`	`enc_pre`	`enc_post`	`dec`	**Params**
ANP	Latent	128	128	4	2	4	2	3	348K
CANP	Determ.	128	–	4	2	4	2	3	332K
		`d_model`	`emb_depth`	`d_ff`	`nhead`	`layers`			**Params**
DANP	Latent	64	4	128	4	6	–	–	335K
TNP-D	Determ.	64	4	128	4	6	–	–	222K

Table 5: CSSDA model configurations. Student architecture is identical to the base model; the EMA teacher is maintained during training only.

CSSDA Model	**Base Model**	`ema_momentum`	`student_ctx_ratio`	**Trainable Params**
CSSDANP	ANP	0.999	0.5	348K
CSSDCANP	CANP	0.999	0.5	332K
CSSDDANP	DANP	0.999	0.5	335K
CSSDTNP	TNP	0.999	0.5	222K

Training. All models are trained with the Adam optimizer. For training, we use a learning rate of 5×10^{-4} for 100,000 steps with a batch size of 16. For fine-tuning on out-of-distribution kernels (Matern, RQ) in the GP regression task, we sweep learning rates in $\{10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$. Distillation fine-tuning additionally sweeps the distillation weight $\lambda \in \{0.01, 0.05, 0.1, 0.5\}$ and student context ratio $r \in \{0.3, 0.5, 0.7\}$, with a minimum ratio of $0.6r$. And for all the models and baselines, we swept the weight-decay hyperparameters in $\{0, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$.

C.2 Inference

At inference time, we use only the student model; neither the teacher nor the distillation objective is required. Given a test task with context set $D^{(c)}$, the predictive distribution is computed by

[2] https://openreview.net/forum?id=uGJxl2odR0

[3] https://docs.python.org/3.11/

[4] https://pytorch.org/blog/pytorch-2-10-release-blog/

[5] https://numpy.org/devdocs/release/2.4.1-notes.html

Algorithm 1 CSSDA Training and Inference

Require: Pretrained base NP parameters θ_0; distillation weight λ; student context ratio $r \in (0, 1]$; minimum ratio $r_{\min}$; EMA momentum m.
Ensure: Trained student parameters θ_s.

Initialization:
Initialize student $\theta_s \leftarrow \theta_0$ and teacher $\theta_t \leftarrow \theta_0$
Freeze teacher: θ_t receives no gradients
Training loop (for each mini-batch):
Sample a task with full context $\mathcal{D}_c = \{(x_k, y_k)\}_{k=1}^{N_c}$ and target set $\mathcal{D}_T$
Compute $n_s = \max\big(\lceil r_{\min} N_c \rceil,\ \lceil r\, N_c \rceil\big)$
Sample student context $\mathcal{D}_s \subset \mathcal{D}_c$ by drawing n_s points uniformly at random
// Teacher forward (no gradient)
$p_t(y \mid x, \mathcal{D}_c) \ \leftarrow \ f_{\theta_t}(\mathcal{D}_c)$ ▷ full context, EMA parameters
// Student forward
$p_s(y \mid x, \mathcal{D}_s) \ \leftarrow \ f_{\theta_s}(\mathcal{D}_s)$ ▷ reduced context, trainable parameters
// Compute loss
$\mathcal{L}_{\mathrm{NP}} \ = \ -\frac{1}{|T|} \sum_{k \in T} \log p_s(y_k \mid x_k, \mathcal{D}_s)$
$\mathcal{L}_{\mathrm{distill}} \ = \ \frac{1}{|T|} \sum_{k \in T} D_{\mathrm{KL}}\big(p_t(\cdot \mid x_k, \mathcal{D}_c) \parallel p_s(\cdot \mid x_k, \mathcal{D}_s)\big)$
$\mathcal{L} \ = \ \mathcal{L}_{\mathrm{NP}} + \lambda\, \mathcal{L}_{\mathrm{distill}}$
Update $\theta_s \leftarrow \theta_s - \eta\, \nabla_{\theta_s} \mathcal{L}$
EMA update: $\theta_t \leftarrow m\, \theta_t + (1 - m)\, \theta_s$
Inference (given test context $\mathcal{D}_c$ and query locations X_*):
Discard teacher θ_t
Return $p_s(y \mid X_*, \mathcal{D}_c) \ = \ f_{\theta_s}(\mathcal{D}_c,\ X_*)$ ▷ full context, student only

conditioning the student on the full available context:

$$q_\theta(y^{(t)} \mid x^{(t)}, D^{(c)}) = \int q_\theta(y^{(t)} \mid x^{(t)}, z, D^{(c)})\, q_\theta(z \mid D^{(c)})\, dz.$$

Unlike training, we do not perform context subsetting at test time. Instead, prediction uses the full observed test context, so CSSDNP introduces no additional test-time optimization or teacher-dependent computation.

For latent neural process variants, this marginalization is approximated by Monte Carlo sampling over z, yielding a Gaussian-mixture predictive distribution. Thus, CSSDNP improves robustness during training while preserving the standard inference procedure of the underlying neural process model. Overall training and inference algorithms are summarized in Algorithm 1.

C.3 Evaluation Metrics

We evaluate all models using the normalized predictive log-likelihood on target points. For latent NP-based models (ANP, DANP, CSSDANP, CSSDDANP), where the representation $\mathbf{z}$ is stochastic, the log-likelihood is estimated via Monte Carlo sampling:

$$\frac{1}{|T|} \sum_{k \in T} \log p(y_k \mid x_k, \mathcal{D}_c) \approx \frac{1}{|T|} \sum_{k \in T} \log \left(\frac{1}{K} \sum_{i=1}^{K} p\Big(y_k \mid x_k, \mathbf{z}^{(i)}\Big) \right), \tag{1}$$

where $\mathbf{z}^{(i)} \overset{\text{i.i.d.}}{\sim} q(\mathbf{z} \mid \mathcal{D}_c)$ and T denotes the set of target indices.

For deterministic NP-based models (CANP, TNP, CSSDCANP, CSSDTNP), the predictive distribution is obtained directly without sampling ($K = 1$):

$$\frac{1}{|T|} \sum_{k \in T} \log p(y_k \mid x_k, \mathcal{D}_c). \tag{2}$$

In both cases, all CSSDA variants use only the student network with the *full* observed context at test time, so inference is identical to the corresponding base model.

C.4 Gaussian Process Regression

Task description. Each episode (meta-task) is a 1D regression problem generated from a GP prior. Input locations $x \in [-2, 2]$ are sampled uniformly, and observations y are drawn from a zero-mean GP, $f \sim \mathcal{GP}(0, k)$, with a given stationary kernel k. We experiment with three kernel families:

$$k_{\mathrm{RBF}}(x, x') = \sigma^2 \exp\left(-\frac{r^2}{2\ell^2}\right), \tag{3}$$

$$k_{\mathrm{Matérn}}(x, x') = \sigma^2 \left(1 + \frac{\sqrt{5}\, r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}\, r}{\ell}\right), \tag{4}$$

$$k_{\mathrm{RQ}}(x, x') = \sigma^2 \left(1 + \frac{r^2}{2\ell^2}\right)^{-1}, \tag{5}$$

where $r = |x - x'|$, and a small diagonal noise $\sigma_\epsilon^2 \mathbf{I}$ ($\sigma_\epsilon = 2 \times 10^{-2}$) is added to each kernel matrix for numerical stability. Kernel hyperparameters are randomised per episode: $\ell \sim \mathrm{Uniform}(0.1, 0.6)$ and $\sigma \sim \mathrm{Uniform}(0.1, 1.0)$. The number of context points $N_c \in [5, 45]$ and target points $N_t \in [5, 50 - N_c]$ are drawn uniformly at random per episode, giving at most 50 points per task. Each training batch contains 16 tasks.

Limited meta-training data. To simulate the limited meta-training data regime, we pre-generate a fixed dataset of N_b batches before training, where each batch comprises 16 independently sampled GP tasks. The model is trained for 100,000 steps by repeatedly cycling over this fixed set, so the effective number of unique tasks seen per epoch is $N_b \times 16$. We vary $N_b \in \{10, 20, 50, 100, 500, \infty\}$ to study the effect of meta-training diversity. In the unlimited setting ($N_b = \infty$), new tasks are sampled online at every step. All models are optimised with Adam at a learning rate of 5×10^{-4} with cosine annealing, and evaluation uses target log-likelihood averaged over a fixed held-out set of 3,000 batches.

Zero-shot evaluation under domain shift. To assess out-of-distribution (OOD) generalisation without any target-domain supervision, models trained exclusively on GP-RBF are evaluated directly on GP-Matérn-5/2 and GP-RQ at test time, with no fine-tuning. This setting tests whether the meta-learned prior over smooth functions transfers across kernel families under the same limited-data regimes. We vary $N_b \in \{50, 100, 200, 400, 1000\}$ to cover a wider range of training data volumes in the OOD setting.

Fine-tuning scenario. To study adaptation to a new kernel family under limited data, we adopt a pretrain-then-fine-tune protocol. A base model is first pre-trained on GP-RBF with $N_b = 1000$. The pretrained checkpoint initialises both the standard fine-tuning baseline and the distillation-augmented model, which are subsequently fine-tuned on a target kernel (Matérn-5/2 or RQ) with limited data ($N_b \in \{50, 100\}$), ensuring a fair comparison. For the baseline we search over learning rates $\eta \in \{1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}\}$. For the distillation model we fix $\eta = 1 \times 10^{-4}$ and sweep over distillation weight $\lambda \in \{0.01, 0.05, 0.1, 0.5\}$ and student context ratio $r \in \{0.3, 0.5, 0.7\}$ (with minimum ratio set to $r_{\min} = 0.6\, r$), reporting the configuration that maximises held-out target log-likelihood on the fine-tuning split.

C.5 Image Completion

Task description. Each episode (meta-task) is an image completion problem: given the (coordinate, intensity) pairs of a randomly selected subset of pixels as context, the model must predict intensity values at the remaining target pixel locations. A pixel at row h and column w of an $H \times W$ image has input coordinate

$$\mathbf{x} = \left(\frac{2h}{H-1} - 1,\ \frac{2w}{W-1} - 1\right) \in [-1, 1]^2, \tag{6}$$

and output $\mathbf{y} = \mathrm{pixel}/255 - 0.5 \in [-0.5, 0.5]^C$, where C is the number of colour channels. For each task, a random subset of up to M pixels is drawn and partitioned into a context set $\mathcal{C}$ ($N_c \in [3,\ M-3]$ chosen uniformly) and a target set $\mathcal{T}$ ($N_t \in [3,\ M - N_c]$ chosen uniformly).

CelebA. We use the CelebA large-scale face dataset [27], with images center-cropped and downsampled to 32×32 ($C = 3$, $d_x = 2$, $d_y = 3$), yielding 1,024 pixels per image. We follow the official train/test split: 162,770 training images and 19,867 evaluation images. During training we set $M = 200$; during evaluation we use $M = 50$ to provide a more challenging completion scenario where only a small fraction of pixels is observed.

Limited meta-training data. As in the other experiments, meta-training diversity is controlled by the total number of distinct training images, $N_{\text{img}} = N_b \times B$, where N_b is the number of batches and B is the batch size. Training proceeds for 100,000 steps by cycling over the fixed image subset, with each pass generating fresh random context/target splits. We vary the regimes as follows:

Dataset	B	N_b	N_{img}	Regime
CelebA	5	1	5	Extreme
	5	3	15	Extreme
	10	1	10	Extreme
	10	3	30	Very limited
	20	1	20	Extreme
	20	3	60	Very limited
	50	1	50	Limited
	50	3	150	Limited
	100	1	100	Moderate
	100	3	300	Moderate
CelebA/∞	full dataset		$\approx$162,770	Full

Table 6: Meta-training data regimes for the image completion experiments.

Distillation hyperparameter search. Hyperparameter selection is performed in the extreme limited-data regime ($N_b = 3$, $B = 5$, $N_{\text{img}} = 15$) for each dataset separately. We sweep over distillation weight $\lambda \in \{0.01, 0.05, 0.1, 0.5\}$ and student context ratio $r \in \{0.3, 0.5, 0.7\}$ (minimum ratio $r_{\min} = 0.6\, r$), selecting the configuration that maximises held-out target log-likelihood. The selected hyperparameters are fixed across all N_{img} settings. All models are optimised with Adam at a learning rate of 5×10^{-4} with cosine annealing over 100,000 steps.

C.6 Lotka-Volterra Population Dynamics

Task Description We consider the problem of inferring and predicting population dynamics in a stochastic predator-prey system. Given a sparse set of observed (time, population) measurements from an unknown trajectory realisation, the goal is to predict population counts at unobserved time points — a representative instance of ecological time-series regression under partial observation.

The system follows the stochastic Lotka-Volterra model simulated via the Gillespie algorithm [14], with four elementary reactions:

$$\theta_0\, XY \to X + 1 \quad \text{(predator birth)}, \tag{7}$$

$$\theta_1\, X \to X - 1 \quad \text{(predator death)}, \tag{8}$$

$$\theta_2\, Y \to Y + 1 \quad \text{(prey birth)}, \tag{9}$$

$$\theta_3\, XY \to Y - 1 \quad \text{(prey death)}, \tag{10}$$

where X and Y denote the predator and prey counts, respectively. Reaction rates are fixed at θ_0=0.01, θ_1=0.5, θ_2=1.0, θ_3=0.01, with initial populations $X_0 = 50 + \lfloor \varepsilon_X \rfloor$ and $Y_0 = 100 + \lfloor \varepsilon_Y \rfloor$, where $\varepsilon_X, \varepsilon_Y \sim \mathcal{N}(0, 1)$. Unlike the deterministic ODE formulation, the Gillespie algorithm produces exact sample paths of the underlying continuous-time Markov chain, yielding irregular event times and integer-valued population counts. Each trajectory therefore constitutes a distinct stochastic realisation of the same dynamical system rather than a different physical parameter configuration.

Neural Process Formulation We cast population dynamics inference as a meta-learning regression problem compatible with the Neural Process framework, following the experimental protocol of Lee et al. [26]. Each task is a single stochastic trajectory; the model's role is to infer the underlying population process from partial temporal observations and generalise to unobserved time points.

The input to the NP is the event time $x = t \in \mathbb{R}$ (z-score normalised using context statistics), and the output is the two-dimensional population vector $\mathbf{y} = (X, Y) \in \mathbb{R}^2$ (also z-score normalised). From a trajectory of M sampled events, N_c points are drawn at random as the context set $\mathcal{C} = \{(t_i, \mathbf{y}_i)\}_{i=1}^{N_c}$ and the remaining N_t points form the target set $\mathcal{T}$, where $N_c \sim \mathcal{U}[15, M - 15]$, $N_t \sim \mathcal{U}[15, M - N_c - 1]$, and $M = 100$ (the maximum number of context plus target points per task).

Training Data Generation Trajectory diversity arises solely from the intrinsic stochasticity of the Gillespie process. For each batch, a new stochastic realisation is drawn by running the algorithm for up to 20,000 reaction events; a random subset of $N_c + N_t$ events is then sampled from the resulting trajectory and partitioned into context and target sets. Unlike the system identification experiments, the kinetic parameters $(\theta_0, \theta_1, \theta_2, \theta_3)$ are *identical* across all training tasks — the model must learn a flexible prior over the stochastic trajectory space of a single dynamical system.

Limited Meta-Training Scenario The central experimental variable is N_b, the number of pre-generated training batches. Each batch contains $B = 16$ independently sampled trajectory realisations, so N_b directly controls how many distinct stochastic trajectories the model encounters during meta-training. We evaluate models under $N_b \in \{10, 20, 50, 100, 500, \infty\}$: small N_b corresponds to a highly repetitive training regime in which the model sees only $N_b \times 16$ unique trajectory snippets — for N_b=10 this amounts to merely 160 distinct realisations. All models are evaluated on a fixed held-out set of 500 batches drawn from the same Gillespie simulator, measuring in-distribution generalisation.

Sim-to-Real Evaluation The sim-to-real experiment directly tests the transfer gap between the idealised simulator and a real ecological dataset. Models are trained exclusively on synthetic Gillespie trajectories; the real-world evaluation set is the Hudson Bay Company Lynx-Hare dataset [30], comprising 21 annual observations of lynx (predator) and snowshoe hare (prey) population counts recorded between 1900 and 1920. This dataset has been widely used as a benchmark for stochastic Lotka-Volterra inference [26]. The transfer gap encompasses multiple sources of distribution shift: (i) the simulator generates continuous-time event sequences at irregular, sub-annual resolution, whereas the real data are annual census counts; (ii) real population dynamics are governed by additional ecological factors absent from the two-species model; and (iii) census counts carry measurement noise not present in simulation. Under limited N_b, the model has seen fewer trajectory shapes and is expected to maintain less calibrated uncertainty, making the sim-to-real transfer harder. Self-distillation may partially compensate by producing more calibrated predictive distributions from the EMA teacher, improving generalisation to the shifted real-data regime.

C.7 European Temperature Downscaling

Task Description. We consider the problem of *spatial temperature downscaling* over Europe, a task of direct relevance to regional climate modelling and high-resolution reanalysis. Given temperature anomaly observations at a sparse set of coarse-resolution grid points, the goal is to predict the anomaly at the intervening fine-resolution grid points that were not observed — a regression problem that mirrors the structure of the ConvCNP/ConvNP climate experiments of Gordon et al. [16] and Bruinsma et al. [5].

The temperature anomaly $y(\mathbf{x})$ at location $\mathbf{x}$ for a given month is the deviation from the long-run monthly climatology at that location. Subtracting the seasonal climatology removes deterministic annual cycles and exposes the *synoptic-scale climate state* of the month — driven by large-scale atmospheric modes such as the North Atlantic Oscillation (NAO), blocking events, or advection anomalies — which the NP encoder must infer from partial spatial observations.

Neural Process Formulation. Each **task** corresponds to one monthly snapshot over the European domain (longitude $-10°$–$40°$, latitude $35°$–$70°$). Inputs and outputs are defined as

$$\mathbf{x} = \left(\frac{\lambda}{90}, \frac{\phi_{\mathrm{s}}}{180}\right) \in [-1, 1]^2, \qquad y = \frac{T_{\mathrm{anom}}}{\sigma_{\mathrm{eur}}} \in \mathbb{R}, \tag{11}$$

where $\lambda \in [-90°, 90°]$ is latitude, $\phi_{\mathrm{s}} \in [-180°, 180°]$ is signed longitude (both normalised to $[-1, 1]$), and T_{anom} is the temperature anomaly standardised by the standard deviation computed over all Europe cells and training months ($d_x = 2$, $d_y = 1$).

The European domain contains 315 NCEP grid points at 2.5° resolution, which are split by spatial resolution into two non-overlapping subsets:

- **Coarse grid** (5° resolution): $N_c = 88$ points obtained by selecting every second grid point along both the latitude and longitude axes within the domain.
- **Fine grid** (2.5° resolution): $N_f = 227$ remaining NCEP points that lie between the coarse grid nodes.

This coarse/fine partition defines the context-target structure for both training and evaluation.

Following Gordon et al. [16], we evaluate in two complementary regimes:

- **Downscaling**: context $= \mathcal{C}_{\text{coarse}}$ (all 88 coarse observations), target $= \mathcal{T}_{\text{fine}}$ (all 227 fine points). This represents the pure downscaling problem, where no fine-resolution data are available at test time.
- **Fusion**: context $= \mathcal{C}_{\text{coarse}} \cup \mathcal{C}_{\text{fine}}^{(k)}$ (all 88 coarse plus k=50 randomly selected fine points), target = remaining 177 fine points. This represents data fusion, where sparse high-resolution observations are available alongside the coarse field.

During training, each batch randomly adopts either mode with equal probability ($p_{\text{fusion}} = 0.5$); in Fusion batches, the number of auxiliary fine context points is sampled uniformly in $[0, 100]$.

Dataset. We use the **NOAA NCEP/NCAR Reanalysis** monthly surface air temperature on a $2.5^\circ \times 2.5^\circ$ global grid (1948–present). The full record is trimmed to complete years, yielding $T = 936$ monthly snapshots. The monthly climatology (mean per calendar month across all years) is subtracted from each snapshot to obtain temperature anomalies, which are then standardised by the standard deviation over all European cells and months. The dataset is split chronologically:

- **Training**: first 80% of months (Jan. 1948 – approximately Jun. 2011, $N_{\text{train}} \approx 748$).
- **Evaluation**: last 20% of months ($N_{\text{eval}} \approx 188$, held out in chronological order and never seen during meta-training).

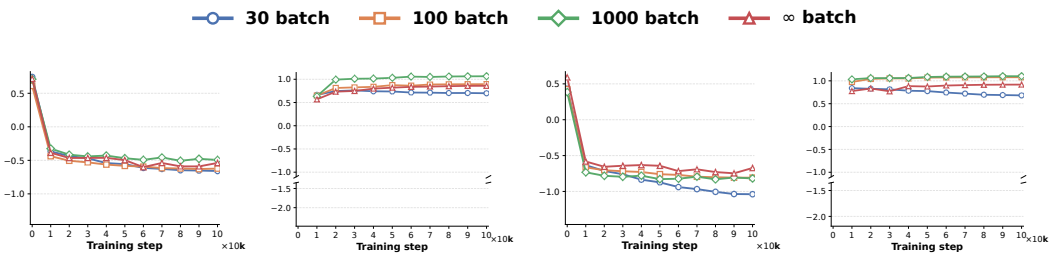
Limited Meta-Training Scenario. The meta-training diversity is controlled by $N_b \in \{10, 20, 50, 100, 500, \infty\}$, which limits the number of distinct monthly snapshots available during training. Each available month generates unlimited training batches via random mode selection and subsampling of fine context/target points, so limited N_b does not reduce the number of gradient steps, but strictly limits the *diversity of synoptic patterns* the model encounters.

N_b	Months available	Approximate span
10	10	$<$ 1 year
20	20	$\sim$1.5 years
50	50	$\sim$4 years
100	100	$\sim$8 years
500	500	$\sim$42 years
∞	$\approx$748	all training data

Table 7: Meta-training data regimes for the European temperature downscaling task.

European temperature anomalies are governed by recurrent large-scale circulation patterns (NAO phases, blocking, cold-air outbreaks) that are sparsely represented at low N_b: a model trained on $N_b = 10$ months may never encounter a strong positive NAO pattern, yet must predict fine-scale temperature structure under such conditions at evaluation time.

Why Self-Distillation Helps. The downscaling task exposes a specific failure mode of standard NPs under data scarcity: when the model has seen few distinct synoptic patterns, it may produce overconfident spatial extrapolations at fine-resolution target points that are distant from any coarse context observation. The EMA teacher provides a calibrated reference posterior that dampens overconfident predictions, discouraging the student from spuriously extrapolating spatial structure that was not well-supported by the training distribution. This regularisation is expected to be most



(a) CSSDANP: loss (b) CSSDANP: target LL (c) CSSDDANP: loss (d) CSSDDANP: target LL

Figure 6: Training dynamics under CSSDNP on the same RBF GP regression as Figure 1. Unlike standard training, CSSDNP maintains stable target log-likelihood even with low-meta-data regime.

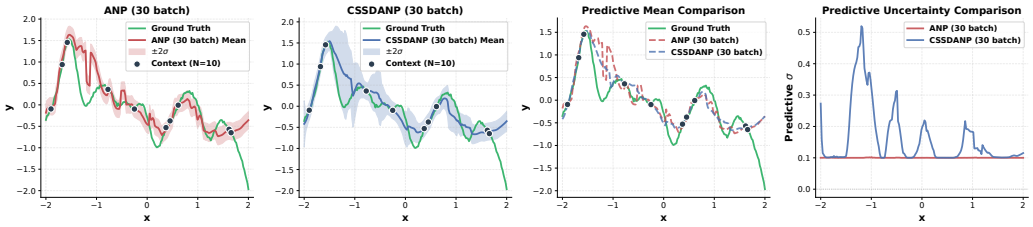


Figure 7: (a,b) Predictive mean and variance of ANP and CSSDANP in the low-meta-data regime. (c,d) Predictive mean and uncertainty comparison.

beneficial at low N_b and in the Downscaling regime, where the model must bridge the largest resolution gap without any fine-resolution observations in the context.

D Additional Experiments

D.1 Observation Revisiting: Additional Models

We extend the analysis of Section 3 to the remaining model families, which are ANP, DANP, and TNP, to verify that the training dynamics and predictive behaviour observed for CANP and TNP are not specific to those architectures. Figures 6, 7, 8, and 9 report the corresponding results.

As shown in Figure 6, CSSDANP and CSSDDANP exhibit the same stable training dynamics observed for CSSDCANP and CSSDTNP in Figure 2. Across all meta-training sizes, the validation target log-likelihood increases steadily without the collapse seen under standard training, confirming that the stabilising effect of CSSDNP generalises across architectures.

Figures 7, 8, and 9 further confirm that the improvement in predictive quality extends beyond CANP. In each case, the base model trained with few meta-training batches produces overconfident predictions with collapsed uncertainty, whereas its CSSD counterpart preserves well-calibrated variance that is small near observed context points and appropriately large in unseen regions. This pattern is consistent across all three architectures, reinforcing the conclusion from Section 3 that CSSDNP's benefit is not architecture-specific but reflects a general improvement in how uncertainty is propagated during training under limited meta-data.

D.2 Additional Baselines for 1D GP Regression Task

In this section, we conduct additional experiments comparing our method against generic regularization and consistency regularization baselines to further validate the effectiveness of CSSD. Specifically, we consider two representative approaches: Dropout [18], which randomly drops hidden units during training as a standard stochastic regularizer, and R-Drop [44], which enforces consistency between the output distributions of two independent forward passes under different dropout masks via a bidirectional KL divergence penalty. We apply these two baselines to all four NP architectures considered in Table 1 (CANP, ANP, DANP, and TNP) and evaluate them under the same finite

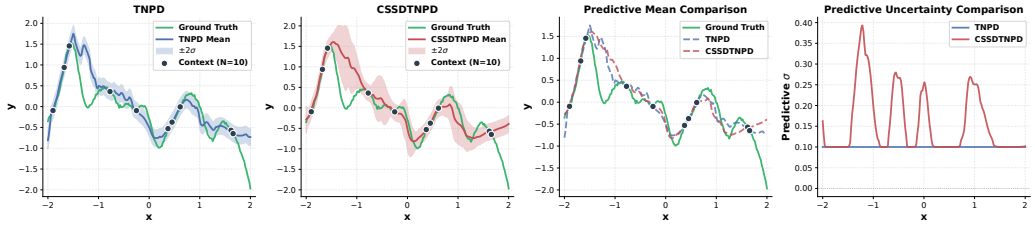


Figure 8: (a,b) Predictive mean and variance of TNP and CSSDTNP in the low-meta-data regime. (c,d) Predictive mean and uncertainty comparison.

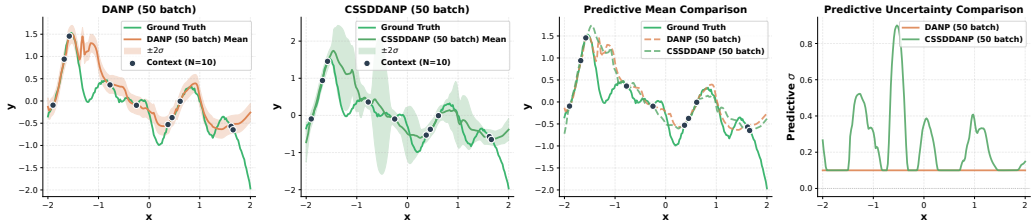


Figure 9: (a,b) Predictive mean and variance of DANP and CSSDDANP in the low-meta-data regime. (c,d) Predictive mean and uncertainty comparison.

meta-training conditions ($N_b \in \{1000, 50\}$) across three GP kernels. Here, because we already have weight decay regularization as the default for all models, we excluded the regularization-based methods.

As shown in Table 8, while these regularization baselines improve over their respective base models in many cases, they also frequently fail to do so, exhibiting inconsistent behavior across architectures and kernels. Even in the low-meta-train regime ($N_b = 50$), where the benefit of regularization is most expected, the gains from Dropout and R-Drop remain marginal compared to those achieved by CSSD. In contrast, CSSD consistently and substantially outperforms both baselines across all architectures, kernels, and meta-training sizes, demonstrating that its improvements stem not from generic regularization but from a principled exploitation of the context-prediction structure inherent to neural processes.

D.3 Additional 1D GP regression Results

While CSSDNP consistently outperforms its base models, Table 9 shows that its advantage diminishes under extreme meta-training data scarcity. At N_b=10, standard baselines often collapse, while CSSD variants still recover meaningful performance. However, the performance remains below the sufficient-data regime, indicating that CSSD cannot fully substitute for broader task diversity. At N_b=1, this limitation becomes absolute: with only a single meta-training batch, no model succeeds in learning a meaningful predictive distribution, regardless of the training objective. This suggests that even though CSSD enriches each training signal by exposing the student to diverse context subsets, the diversity of views over a fixed set of tasks cannot substitute for a broader coverage of the task space itself. When the meta-training distribution is reduced to a single task, there is simply insufficient variety for any form of generalisation to emerge.

D.4 Uncertainty Quantification

Metrics. We evaluate predictive uncertainty using three complementary metrics. **Root Mean Squared Error (RMSE)** measures the accuracy of the predictive mean, capturing how well the model's point estimate tracks the ground truth. **Coverage@95** is the empirical fraction of target observations that fall within the model's 95% predictive interval; a well-calibrated model should achieve coverage close to 0.95. **Continuous Ranked Probability Score (CRPS)** [15] is a proper scoring rule that jointly rewards sharpness and calibration: it penalises both overconfident predictive distributions that miss the observation and underconfident ones that spread probability mass too broadly. For a Gaussian predictive $\mathcal{N}(\mu, \sigma^2)$, CRPS admits a closed form and is lower-is-better.

Table 8: Target log-likelihoods on 1D GP regression across kernels and meta-training sizes.

Model	Enough-meta-train ($N_b = 1000$)			Low-meta-train ($N_b = 50$)		
	RBF	Matern	Rat.Quad.	RBF	Matern	Rat.Quad.
CANP	0.871±0.003	0.162±0.005	1.032±0.005	0.503±0.010	-0.888±0.021	0.723±0.018
CANP+Drop	0.855±0.002	0.211±0.006	1.031±0.005	0.561±0.006	-0.568±0.015	0.746±0.013
CANP+R-Drop	0.788±0.003	0.257±0.003	1.005±0.004	0.551±0.007	-0.452±0.017	0.799±0.009
CSSDCANP	**0.901** ±0.002	**0.241**±0.003	**1.060**±0.003	**0.747** ±0.008	**-0.017** ±0.013	**0.962**±0.011
TNP	0.963 ±0.002	**0.458**±0.004	1.088±0.004	0.401 ±0.009	-0.569±0.017	0.666±0.017
TNP+Drop	0.947 ±0.003	0.329±0.005	1.087±0.006	0.637 ±0.007	-0.621±0.023	0.907±0.011
TNP+R-Drop	0.922 ±0.003	0.276±0.005	1.079±0.005	0.682 ±0.010	-0.466±0.019	0.913±0.018
CSSDTNP	**0.972**±0.003	0.448±0.003	**1.099**±0.003	**0.849**±0.011	**0.124**±0.020	**1.029**±0.019
ANP	0.908 ±0.005	0.365±0.007	1.055±0.003	0.629 ±0.011	0.027 ±0.025	0.896±0.021
ANP+Drop	0.883 ±0.007	0.379±0.009	1.054±0.005	0.670 ±0.018	0.155 ±0.028	0.905±0.015
ANP+R-Drop	0.827 ±0.005	0.250±0.009	1.036±0.008	0.659 ±0.016	-0.154 ±0.023	0.865±0.017
CSSDANP	**0.919**±0.004	**0.373**±0.005	**1.063**±0.005	**0.827** ±0.010	**0.208** ±0.022	**0.997**±0.018
DANP	0.972 ±0.003	0.514±0.003	1.094±0.003	0.548±0.009	-0.518 ±0.018	0.814±0.018
DANP+Drop	0.943 ±0.004	0.394±0.004	1.087±0.004	0.754±0.012	-0.234 ±0.023	0.896±0.019
DANP+R-Drop	0.919 ±0.003	0.213±0.003	1.067±0.003	0.725±0.009	-0.177 ±0.020	0.923±0.017
CSSDDANP	**0.987**±0.004	**0.516** ±0.003	**1.109**±0.004	**0.868**±0.010	**0.200**±0.015	**1.035**±0.015

Table 9: Target log-likelihood on 1D GP regression (RBF) under small meta-training set sizes.

Method	tar_ll (N_b=10)	tar_ll (N_b=1)
CANP	-2.294 ± 0.032	-12.648 ± 0.103
CSSDCANP	$\mathbf{0.432}\pm0.015$	$\mathbf{-10.098}\pm0.075$
ANP	-1.442 ± 0.042	-11.846 ± 0.098
CSSDANP	$\mathbf{0.578}\pm0.025$	$\mathbf{-0.528}\pm0.044$
DANP	-1.485 ± 0.055	-5.911 ± 0.112
CSSDDANP	$\mathbf{0.652}\pm0.045$	$\mathbf{-5.106}\pm0.108$
TNP	-0.552 ± 0.035	-7.813 ± 0.100
CSSDTNP	$\mathbf{0.312}\pm0.022$	$\mathbf{-3.736}\pm0.061$

Experimental setting. We evaluate all models on 1D GP regression under three kernel families, e.g., RBF, Matern-$5/2$, and Rational Quadratic, across two context-set sizes, $N = 50$ and $N = 1000$, mirroring the setting of Table 1. Stochastic models (ANP, DANP, and their CSSD variants) use 50 latent samples at test time; the predictive mean and variance are obtained via the law of total expectation and total variance over those samples.

Results. The results reveal a consistent pattern across all three kernels and both context sizes. Baseline models systematically *undercover*: their 95% predictive intervals capture substantially fewer than 95% of target observations, indicating that the predictive uncertainty is underestimated. CSSD variants close this gap considerably, achieving coverage much closer to the nominal level. This improvement is not achieved by simply inflating the predictive variance—CRPS, which penalises unnecessary spread, also improves for CSSD models in the large-context regime ($N = 1000$), confirming that the gain comes from a genuinely better-calibrated distribution rather than from wider but less informative intervals.

The benefit of CSSD is most pronounced under the Matern kernel, which presents sharper local variations than RBF, and in the low-context regime ($N_b = 50$) where baselines exhibit the largest calibration gap. In this regime, context subsampling during training exposes the student to harder inference problems, and the EMA teacher provides a stable supervision signal that guides the student toward more conservative, yet accurate, uncertainty estimates. In the large-context regime ($N_b = 1000$), accuracy gains (RMSE and CRPS) become the dominant story: with sufficient context the models are no longer severely undercovering, and CSSD's advantage manifests primarily as sharper and more accurate predictions.

One notable exception is the DANP–CSSDDANP pair under the Matern kernel at $N = 50$, where CSSDDANP shows a slight RMSE regression while still improving coverage and CRPS. We attribute

Table 10: Uncertainty metrics on 1D GP regression with **RBF** kernel. RMSE and CRPS: ↓; Coverage@95: closer to 0.95 is better. Bold indicates the better value within each baseline–CSSD pair.

	$N = 50$			$N = 1000$		
Model	RMSE↓	Cov@95	CRPS↓	RMSE↓	Cov@95	CRPS↓
CANP	0.358 ±0.017	0.724±0.018	0.167±0.009	0.319±0.009	0.872±0.008	0.128±0.005
CSSDCANP	**0.351**±0.010	**0.864**±0.007	**0.144** ±0.005	**0.295** ±0.007	**0.936**±0.007	**0.117** ±0.004
ANP	0.377±0.011	0.711±0.011	0.178±0.007	0.304±0.006	0.915±0.006	**0.118**±0.005
CSSDANP	**0.332** ±0.013	**0.901** ±0.012	**0.134** ±0.010	**0.299** ±0.005	**0.944** ±0.007	**0.118**±0.006
DANP	**0.343** ±0.016	0.763±0.013	0.155±0.014	0.299±0.007	0.922±0.006	0.118±0.006
CSSDDANP	0.351±0.012	**0.902**±0.011	**0.141**±0.008	**0.280**±0.004	**0.947**±0.005	**0.106**±0.004
TNP	0.367±0.014	0.760±0.014	0.153±0.010	0.307±0.008	0.905±0.007	0.122±0.011
CSSDTNP	**0.347**±0.008	**0.875**±0.006	**0.144**±0.004	**0.281** ±0.005	**0.951**±0.003	**0.111**±0.004

Table 11: Uncertainty metrics on 1D GP regression with **Matern** kernel.

	$N = 50$			$N = 1000$		
Model	RMSE↓	Cov@95	CRPS↓	RMSE↓	Cov@95	CRPS↓
CANP	0.393±0.018	0.675±0.017	0.193±0.013	0.364±0.008	0.836±0.007	0.156±0.007
CSSDCANP	**0.377**±0.010	**0.877**±0.009	**0.173**±0.011	**0.322**±0.008	**0.922**±0.008	**0.139**±0.008
ANP	0.424±0.015	0.653±0.011	0.210±0.014	0.336±0.010	0.904±0.011	0.149±0.009
CSSDANP	**0.346**±0.010	**0.904**±0.011	**0.156**±0.011	**0.323**±0.007	**0.926**±0.008	**0.140**±0.006
DANP	**0.380**±0.015	0.706±0.015	0.179±0.009	0.319±0.008	0.926±0.006	0.136±0.006
CSSDDANP	0.398±0.011	**0.885**±0.012	**0.173**±0.011	**0.314**±0.006	**0.935**±0.006	**0.133**±0.005
TNP	0.415±0.017	0.699±0.016	**0.182**±0.015	0.334±0.008	0.911±0.008	**0.138**±0.006
CSSDTNP	**0.381**±0.019	**0.849**±0.017	**0.182**±0.017	**0.315**±0.010	**0.934**±0.010	**0.138**±0.008

this to the limited number of available training runs for this configuration; with more seeds the trend is expected to align with the other pairs.

D.5 Comparison with NP-Specific Low-Meta-Data Methods

We further evaluate CSSDNP on two NP-specific methods designed for learning with limited meta-training data: Sparse Gaussian Neural Processes [SGNP; 38] and Bayesian Neural Processes [BNNP; 39]. SGNP incorporates specialized sparse-GP structure, while BNNP performs amortized inference with a meta-learned Bayesian neural network prior. Unlike these approaches, CSSDNP improves the training objective without modifying the underlying model architecture.

As shown in Table 13, CSSDNP consistently improves both SGNP and BNNP across all three GP kernels. In particular, CSSDNP improves SGNP by 0.331–0.554 and BNNP by 0.134–0.218 in target log-likelihood. These results demonstrate that CSSDNP is compatible with existing NP-specific approaches for low-meta-data learning and provides complementary benefits beyond their architectural or prior-modeling mechanisms.

D.6 Teacher Calibration and Predictive Mismatch

Our theoretical analysis shows that full-context distillation preserves the Bayes-optimal reduced-context predictor under a calibrated teacher. However, an EMA teacher trained with limited meta-training tasks is not necessarily calibrated, particularly during early training or under extreme data scarcity.

To quantify teacher mismatch, we compare the teacher predictive distribution against the exact RBF-GP posterior across different meta-training sizes. We measure the KL divergence between the oracle posterior and the teacher prediction, together with the ratio of their predictive standard deviations.

Table 12: Uncertainty metrics on 1D GP regression with **Rational Quadratic** kernel.

	$N = 50$			$N = 1000$		
Model	RMSE↓	Cov@95	CRPS↓	RMSE↓	Cov@95	CRPS↓
CANP	**0.221**±0.020	0.816±0.018	0.113±0.018	0.200±0.011	0.892±0.013	0.090±0.013
CSSDCANP	0.228±0.015	**0.871**±0.015	**0.099**±0.014	**0.183**±0.008	**0.947**±0.010	**0.078**±0.008
ANP	0.229±0.013	0.805±0.012	0.108±0.012	0.188±0.006	0.933±0.006	**0.077**±0.005
CSSDANP	**0.207**±0.013	**0.907**±0.013	**0.089**±0.011	**0.180**±0.005	**0.954**±0.005	**0.077**±0.005
DANP	0.229±0.011	0.835±0.013	0.096±0.013	0.190±0.008	0.937±0.007	0.078±0.007
CSSDDANP	**0.211**±0.009	**0.918**±0.011	**0.094**±0.011	**0.176**±0.006	**0.958**±0.006	**0.074**±0.005
TNP	**0.215**±0.011	0.831±0.012	**0.096**±0.010	0.202±0.006	0.907±0.007	0.083±0.006
CSSDTNP	0.227±0.013	**0.889**±0.013	**0.096**±0.012	**0.179**±0.007	**0.955**±0.006	**0.076**±0.006

Table 13: Test log-likelihoods of SGNP and BNNP with and without CSSDNP on GP regression.

Kernel	SGNP	CSSDSGNP	BNNP	CSSDBNNP
RBF	0.035	**0.485**	0.231	**0.449**
Matern	-0.115	**0.216**	0.069	**0.193**
Rat.Quad.	0.385	**0.939**	0.744	**0.920**

As shown in Table 14, teacher mismatch decreases substantially as the number of meta-training tasks increases. At $N_b = 50$, the teacher's predictive standard deviation closely matches that of the exact GP posterior, while at $N_b = 100$, the KL divergence decreases further.

In contrast, at $N_b = 1$, the teacher severely underestimates predictive uncertainty, with its standard deviation reaching only 0.42 times the oracle value. This coincides with the extreme generalization collapse observed under severely limited task diversity.

These results support our theoretical characterization: CSSDNP approaches the Bayes-optimal reduced-context target as the teacher approaches the correct full-context predictive distribution, whereas substantial teacher mismatch can introduce population bias and degrade learning, as characterized in Proposition 11.

D.7 Generalization under Extreme Meta-Data Scarcity

Although CSSDNP substantially improves generalization in low-meta-data regimes, its benefits become limited when the available meta-training tasks are extremely scarce. To better understand this limitation, we analyze the training and held-out target log-likelihoods of CSSDNP under moderate and extreme meta-data scarcity in RBF-GP regression.

As shown in Table 15, the generalization gap remains small at $N_b = 50$, whereas it increases dramatically to 22.29 at $N_b = 1$. In this extreme setting, CSSDNP achieves high likelihood on the observed training tasks but fails to generalize to unseen task functions, indicating severe task-level memorization.

At $N_b = 1$, context subsets are randomly sampled from the same limited collection of training functions. Although they provide multiple context views, they cannot introduce new function-level diversity beyond the observed tasks. Consequently, the teacher also becomes severely under-dispersed on held-out tasks, as shown in Table 14.

These observations clarify the limitation of CSSDNP: while context-asymmetric self-distillation regularizes learning across different context views, it cannot recover information about unseen functions absent from the meta-training distribution.

D.8 Additional Ablations

Settings. We provide additional ablations to analyze the implementation choices in CSSDNP. Unless otherwise specified, experiments are conducted on 1D GP regression with the RBF kernel in the low-meta-data regime, and performance is measured by target log-likelihood. We use CSSDCANP for the EMA-related ablations and CSSDDANP for the Monte Carlo distillation ablation. Here, τ

Table 14: Predictive mismatch between the EMA teacher and the exact RBF-GP posterior across meta-training sizes.

N_b	Oracle–teacher KL	Teacher/oracle std ratio
1	39.80	0.42
10	8.74	0.76
50	3.58	0.99
100	2.68	1.18

Table 15: Student generalization under moderate and extreme meta-data scarcity in RBF-GP regression. The generalization gap is the difference between training-task and held-out target log-likelihoods.

N_b	Train-task LL	Held-out LL	Generalization gap
1	1.38	-20.91	22.29
50	0.85	0.74	0.11

denotes the EMA momentum of the teacher, λ denotes the distillation weight, r denotes the student context ratio, and K denotes the number of Monte Carlo samples used to approximate the forward-KL distillation objective for latent NP variants.

Effect of EMA teacher. Table 16 compares a non-EMA teacher ($\tau = 0$) with an EMA teacher ($\tau = 0.999$) under CSSDCANP, fixing $\lambda = 0.5$ and $r = 0.5$. The non-EMA teacher uses the current student parameters at every step, whereas the EMA teacher provides a temporally smoothed predictive target. Across all meta-training sizes, the EMA teacher consistently improves target log-likelihood: from 0.717 to 0.747 at $N = 50$, from 0.797 to 0.844 at $N = 100$, and from 0.860 to 0.876 at $N = 200$. The gain is largest at $N = 100$, suggesting that EMA smoothing is especially helpful when the student predictor changes rapidly during training. These results support the use of an EMA teacher as a stable full-context guide for self-distillation.

Sensitivity to EMA momentum. Table 17 sweeps the EMA momentum τ at fixed $N = 100$, $\lambda = 0.5$, and $r = 0.5$. The non-EMA setting ($\tau = 0$) gives the weakest performance, while introducing EMA substantially improves the result. In particular, increasing τ from 0 to 0.9 improves target log-likelihood from 0.797 to 0.834, and larger momentum values remain consistently strong. The best result is obtained at $\tau = 0.999$, but the performance is relatively stable for $\tau \in \{0.95, 0.99, 0.995, 0.999\}$. This indicates that CSSDNP benefits from a smoothed teacher target, while not being highly sensitive to the exact momentum once sufficient smoothing is applied. We therefore use $\tau = 0.999$ as the default in the main experiments.

Sensitivity to MC distillation samples. Table 18 varies the number of Monte Carlo samples K used to approximate the forward-KL distillation loss for CSSDDANP. Overall, performance is stable across K. At $N = 50$, increasing K gradually improves target log-likelihood from 0.868 at $K = 1$ to 0.877 at $K = 16$. At $N = 100$, the best result is achieved at $K = 8$, but the gap from $K = 1$ is only 0.004. Thus, while using more teacher samples can slightly improve the approximation of the distillation objective, the gains are marginal. Since the distillation cost increases with the number of teacher samples, we use $K = 1$ as the default, which provides a favorable trade-off between computational efficiency and predictive performance.

Effect of the Number of Monte Carlo Samples For latent NP variants, we approximate the forward-KL distillation objective using Monte Carlo samples from the full-context teacher predictive distribution. In our implementation, we use a single sample ($K = 1$) per training step.

Conditioned on the teacher predictive distribution, $K = 1$ provides an unbiased Monte Carlo estimator of the teacher-to-student cross-entropy, and, under standard regularity conditions, an unbiased stochastic gradient for the student parameters. Increasing K can reduce the sampling variance of this estimator, but does not change the population objective.

To examine the empirical effect of K, we vary the number of teacher samples in latent NP training. At $N_b = 50$, increasing K from 1 to 16 improves target log-likelihood only slightly, from 0.868

Table 16: EMA vs. no-EMA teacher (CSSDCANP, RBF, $\lambda = 0.5$, $r = 0.5$).

N	No-EMA ($\tau = 0$)	EMA ($\tau = 0.999$)
50	0.717±0.010	**0.747**±0.008
100	0.797±0.009	**0.844**±0.006
200	0.860±0.010	**0.876**±0.007

Table 17: EMA momentum sweep (CSSDCANP, RBF, $N = 100$, $\lambda = 0.5$, $r = 0.5$).

τ	Target LL
0.0	0.797±0.005
0.9	0.834±0.007
0.95	0.843±0.005
0.99	0.839±0.006
0.995	0.840±0.006
0.999	**0.844**±0.006

to 0.877. At $N_b = 100$, $K = 1$ achieves 0.941, compared with the best observed value of 0.945 at $K = 8$, while $K = 16$ again yields 0.941. These results suggest that increasing the number of Monte Carlo samples provides limited practical benefits in our setting. We therefore use $K = 1$ as a computationally efficient choice throughout our experiments.

D.9 Effect of EMA Smoothing

We further investigate the relative contributions of context asymmetry and EMA smoothing. To isolate the effect of EMA, we compare CSSDNP with a non-EMA teacher ($\tau = 0$), which shares the current student parameters but still observes the full context.

As shown in Table 19, CSSDNP without EMA already improves upon standard CANP by 0.193, demonstrating that context-asymmetric distillation provides substantial benefits even without temporal smoothing. EMA further improves target log-likelihood by 0.047, yielding the best performance.

These results suggest that context asymmetry accounts for most of the improvement, while EMA provides a complementary stabilization effect by supplying a temporally smoothed predictive target.

D.10 Multi-dimensional GP regression

In this section, we evaluate CSSDNP in a varying-dimensional GP regression setting using the DANP model as a base model. Here, the models are trained using two different input dimensions (2D, 4D) and assessed on held-out dimensions they have not explicitly encountered during training. This setup jointly tests two challenging capabilities: scaling to higher input dimensions, and generalizing to unseen dimensions under limited meta-training data.

As shown in Table 21, even under sufficient meta-training (N=1000), CSSDDANP improved log-likelihood performance across all dimensional inputs, suggesting that CSSDNP successfully operates on high-dimensional data. And, the critical distinction emerges at N=100: CSSDDANP consistently maintains strong generalization to unseen dimensions across the entire 1D–5D range, while standard DANP degrades sharply as dimensionality increases, with the gap expanding from $+0.10$ at 1D to $+0.29$ at 2D and persisting through 5D. These results highlight that CSSDNP provides a crucial inductive bias for dimension generalization precisely when meta-training data is scarce, a practically important regime where the base model alone fails to transfer reliably to novel input dimensions.

To further validate this finding, we also trained models jointly on 1D+3D inputs (Table 20). The same trend is consistently observed: at N=100, CSSDDANP outperforms DANP on both seen dimensions (1D: $+0.11$, 3D: $+0.13$) and unseen dimensions (2D: $+0.35$, 4D: $+0.04$, 5D: $+0.07$), confirming that the benefit of self-distillation extends across different training dimension configurations. Together, these experiments demonstrate that CSSDNP robustly enhances cross-dimensional generalization regardless of which dimensions are used during meta-training.

Table 18: MC distillation samples sweep (CSSDDANP, RBF, $\lambda = 0.5$, $r = 0.5$).

K	$N = 50$	$N = 100$
1	0.868±0.010	0.941±0.008
2	0.871±0.008	0.944±0.008
4	0.872±0.008	0.942±0.006
8	0.874±0.007	**0.945**±0.007
16	**0.877**±0.005	0.941±0.004

Table 19: Effect of EMA smoothing on RBF-GP regression at $N_b = 100$.

Method	Target log-likelihood
CANP	0.604
CSSD without EMA ($\tau = 0$)	0.797
CSSD with EMA	**0.844**

D.11 Image Completion

We evaluate CSSD on image completion using CelebA, varying the training batch size bs to control meta-training diversity. As shown in Table 22 and Figure 10, CSSD substantially improves all base architectures in the extremely low-data setting and remains competitive or beneficial in the larger-data setting. The gains are most pronounced at small bs, where base models such as ANP fail to generalise from the narrow meta-training distribution and produce context-insensitive completions regardless of the observed pixels. CSSD effectively overcomes this limitation: by training the student under subsampled context and aligning it with the EMA teacher's stable predictions, CSSD equips the model with the generalisation capacity that the limited training data alone cannot provide. Figure 10 illustrates this qualitatively, even with only a handful of training images (80 images), CSSDANP successfully reconstructs plausible facial structure and colour from sparse pixel observations, a level of quality that the base ANP fails to achieve under the same data budget. CSSD also continues to improve performance at large bs, confirming that the distillation objective serves as a consistent regulariser well beyond the data-scarce regime.

D.12 Climate Downscaling

We evaluate CSSD on a real-world spatial regression task under limited meta-training diversity using *European temperature downscaling* [16]. Each task is a monthly snapshot of surface air temperature anomalies over Europe; the model observes 88 coarse-grid ($5°$) context points and predicts 227 fine-grid ($2.5°$) targets. Meta-training diversity is controlled by varying the number of available monthly snapshots $N_b \in \{20, 50, 100, 500\}$.

As shown in Table 23, CSSD consistently outperforms all four base architectures across every N_b setting. The gains are most pronounced in the low-data regime ($N_b \leq 50$), where base models struggle to generalise from a limited set of synoptic patterns, while CSSD-augmented models remain well-calibrated, a direct effect of the EMA teacher providing a stable supervision signal that compensates for the narrow meta-training distribution. Figure 11 further illustrates this qualitative advantage: at N_b=20, CSSDDANP faithfully recovers the large-scale spatial anomaly structure, whereas DANP produces spatially inaccurate predictions. Notably, CSSD continues to improve log-likelihood even at N_b=500, confirming that the distillation objective benefits performance regardless of data availability.

D.13 Computational Cost

Although CSSDNP does not introduce additional inference-time computation, it requires an additional full-context teacher forward pass during training. To quantify this overhead, we compare standard CANP and CSSDCANP under the same 100,000-step training setting.

As shown in Table 24, CSSDCANP increases the training time per iteration from 14.2 ms to 16.6 ms, corresponding to approximately 17% overhead. This additional cost remains limited because the full-context teacher requires only a forward pass without backpropagation and is updated through an inexpensive EMA operation.

Table 20: Target log-likelihood on multi-dimensional GP regression (1D, 3D RBF kernel) with DANP model.

		Input Dimension				
N	Method	1D	2D	3D	4D	5D
N=100	DANP	0.829±0.011	-0.609±0.009	-0.703±0.013	-0.774±0.014	-0.933±0.017
	CSSDDANP	**0.939**±0.008	**-0.255**±0.007	**-0.570**±0.007	**-0.733**±0.011	**-0.859**±0.014
N=1000	DANP	0.904±0.009	**-0.301**±0.010	-0.619±0.011	-0.746±0.011	-0.920±0.012
	CSSDDANP	**0.916**±0.007	-0.398±0.008	**-0.577**±0.008	**-0.696**±0.009	**-0.761**±0.010

Table 21: Target log-likelihood on multi-dimensional GP regression (2D, 4D RBF kernel) with DANP model.

		Input Dimension				
N	Method	1D	2D	3D	4D	5D
N=100	DANP	0.854 ±0.015	0.030±0.017	-0.532±0.015	-0.852 ±0.019	-0.907 ±0.021
	CSSDDANP	**0.952**±0.016	**0.319**±0.016	**-0.279**±0.014	**-0.581**±0.017	**-0.684**±0.020
N=1000	DANP	0.862±0.013	**0.322**±0.013	-0.346±0.012	-0.592±0.013	**-0.674**±0.015
	CSSDDANP	**0.957**±0.014	**0.322**±0.015	**-0.303**±0.014	**-0.570**±0.015	**-0.674**±0.015

Peak memory does not increase in our implementation, partly because the student uses a reduced context and teacher predictions are computed without gradient tracking. Importantly, the teacher is discarded after training, leaving the model architecture, parameter count, and inference-time computation unchanged.

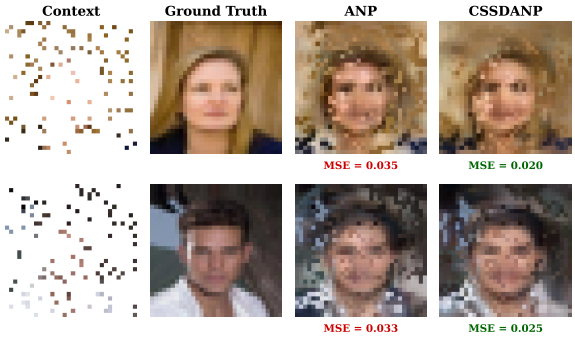
E Societal Impact

This paper empirically observes that Neural Processes fail to generalize to unseen meta-validation tasks when meta-training tasks are scarce. We then propose *Context-Subset Self-Distillation Neural Processes* (CSSDNP), a training framework that uses random context subsets and full-context self-distillation during training. Since Neural Processes are primarily probabilistic regression and meta-regression models, and our contribution is a training method for improving their generalization and uncertainty estimation, we do not anticipate direct harmful societal impacts.

Our experiments are conducted on standard benchmark settings, including synthetic Gaussian process data, climate and ecological regression tasks, and image completion benchmarks such as CelebA. We do not collect new personal data or deploy the model in user-facing applications. For datasets involving human images, we use them only as established academic benchmarks for image completion, and any practical use should respect the corresponding dataset licenses and privacy considerations.

CSSDNP also does not introduce additional inference cost over the base Neural Process model. All experiments in this paper were conducted using two RTX Pro 6000 GPUs, which are substantially smaller in scale than recent large language model training pipelines. Therefore, the expected environmental impact of this work is limited. Nevertheless, as with other uncertainty-aware models, care should be taken when applying CSSDNP to high-stakes domains such as healthcare, finance, or climate-risk assessment, where miscalibrated predictions may lead to harmful decisions. We view CSSDNP as a tool for improving robustness under limited meta-data, rather than a guarantee of safe deployment in all settings.

Table 22: Image completion results on CelebA using ANP and CSSDANP only 5 batches.



Model	bs=5	bs=100
CANP	-3.111±0.335	1.369±0.013
CSSDCANP	**0.850**±0.027	**1.418** ±0.009
ANP	-1.543±0.103	1.825±0.009
CSSDANP	**0.902**±0.014	**1.906**±0.011
DANP	1.129±0.008	**1.883**±0.007
CSSDDANP	**1.247**±0.008	1.833±0.006
TNP	0.833±0.007	1.544±0.006
CSSDTNP	**0.850**±0.006	**1.556**±0.008

Figure 10: **CelebA image completion** with limited training data. (*Left*) Qualitative comparison with 200 context pixels: CSSDANP produces sharper and more accurate completions than ANP. (*Right*) Quantitative results across context sizes. CSSD consistently improves ANP at all training data scales, with the largest gain at bs=5.

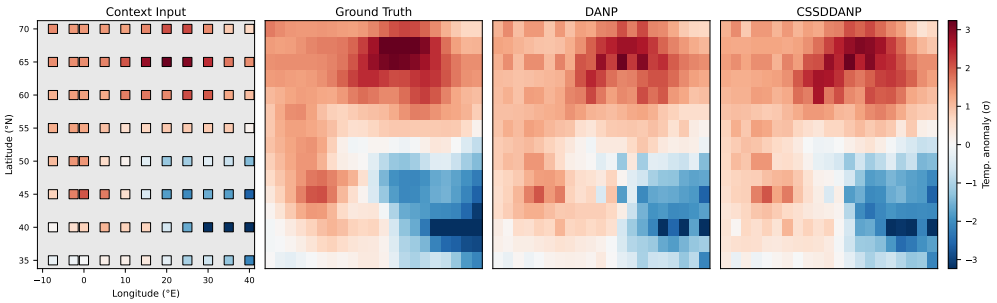


Figure 11: Qualitative comparison of temperature predictions over Europe. **(Left)** Context inputs. **(Center-left)** Ground truth fine-grid temperature anomaly. **(Center-right)** Prediction based on DANP with $N_b = 20$. **(Right)** Prediction based on CSSDDANP.

Table 23: Target log-likelihood on European temperature for varying meta-training sizes N_b.

Model	$N_b = 20$	$N_b = 50$	$N_b = 100$	$N_b = 500$
ANP	-15.743±2.138	-8.313±1.058	-3.222±0.872	-0.030±0.015
CSSDANP	**-1.439**±0.329	**-0.905**±0.394	**-0.782**±0.154	**0.293**±0.012
CANP	-15.063±1.924	-10.015±1.123	-4.150±0.904	-0.402±0.025
CSSDCANP	**-7.529**±1.091	**-1.459**±0.421	**-1.298**±0.389	**0.200**±0.023
DANP	-9.701±1.231	-0.818±0.212	-0.377±0.052	0.466±0.021
CSSDDANP	**-1.592**±0.383	**0.040**±0.139	**0.393**±0.072	**0.480**±0.018
TNP	-8.588±1.487	-0.958±0.315	-0.286±0.045	0.400±0.018
CSSDTNP	**-3.975**±0.772	**0.147**±0.083	**0.190**±0.051	**0.469**±0.017

Table 24: Computational cost of CANP and CSSDCANP.

Method	Time/iteration	Peak memory	Test-time cost
CANP	14.2 ms	37.7 MiB	1.0×
CSSDCANP	16.6 ms	33.1 MiB	1.0×
Relative cost	1.17×	0.88×	1.0×

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Contributions of this paper are well listed in the introduction. And we clearly and successfully discussed these contributions in Sec 3, 4, and 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of the experimental results in the Conclusion section and Sec D.3. Also, we discuss the limitation of our theoretical results in Sec B.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Our paper clearly contains the assumptions and proofs in Sec B. Also, we additionally discuss how the assumptions are strong in Sec B.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We fully describe our method in Sec 4. Also, we describe the experimental details in Sec C, and we also contain our code base in the supplementary material.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We contain our code base in the supplementary material. It contains the models and training code for GP regression task.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We describe all the experimental details in Sec C

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report the averaged performance and standard deviation of experimental results over five random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We discuss our compute resources in Sec C.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We anonymize our paper clearly and follow the Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the societal impact in Sec E. Our method does not have a critical or harmful impact on itself.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our method is not directly related to the misuse issue. Our method is a training framework for the regression model.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We contain the details of used packages and assets in Sec C with their official links.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We submit our code in the supplementary materials. We don't have any new dataset or pre-trained model.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We don't have any crowdsourcing or research with human subjects without the CelebA dataset. And the CelebA dataset is a well-known open source dataset with pictures of celebrities. Also, we discussed the details about the CelebA dataset in Sec C.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We don't have any special crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We used LLMs only for writing, editing, or formatting purposes.

    Guidelines:

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