DnNP: Denoising Input Uncertainty in Neural Processes

Fatemeh Tohidian[1] **Chengzhi Shi**[1] **Soomi Lee**[1] **Matthew Elia**[2]
Amy Mueller[2,3] **Stratis Ioannidis**[1] **Jennifer Dy**[1]

[1]Electrical & Computer Engineering [2]Civil & Environmental Engineering [3]Marine & Environmental Sciences
Northeastern University, Boston, MA, USA

Abstract

Neural Processes (NPs) are flexible meta-learning models capable of uncertainty quantification across diverse applications. However, the standard NP framework assumes access to noise-free inputs. This is unrealistic in many practical settings, where sensor noise or measurement errors are unavoidable. This discrepancy between model assumptions and real-world data can significantly degrade prediction quality and uncertainty estimates. We introduce Denoising Neural Processes (DnNPs), a principled extension of the NP framework that explicitly models input uncertainty. We train DnNPs via variational inference. DnNPs jointly learn to denoise corrupted inputs and perform meta-learning, making them robust to noisy training data. We demonstrate DnNPs' effectiveness on both synthetic and real-world datasets, showing consistent improvements in prediction and uncertainty quantification compared to standard NPs. On $PM_{2.5}$ spatial interpolation from real-world air quality sensors, DnNP reduces RMSE by up to 30% and NLL by up to 73% relative to standard NPs. We also provide theoretical grounding: we show that the Bayes-optimal predictor under noisy inputs belongs to the DnNP model class, and that plug-in models with fixed observation variance incur an irreducible KL divergence gap that DnNP provably avoids. Code is available at `https://github.com/iSUPER-ML/DnNP`.

1 Introduction

Meta-learning enables models to quickly adapt to new tasks by learning from many related tasks [8, 4]. Neural Processes (NPs) are flexible meta-learning models that provide uncertainty quantification and have shown strong performance across diverse applications [10]. However, NPs assume that inputs are observed without noise. In practice, inputs are often collected from sensors with measurement uncertainty [25]. This mismatch between model assumptions and real-world data can degrade prediction quality.

Classic NPs [5, 6, 7] do not model input noise. Inputs are treated as deterministic observations, and all uncertainty is captured through the latent variable and the output likelihood [6]. Although the problem of handling label noise for NPs has been studied [21], training NPs with inputs contaminated by noise remains unaddressed. This problem has been studied for Gaussian Processes [14], another method that outputs distributions over functions in a principled manner [24]. Moreover, handling noisy inputs in NPs is challenging for several reasons. Noisy inputs corrupt the learned relationships between features and outputs. This leads to both degraded predictions and miscalibrated uncertainty estimates that fail to account for measurement noise.

We propose Denoising Neural Processes (DnNP), which jointly learn to denoise inputs and make predictions with uncertainty quantification. We derive a principled training objective using variational inference that explicitly accounts for input uncertainty, which is applicable to other NP variants that

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

follow the standard NP framework. We evaluate DnNP across multiple scenarios including synthetic datasets and real-world air quality data, demonstrating consistent improvements. To summarize, our contributions are as follows:

- We propose DnNP, a Neural Process model that explicitly handles noisy input coordinates through a variational denoising mechanism, extending the NP family to a practically important but previously unaddressed setting.
- We provide theoretical guarantees showing that the Bayes-optimal predictor under noisy coordinates belongs to the DnNP model class, and that plug-in models with fixed observation variance incur an irreducible KL divergence gap that DnNP provably avoids.
- We demonstrate the effectiveness of DnNP on real-world $PM_{2.5}$ spatial interpolation on an Air Quality dataset, achieving up to 30% reduction in RMSE and 73% reduction in NLL over standard NPs across multiple feature configurations (Table 2).

2 Related Work

Neural Processes (NPs) have shown strong performance for function estimation across various applications, including weather forecasting [9, 1], spatiotemporal prediction [23, 19, 16], and medical image analysis [2]. NPs leverage the fast and scalable nature of neural networks, while providing a distribution over functions like Gaussian Processes (GPs). The first model of the Neural Processes family, Conditional Neural Processes (CNPs) [5], was inspired by the flexibility of GPs while remaining scalable to large and complex datasets. Latent Neural Processes (LNPs) [6] introduce a global latent variable that enables sampling functions which better match the underlying distribution by capturing global structure and uncertainty, addressing the independent prediction issue of CNPs, whereby predictions at different target points are made independently and thus lack coherence across the output space.

Attentive Neural Processes (ANPs) [11] leverage attention to allow each target prediction to selectively attend to the most relevant context points, in contrast to prior work that treats all context points equally through mean pooling. By explicitly modeling translation equivariance, a suitable structural assumption when the underlying process is stationary, Convolutional Conditional Neural Processes (ConvCNPs) [7] broaden the applicability of the Neural Processes family. Transformer Neural Processes (TNPs) [17] adopt a transformer architecture, recasting function approximation as a sequence modeling problem. Distance-Informed Neural Processes (DNPs) [22] enrich the latent space by adding a local latent variable. Functional Neural Processes (FNPs) [13] learn graph-based dependencies between data points rather than relying solely on global latent variables. Despite these advances, all the mentioned variants assume that the model inputs are noise-free, which causes overconfident and inaccurate predictions at test time in real-world settings where inputs are corrupted by noise from various sources.

Robust Neural Processes [21] introduce a training strategy to overcome label noise through two modifications to the standard NP loss function: an additional variance penalty term that discourages the model from inflating pointwise variance, instead encouraging uncertainty to be expressed at the global function level, and restricting the reconstruction loss to target points only to prevent memorization of noisy context values. In contrast, DnNP differs not only in the type of noise addressed, inputs x rather than labels y, but also in the nature of the contribution itself: while Robust NPs propose a training strategy applied to existing architectures, DnNP introduces changes to both the generative model and the architecture.

Gaussian Processes (GPs) have motivated most design choices in the NP family. Gaussian Process Training with Input Noise [14] addresses the problem of noisy inputs by applying a first-order Taylor expansion around the noisy observations as a proxy for the latent noise-free inputs. While an exact treatment would require maintaining a distribution over Taylor expansions, whose first and second moments are analytically tractable but computationally prohibitive, the authors instead differentiate only the GP mean function. This yields a linear model for the input noise, adding a term to the likelihood proportional to the squared posterior mean gradient. DnNP instead treats noise-free inputs as latent variables and learns a variational posterior over them within the NP meta-learning framework, enabling amortized inference across tasks.

3 Background

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$. A regression *episode* (task) $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ consists of input–output pairs, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, $i \in [n] \equiv \{1, \ldots, n\}$, are generated from an episode-specific latent function $g : \mathcal{X} \to \mathcal{Y}$. We partition indices $[n]$ into a *context* set C and *target* set T, where $C \cup T = [n]$. For $S \subseteq [n]$, denote by $\mathcal{D}_S = \{(x_i, y_i)\}_{i \in S}$, $x_S = \{x_i\}_{i \in S}$ and $y_S = \{y_i\}_{i \in S}$, the corresponding restricted datasets, features, and labels, respectively. Given a meta-dataset of episodes $\mathcal{M} = \{\mathcal{D}^{(k)}\}_{k=1}^K$, the meta-learning objective is to learn the conditional predictive distribution $p(y_T \mid x_T, \mathcal{D}_C)$. Intuitively, at test time, given a context dataset $\mathcal{D}_C$, and target inputs x_T, we should be able to predict the corresponding target labels y_T. For brevity, we omit superscript $^{(k)}$ from all per-episode quantities (e.g. $\mathcal{D}_T$, x_C, etc.) where it is unambiguous and write it explicitly when distinguishing across episodes matters (e.g., in the meta-objective).

Neural Processes (NPs) learn conditional $p(y_T \mid x_T, \mathcal{D}_C)$ across episodes in $\mathcal{M}$ while providing uncertainty quantification, assuming that inputs $x_{[n]}$ are observed accurately. In particular, NPs introduce a *global* latent variable z per episode to represent uncertainty over functions. The standard NP generative model is:

$$z \sim \mathcal{N}(0, I) \text{ (prior)}, \qquad y_i \mid x_i, z \sim \mathcal{N}\big(f_\theta(x_i, z),\, \sigma_y^2\big), \quad i \in [n], \text{ (label likelihood)} \tag{1}$$

where f_θ is a learnable decoder parameterized by θ and σ_y^2 is the observation noise variance. We write $f^\star(\cdot, z)$ for the ground-truth counterpart of $f_\theta(\cdot, z)$, i.e., the realizable optimum used in our theoretical analysis (Section 5). NPs utilize amortized inference, i.e., approximate the posterior on z via an encoder $q_\phi(z|\cdot)$, parameterized by ϕ; from this, NPs construct two approximate posteriors over z given the context, $q_\phi(z \mid \mathcal{D}_C)$, and given context plus targets, $q_\phi(z \mid \mathcal{D}_C \cup \mathcal{D}_T)$, used in training. The NP training objective is the evidence lower bound (ELBO) on the conditional log-likelihood:

$$\log p_\theta(y_T \,|\, x_T, \mathcal{D}_C) \geq \mathbb{E}_{q_\phi(z|\mathcal{D}_C \cup \mathcal{D}_T)}\Big[\sum_{i \in T} \log p_\theta(y_i \,|\, x_i, z)\Big] - \mathrm{KL}\Big(q_\phi(z \,|\, \mathcal{D}_C \cup \mathcal{D}_T) \,\|\, q_\phi(z \,|\, \mathcal{D}_C)\Big). \tag{2}$$

We denote the right-hand side of (2) as $\mathcal{L}_{\mathrm{ELBO}}^{(k)}(\Theta)$ when evaluated on episode $\mathcal{D}^{(k)}$ with split $(C^{(k)}, T^{(k)})$, where $\Theta = (\theta, \phi)$ collects all learnable parameters. The parameters are learned by maximizing the average per-episode ELBO over the meta-dataset:

$$\Theta^\star \;=\; \arg\max_{\Theta} \; \tfrac{1}{K} \textstyle\sum_{k=1}^K \mathcal{L}_{\mathrm{ELBO}}^{(k)}(\Theta). \tag{3}$$

This objective is optimized via stochastic gradient ascent. At each step, a mini-batch of episodes is sampled from $\mathcal{M}$, a context/target split is sampled for each, and the per-episode ELBO is estimated with a single Monte Carlo sample from q_ϕ using the reparameterization trick. At test time, given a new episode with context $\mathcal{D}_C$ and target inputs x_T, predictions can be obtained by: (i) drawing $z \sim q_\phi(\cdot \mid \mathcal{D}_C)$, then (ii) for each $i \in T$, predicting $\hat{y}_i = f_\theta(x_i, z)$.

4 Denoising Neural Processes (DnNP)

Neural Processes capture two sources of uncertainty: function-level uncertainty through the global latent variable z, and output noise through the variance σ_y^2. However, they assume that inputs are observed precisely, and cannot directly model input noise. In many practical scenarios, such as sensor networks with known measurement error, the observed input $x_i \in \mathcal{X}$ is a noisy version of an underlying clean input $x_i^* \in \mathcal{X}$. A standard NP applied directly to x_i implicitly assumes the inputs are noise-free; the resulting model is misspecified whenever measurement noise is non-negligible.

We address this by establishing a data-generative process that explicitly accounts for noisy inputs and propose a methodology for handling them within the Neural Process framework. We adopt the same episodic setup as in Section 3: a meta-dataset $\mathcal{M} = \{\mathcal{D}^{(k)}\}_{k=1}^K$ of episodes, each split into context and target sets C, T. The key difference is that we assume that episodes $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ comprise *noisy* input–output pairs: each x_i is a corrupted observation of an unobserved noise-free input x_i^*. We extend the NP generative process (1) by treating noise-free inputs as latent variables as follows:

$$x_i^* \sim p(x_i^*), \qquad x_i \mid x_i^* \sim \mathcal{N}(x_i^*, \Gamma), \qquad z \sim \mathcal{N}(0, I), \qquad y_i \mid x_i^*, z \sim \mathcal{N}\big(f_\theta(x_i^*, z),\, \sigma_y^2\big), \tag{4}$$

where $\Gamma \succ 0$ is the known measurement-noise covariance. The joint model is

$$p(\mathcal{D}, x^*_{[n]}, z) = p(z) \prod_{i=1}^n p(x_i^*)\, p(x_i \mid x_i^*)\, p_\theta(y_i \mid x_i^*, z). \tag{5}$$

The graphical model contains a *global* latent z (function uncertainty) and *local* latents x_i^* (measurement uncertainty); see Figure 1. The prior $p(x_i^*)$ can be chosen to reflect domain knowledge, e.g., uniform on a bounded region or a broad Gaussian.

The posterior $p(x^*_{[n]}, z \mid \mathcal{D})$ is intractable in general. We introduce a structured variational family that mirrors the conditional dependence in the generative model: a denoiser q_ψ, parameterized by ψ, that infers noise-free inputs from noisy ones, and an NP-style latent encoder q_ϕ that infers the function-level latent given noise-free inputs and observations:

$$q_{\psi,\phi}(x^*_{[n]}, z \,|\, \mathcal{D}) = q_\psi(x^*_{[n]} \,|\, x_{[n]})\; q_\phi(z \mid x^*_{[n]}, y_{[n]}), \quad q_\psi(x^*_{[n]} \,|\, x_{[n]}) = \textstyle\prod_{i=1}^n q_\psi(x_i^* \,|\, x_i). \tag{6}$$

The factorization $q_\psi(x_i^* \mid x_i)$ reflects an architectural choice: each noise-free input is inferred from its corresponding noisy measurement (and optionally a known noise level Γ), independently across observations. The latent encoder q_ϕ shares the form of the standard NP variational posterior (2) but now takes *denoised* inputs $x^*_{[n]}$ rather than the noisy $x_{[n]}$.

Combining the variational family (6) with the joint model (5) and applying Jensen's inequality yields the evidence lower bound (ELBO) on the conditional log-likelihood of the labels given the noisy inputs (full derivation in Appendix D):

$$\begin{aligned}\log p(y_{[n]} \mid x_{[n]}) \;\geq\; \mathbb{E}_{q_{\psi,\phi}}\Big[\underbrace{\textstyle\sum_{i=1}^n \log p_\theta(y_i \mid x_i^*, z)}_{\substack{\text{predictive}\\ \text{reconstruction}}} + \underbrace{\textstyle\sum_{i=1}^n \log p(x_i \mid x_i^*)}_{\substack{\text{measurement}\\ \text{consistency}}} \Big] \\ - \underbrace{\mathrm{KL}\Big(q_\psi(x^*_{[n]} \mid x_{[n]}) \,\|\, p(x^*_{[n]})\Big)}_{\substack{\text{denoiser}\\ \text{regularization}}} - \underbrace{\mathbb{E}_{q_\psi} \mathrm{KL}\Big(q_\phi(z \mid x^*_{[n]}, y_{[n]}) \,\|\, p(z)\Big)}_{\substack{\text{latent}\\ \text{regularization}}}.\end{aligned} \tag{7}$$

This bound has three interpretable components: (i) a predictive reconstruction term that scores y under *denoised* inputs, (ii) a measurement-consistency term $\log p(x_i \mid x_i^*)$ that ties the denoiser to the noise model, and (iii) two KL regularizers that prevent the denoiser and the latent encoder from drifting too far from their priors.

DnNP extends the NP parameter set with a denoiser network: $\Theta = (\theta, \phi, \psi)$, where θ parameterizes the decoder, ϕ the latent encoder, and ψ the denoiser variational posterior q_ψ. For meta-learning, the quantity of interest is the conditional likelihood of the targets given the context, $\log p(y_T \mid x_{[n]}, y_C)$. Applying the same bound to this conditional replaces the prior $p(z)$ in (7) with the conditional prior $p(z \mid x_C^*, y_C)$, and restricts the predictive reconstruction term to the target set. Since this conditional prior is intractable, we follow [6] and approximate it with the variational posterior $q_\phi(z \mid x_C^*, y_C)$, which yields the training objective:

$$\begin{aligned}\mathcal{L}^{(k)}_{\mathrm{ELBO}}(\Theta) = \mathbb{E}_{q_\psi(x^*_{[n]} | x_{[n]})}\left[\sum_{i=1}^n \log p(x_i \mid x_i^*)\right] - \mathrm{KL}\Big(q_\psi(x^*_{[n]} \,|\, x_{[n]}) \,\|\, p(x^*_{[n]})\Big) \\ + \mathbb{E}_{q_\psi(x^*_{[n]} | x_{[n]})}\left[\mathbb{E}_{q_\phi(z | x^*_{[n]}, y_{[n]})} \sum_{i \in T} \log p_\theta(y_i \mid x_i^*, z) - \mathrm{KL}\Big(q_\phi(z \,|\, x^*_{[n]}, y_{[n]}) \,\|\, q_\phi(z \,|\, x_C^*, y_C)\Big)\right].\end{aligned} \tag{8}$$

We learn Θ by maximizing the average per-episode ELBO over the meta-dataset, mirroring the NP meta-objective (3). We optimize this objective by stochastic gradient ascent with Adam: at each step, we sample a mini-batch of episodes from $\mathcal{M}$, sample a context/target split for each, and estimate the per-episode ELBO with a single Monte Carlo sample from each variational distribution (q_ψ and q_ϕ) using the reparameterization trick. Gradients flow through both sampling steps to update Θ. At test time, given a new episode with noisy context (x_C, y_C) and noisy target inputs x_T, we predict y_T by ancestral sampling: (i) draw $x^*_{[n]} \sim q_\psi(\cdot \mid x_{[n]})$, (ii) draw $z \sim q_\phi(\cdot \mid x_C^*, y_C)$, and (iii) for each $i \in T$, predict $\hat{y}_i = f_\theta(x_i^*, z)$. Note that the denoising step (i) departs from standard NP inference. If some inputs, or individual input dimensions, are known to be noise-free, the denoiser is skipped for them and the clean values are passed directly to the encoder and decoder.

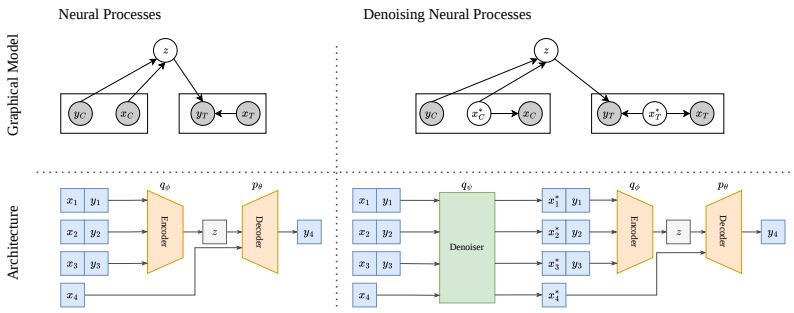

Figure 1: Comparison of Neural Processes (NP) and DnNP in terms of graphical model (top) and architecture (bottom). In the graphical model, shaded nodes represent observed variables. Standard NPs assume that inputs are observed noise-free, an assumption violated in many real-world settings where measurements are inherently corrupted by noise. DnNP addresses this by treating noise-free inputs as latent variables x^* that must be inferred from noisy observations x, while preserving the standard NP structure of context and target sets with a global latent z that captures function-level uncertainty. In the architecture, DnNP extends the standard NP encoder-decoder pipeline by prepending a denoising component (shown in green) that recovers noise-free input estimates from noisy inputs before passing them to the encoder.

5 Theoretical Guarantees

In this section, we analyze DnNP theoretically and show that it provably outperforms noise-unaware NP baselines in the presence of input noise. We proceed in four steps. First, under a Gaussian prior, the posterior for the unknown noise-free input x^* is Gaussian (Proposition 5.1). Second, we derive the Bayes-optimal predictive distribution $p^\star(y \mid x, z)$ under noisy inputs and characterize how input uncertainty inflates predictive variance (Equation 10). Third, we compare two predictive families: the DnNP family, which marginalizes over a learned posterior on x^*, and the plug-in family used by NP, which treats x as if it were noise-free. We prove that any plug-in model with fixed observation variance σ_y^2 incurs an irreducible KL gap to $p^\star$ whenever input uncertainty propagates through the ground-truth function (Theorem 5.2), while DnNP can close this gap entirely under realizability (Theorem 5.3). Fourth, we quantify how the plug-in gap scales with posterior input uncertainty: a moment expansion (Lemma 5.4) yields explicit KL lower bounds in the small-noise regime (Theorem 5.5), which simplifies to a clean $\tau^4\|\nabla f^\star\|^4/(4\sigma_y^4)$ scaling under isotropic posterior covariance (Corollary 5.6). All proofs are deferred to Appendix F. We use a Gaussian prior on x^* as a concrete example; the proof and an extension to a uniform prior are in Appendix C.

Proposition 5.1 (Gaussian posterior under Gaussian corruption)**.** *Under* (4) *with a Gaussian prior* $x^* \sim \mathcal{N}(\mu, \Sigma)$, $\Sigma \succ 0$*, the posterior over the noise-free input is Gaussian,* $p(x^* \mid x) = \mathcal{N}\big(x^* \mid \mu_{\text{post}}(x),\, \Sigma_{\text{post}}\big)$*, with* $\mu_{\text{post}}(x) = (\Sigma^{-1} + \Gamma^{-1})^{-1}(\Sigma^{-1}\mu + \Gamma^{-1}x), \Sigma_{\text{post}} = (\Sigma^{-1} + \Gamma^{-1})^{-1}$.

A bounded-domain version, obtained by replacing the Gaussian prior with a uniform prior on a hyper-rectangle, yields a truncated multivariate normal posterior; see Appendix C.

Bayes-Optimal Predictive Conditional. Given Proposition 5.1, the Bayes-optimal predictive conditional under noisy inputs is $p^\star(y \mid x, z) = \int \mathcal{N}\big(y \mid f^\star(x^*, z), \sigma_y^2\big)\, p(x^* \mid x)\, dx^*$. If $X \sim p(x^* \mid x)$ and $\varepsilon \sim \mathcal{N}(0, \sigma_y^2)$ is independent of X, then $Y = f^\star(X, z) + \varepsilon$ has conditional mean and variance

$$\mu_\star(x, z) := \mathbb{E}[Y \mid x, z] = \mathbb{E}[f^\star(X, z) \mid x]\,, \tag{9}$$

$$v_\star(x, z) := \mathrm{Var}(Y \mid x, z) = \sigma_y^2 + \mathrm{Var}(f^\star(X, z) \mid x)\,. \tag{10}$$

Equation (10) shows explicitly how input uncertainty inflates predictive variance. Since $p^\star(y \mid x, z)$ requires integrating over the intractable posterior $p(x^* \mid x, z)$, we instead utilize the variational posterior $q_\psi(x^* \mid x)$. The difference between DnNP and noise-unaware NP is most transparent at the level of conditional predictive distributions given the episode representation. Accordingly, throughout this section we condition on the episode latent z (random for NP). We assume the scalar-output generative model as in Section 4.

The DnNP predictive family (conditional on z) is

$$q_{\theta,\psi}^{\mathrm{DnNP}}(y \mid x, z) = \int \mathcal{N}\big(y \mid f_\theta(x^*, z), \sigma_y^2\big)\, q_\psi(x^* \mid x)\, dx^*. \tag{11}$$

In contrast, a noise-unaware NP baseline that plugs in x as if it were noise-free has conditional form

$$q_\theta^{\mathrm{plug}}(y \mid x, z) = \mathcal{N}\big(y \mid m_\theta(x, z), \sigma_y^2\big), \tag{12}$$

where m_θ is a measurable mean function. For NPs, (12) is the conditional decoder after fixing the global latent z.

Theorem 5.2 (Irreducible error in NPs)**.** *Fix (x, z) and let $P_{x,z} := p^\star(\cdot \mid x, z)$. For any measurable mean function $m(x, z)$,*

$$\mathrm{KL}\big(P_{x,z} \,\|\, \mathcal{N}\big(m(x,z), \sigma_y^2\big)\big) \geq \frac{\big(\mu_\star(x,z) - m(x,z)\big)^2}{2\sigma_y^2} + \frac{1}{2}\left(\frac{v_\star(x,z)}{\sigma_y^2} - 1 - \log\frac{v_\star(x,z)}{\sigma_y^2}\right).$$

In particular,

$$\inf_m \mathrm{KL}\big(P_{x,z} \,\|\, \mathcal{N}(m, \sigma_y^2)\big) \geq \frac{1}{2}\left(\frac{v_\star(x,z)}{\sigma_y^2} - 1 - \log\frac{v_\star(x,z)}{\sigma_y^2}\right), \tag{13}$$

and the right-hand side is strictly positive whenever $\mathrm{Var}\big(f^\star(X, z) \mid x\big) > 0$.

Equation (13) shows that NP-based models incur an irreducible KL gap whenever input uncertainty propagates through $f^\star$. We now show that DnNP, in contrast, can close this gap entirely because of the noise awareness.

Theorem 5.3 (Noise-aware realizability of DnNP)**.** *Assume there exist parameters $(\theta^\star, \psi^\star)$ such that*

$$\begin{aligned} f_{\theta^\star}(x^*, z) &= f^\star(x^*, z) \quad \textit{for all } (x^*, z), \\ q_{\psi^\star}(x^* \mid x) &= p(x^* \mid x) \quad \textit{for all } x. \end{aligned} \tag{14}$$

Then the noise-aware predictor coincides with the Bayes predictor:

$$q_{\theta^\star,\psi^\star}^{\mathrm{DnNP}}(y \mid x, z) = p^\star(y \mid x, z) \quad \textit{for all } (x, z). \tag{15}$$

Consequently,

$$\inf_{\theta,\psi} \mathbb{E}_{x,Z}\Big[\mathrm{KL}\big(p^\star(\cdot \mid x, Z) \,\|\, q_{\theta,\psi}^{\mathrm{DnNP}}(\cdot \mid x, Z)\big)\Big] = 0. \tag{16}$$

Theorem 5.3 shows that the Bayes predictor under noisy inputs belongs to the DnNP class. We now quantify how the plug-in KL gap from Theorem 5.2 scales with posterior input uncertainty in the small-noise regime. We first establish a moment expansion for the Bayes predictive moments, then combine it with Theorem 5.2 to obtain explicit KL lower bounds.

Lemma 5.4 (Small-noise moment expansion)**.** *Assume $f^\star(\cdot, z)$ is twice continuously differentiable with globally Lipschitz Hessian. As $\|\Sigma_{\mathrm{post}}\| \to 0$,*

$$\mu_\star(x, z) = f^\star(\mu_{\mathrm{post}}(x), z) + \frac{1}{2}\mathrm{tr}\big(\nabla_{x^*}^2 f^\star(\mu_{\mathrm{post}}(x), z)\, \Sigma_{\mathrm{post}}\big) + O(\|\Sigma_{\mathrm{post}}\|^{3/2}), \tag{17}$$

$$v_\star(x, z) - \sigma_y^2 = \big(\nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z)\big)^\top \Sigma_{\mathrm{post}}\, \nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z) + O(\|\Sigma_{\mathrm{post}}\|^2). \tag{18}$$

This lemma shows the posterior mean and variance of the ground truth function $f^\star$.

Theorem 5.5 (NP KL gap in the small-noise regime)**.** *Under the assumptions of Lemma 5.4, as $\|\Sigma_{\mathrm{post}}\| \to 0$,*

$$\begin{aligned} \mathrm{KL}\big(P_{x,z} \,\|\, \mathcal{N}\big(f^\star(\mu_{\mathrm{post}}(x), z),\, \sigma_y^2\big)\big) \geq\ & \frac{1}{8\sigma_y^2}\big(\mathrm{tr}(\nabla_{x^*}^2 f^\star(\mu_{\mathrm{post}}(x), z)\, \Sigma_{\mathrm{post}})\big)^2 \\ & + \frac{1}{4\sigma_y^4}\big(\nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z)^\top \Sigma_{\mathrm{post}}\, \nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z)\big)^2 + o(\|\Sigma_{\mathrm{post}}\|^2). \end{aligned} \tag{19}$$

Moreover, after optimizing over the predictive mean,

$$\inf_m \mathrm{KL}\big(P_{x,z} \,\|\, \mathcal{N}(m, \sigma_y^2)\big) \geq \frac{1}{4\sigma_y^4}\big(\nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z)^\top \Sigma_{\mathrm{post}}\, \nabla_{x^*} f^\star(\mu_{\mathrm{post}}(x), z)\big)^2 + o(\|\Sigma_{\mathrm{post}}\|^2). \tag{20}$$

Then, in small noise regime, we have the concrete KL lower bound for NP and the optimal Bayes predictor. Moreover, if we consider an isotropic posterior covariance, we have the following corollary.

Corollary 5.6 (Isotropic posterior covariance)**.** *Under the assumptions of Theorem 5.5, if* $\Sigma_{\text{post}} = \tau^2 I_d$*, then the irreducible term in* (20) *scales as* $\frac{\tau^4}{4\sigma_y^4}\left\|\nabla_{x^*} f^\star(\mu_{\text{post}}(x), z)\right\|_2^4 + o(\tau^4)$.

Thus, in the small noise regime, the irreducible error is related to the gradient of $f^\star$ at the ground truth x^*, the input noise variance, and the output noise variance.

6 Experiments

To evaluate the effectiveness of DnNP in handling noisy inputs, we conduct experiments comparing it against multiple Neural Processes variants on four datasets. We begin by describing the datasets and baseline methods, then introduce the experimental configuration, and conclude with a discussion of the results.

Data. Table 3 in the appendix summarizes the datasets used in our experiments. The synthetic dataset comprises 1D regression tasks with functions sampled from a GP with an RBF kernel. We also include MNIST [12] image completion, a widely-used benchmark in Neural Processes evaluations. Additionally, we apply our method to two real-world spatial interpolation datasets: an Air Quality dataset of $PM_{2.5}$ measurements that we collected from a network of urban sensors in Brookline, MA, which we make publicly available at `https://hdl.handle.net/2047/D20881761`, and a publicly available dataset of daily sea-level pressure observations from NOAA weather stations across California [18], obtained via the Meteostat library [15]. Both datasets contain real-world sensor measurements with inherent noise, demonstrating the importance of accounting for input uncertainty in practical applications.

Methods. We compared DnNP with standard Neural Processes (NP) [6], Conditional Neural Processes (CNP) [5], Attentive Neural Processes (ANP) [11], and Convolutional Conditional Neural Processes (ConvCNP) [7]. To assess the impact of input noise, we train an additional baseline on noise-free data for synthetically corrupted datasets (1D GP and MNIST), representing the best achievable performance without input noise. To simulate input uncertainty, we add zero-mean Gaussian noise to the model inputs with varying noise levels. Noise is applied independently across input dimensions and data points, modeling realistic measurement scenarios in which each sensor reading is affected by independent noise.

Metrics. We evaluate model performance on the target set using the posterior predictive, with latent variables marginalized by Monte Carlo: we draw K samples from the variational posteriors, obtaining predictive means and standard deviations (μ_k, σ_k), and compute $\text{NLL} = -\frac{1}{|T|}\sum_{i\in T}\log\frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(y_i \mid \mu_{k,i}, \sigma_{k,i}^2)$ and $\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{i\in T}(y_i - \bar{\mu}_i)^2}$, where $\bar{\mu}_i = \frac{1}{K}\sum_k \mu_{k,i}$. We compare our method against two neural process (NP) baselines. The first baseline is an NP trained on noise-free data (prior to noise injection). The second baseline is an NP trained directly on noisy inputs, representing the setting in which input uncertainty is ignored during modeling. We adopt the same backbone architecture for DnNP as for NP, with the addition of a denoising component. The model is trained using the loss defined in Equation 8. Hyper-parameters, including the weighting coefficient β that controls the strength of the denoiser KL regularizer relative to the predictive reconstruction term, are selected via grid search. The model achieving the lowest validation NLL is selected and evaluated on the test set, where it is compared against all baselines.

6.1 1D Synthetic Functions

We evaluated on synthetic 1D regression tasks where functions are sampled from a Gaussian Process with an RBF kernel (length-scale 0.5). During both training and evaluation, context points are corrupted by additive Gaussian noise $x_C = x_C^* + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_x^2)$, across five noise levels $\sigma_x \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, with a separate model trained for each noise level. All models share the same encoder-decoder architecture (hidden dim 128, latent dim 64) and are trained on 10,000 GP-sampled tasks with random context size drawn uniformly from [5, 30]. Clean NP is trained and evaluated on noise-free inputs x^* and serves as a lower bound on both NLL and RMSE, while Noisy NP is trained and evaluated on corrupted inputs x. Figure 2 shows that as noise increases, Noisy NP

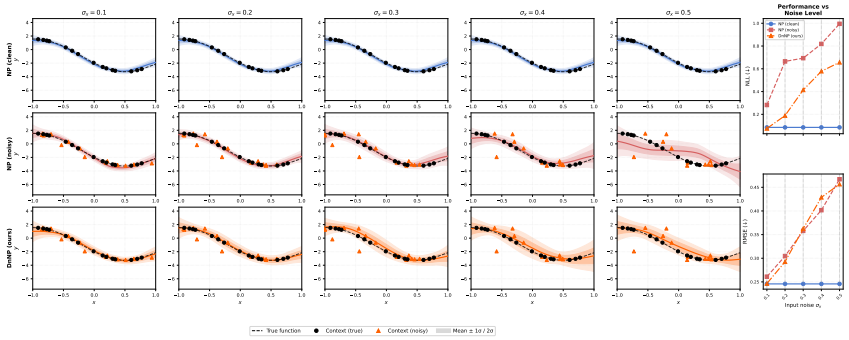

Figure 2: Comparison of Clean NP, Noisy NP, and DnNP on 1D GP regression across noise levels $\sigma_x \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Shaded regions show the predictive mean $\pm\sigma$ and $\pm 2\sigma$ uncertainty bands. Black circles indicate true context inputs and orange triangles show the corrupted observed input. As input noise increases, Noisy NP produces increasingly biased mean predictions, whereas DnNP maintains closer agreement with the true function. The NLL sweep (right) confirms this quantitatively, showing that DnNP achieves consistently lower NLL than Noisy NP across all noise levels.

progressively drifts from the true function. The NLL sweep confirms this quantitatively, showing that DnNP achieves consistently lower NLL across all noise levels.

DnNP is also robust to misspecification of the input noise level: on the 1D benchmark with true $\sigma_x = 0.3$, training with an assumed σ ranging from 0.5 to 2 times the true value changes RMSE only from 0.363 to 0.406 (vs. 0.373 at the true σ), with NLL nearly flat (≈ 0.47–0.54), within seed variation. This is consistent with our real-world experiments, where σ is set from manufacturer specifications rather than known exactly. When no estimate of σ is available, σ can be learned jointly with the model; the learned value tracks the true noise level monotonically, with a slight upward bias, and gives predictive performance close to, though not exceeding, the fixed-σ model, where the known σ is provided. We also experimented with more complex function classes such as periodic kernels, where all models, including Clean NP, underfit; this is consistent with prior observations that latent NPs with a global representation require more expressive architectures, such as attentive or convolutional NPs, to capture such structure [3].

6.2 MNIST Image Completion

We use the MNIST dataset of 28×28 grayscale handwritten digits with the standard 60K/10K train/test split. We further split the test set into 5K validation and 5K test subsets (fixed random seed for reproducibility). The task is image completion: given sparse pixel observations, predict the full 784-pixel image. During training, we randomly sample 3-200 context points and use all 784 pixels as targets. At test time, we use 100 context points. Inputs are 2D pixel coordinates (x_1, x_2) normalized to $[-1, 1]^2$. To simulate input uncertainty, we corrupt context inputs by adding independent Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbf{I}_2)$ where $\sigma_x \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.5\}$ in the normalized input space. These noise levels correspond to average pixel displacements of 0.02–7.4 pixels. We train separate NP and DnNP models for each noise level.

Table 1 and Figure 3 summarize the results on MNIST image completion under input noise. DnNP outperforms Noisy NP in NLL and RMSE at every noise level, and in accuracy for $\sigma_x \leq 0.1$, demonstrating that explicitly modeling input noise leads to more accurate and better-calibrated predictions. Notably, at low to moderate noise levels, DnNP recovers performance close to the noise-free baseline. As noise increases further, accuracy degrades for all methods as expected, since highly corrupted coordinates make the spatial structure of the image increasingly difficult to recover; at the largest noise levels ($\sigma_x \geq 0.2$), accuracy drops close to chance for both models, and the two are comparable. Note that RMSE and NLL remain nearly constant across noise levels because they are dominated by pixel-intensity reconstruction, which changes little as coordinate noise grows, whereas classification accuracy depends directly on whether the spatial structure of the digit is preserved; accuracy is therefore the more discriminative metric for this task. Qualitatively, Figure 3 shows that

Table 1: MNIST results. Left: clean-input baselines ($\sigma_x = 0$). Right: performance under increasing input noise σ_x for Noisy NP vs. DnNP; bold marks the better of the two at each noise level.

Method	NLL ↓	RMSE ↓	Acc ↑
NP	-0.0631	0.1413	0.706
CNP	-0.00571	0.232	0.319
ANP	-0.00702	0.190	0.733
ConvCNP	-0.00454	0.213	0.576

Metric	Method	σ_x 0.001	0.005	0.01	0.05	0.1	0.2	0.5
NLL ↓	Noisy NP	-0.0349	-0.0398	-0.0403	-0.0400	-0.0401	-0.0393	-0.0380
	DnNP	**-0.0616**	**-0.0630**	**-0.0624**	**-0.0637**	**-0.0646**	**-0.0637**	**-0.0628**
RMSE ↓	Noisy NP	0.243	0.236	0.232	0.233	0.233	0.235	0.239
	DnNP	**0.148**	**0.142**	**0.145**	**0.138**	**0.134**	**0.139**	**0.143**
Acc ↑	Noisy NP	0.205	0.207	0.221	0.210	0.215	**0.193**	**0.157**
	DnNP	**0.660**	**0.726**	**0.710**	**0.592**	**0.370**	0.184	0.123

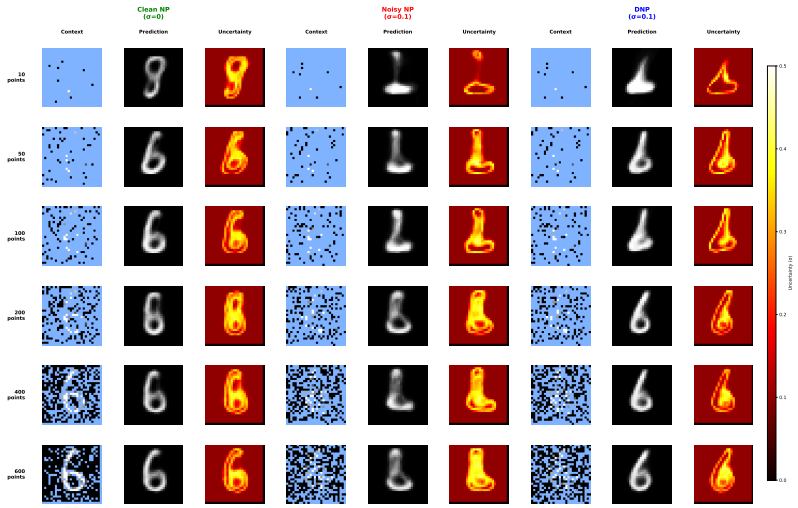

Figure 3: MNIST image completion with noisy pixel coordinates ($\sigma_x = 0.1$) at varying context sizes. Each row shows context, prediction, and per-pixel uncertainty for Clean NP ($\sigma_x = 0$, reference), Noisy NP, and DnNP. Noisy NP fails to recover the correct digit at all context sizes, while DnNP recovers digits comparable to Clean NP, with uncertainty concentrated along edges.

Noisy NP produces a persistent shape distortion regardless of context size, whereas DnNP recovers sharper and more coherent digit structures.

6.3 Spatial Interpolation

We collect hourly $PM_{2.5}$ data from 34 QuantAQ [20] air quality sensors located in Brookline, Massachusetts, USA, a town of approximately 17.6 km^2. The dataset is publicly available at `https://hdl.handle.net/2047/D20881761`. The analysis period spans May 24 to September 19, 2024, and all sensors used had more than 90% data completeness over the study period. Missing values were filled using linear imputation. We apply a log transformation to the $PM_{2.5}$ data and min-max normalization to other predictor variables such as latitude, longitude, temperature, and humidity. The dataset is divided into three splits along both the temporal and spatial axes. Temporally, the data are sequentially partitioned into three consecutive segments for training, validation, and testing: the first 80% for training, the next 10% for validation, and the remaining 10% for testing. Spatially, sensors are randomly split into three non-overlapping groups—context, target, and held-out—at approximately 55%, 15%, and 30%, respectively. For the training and validation splits, the context sensors are used to construct the feature vector and the model is trained/evaluated on the target sensors. To assess generalization to unobserved locations, the test split uses context-sensor values to construct the feature vector and evaluates predictions on the held-out sensors.

Input noise for the temperature and relative humidity dimensions is derived from the MODULAIR™ sensor specifications [20], which report measurement accuracies of $\pm 0.2°C$ for temperature and $\pm 1.8\%$ for relative humidity. Table 2 presents results on $PM_{2.5}$ spatial interpolation on the Air Quality dataset across three feature configurations. DnNP consistently outperforms standard NPs across all feature sets in both NLL and RMSE, and both improve over the mean baseline. These

Table 2: Spatial interpolation results on two real-world datasets, reported as mean $\pm$ standard deviation over 5 seeds. Mean RMSE is the constant-prediction baseline (predicting the dataset mean). Bold marks the better mean of NP vs. DnNP per metric within each row.

Dataset	Feature Set	Mean	NP		DnNP (ours)	
		RMSE ↓	NLL ↓	RMSE ↓	NLL ↓	RMSE ↓
PM2.5 (Air Quality)	Lat, Lon, RH	0.776	1.098 ± 0.273	0.457 ± 0.036	$\mathbf{0.296 \pm 0.078}$	$\mathbf{0.320 \pm 0.023}$
	Lat, Lon, Temp	0.776	1.038 ± 0.187	0.439 ± 0.030	$\mathbf{0.363 \pm 0.089}$	$\mathbf{0.337 \pm 0.055}$
	Lat, Lon, RH, Temp	0.776	0.925 ± 0.161	0.430 ± 0.032	$\mathbf{0.351 \pm 0.130}$	$\mathbf{0.348 \pm 0.062}$
Pressure (NOAA CA 2026)	Lat, Lon, Temp	0.108	$\mathbf{-1.240 \pm 0.032}$	0.052 ± 0.008	-1.239 ± 0.021	$\mathbf{0.036 \pm 0.006}$
	Lat, Lon, RH	0.108	-1.091 ± 0.064	0.076 ± 0.008	$\mathbf{-1.229 \pm 0.036}$	$\mathbf{0.039 \pm 0.010}$
	Lat, Lon, Temp, RH	0.108	-1.171 ± 0.031	0.064 ± 0.005	$\mathbf{-1.205 \pm 0.067}$	$\mathbf{0.048 \pm 0.015}$

results demonstrate that explicitly accounting for noisy input is particularly beneficial in spatial interpolation settings, where sensor data uncertainty is a natural and unavoidable source of noise. We further verified that the denoiser does not collapse to the identity map: its corrections to the noisy inputs are non-zero across all feature configurations and seeds, and small relative to the assumed noise level, as expected when sensor noise is small compared to the spread of each feature.

We additionally evaluate DnNP on sea-level pressure interpolation across NOAA weather stations in California. We retrieve daily observations of sea-level pressure, temperature, and relative humidity via the Meteostat library [15] for Q1 2026, retaining 29 stations with $\geq 95\%$ data completeness over the 90-day period. Missing values are linearly imputed, sea-level pressure is the prediction target, and predictor variables (latitude, longitude, temperature, humidity) are min-max normalized. We split the 90 days temporally into 80% training, 10% validation, and 10% test. Within each daily episode, sensors are randomly partitioned into context and target sets following the same protocol as the air quality experiment. Input noise standard deviations for temperature and relative humidity are derived from ASOS sensor specifications, which report typical accuracies of $\pm 0.5°$C for the platinum resistance thermometer and $\pm 2\%$ for the Vaisala capacitive humidity sensor. The corresponding rows of Table 2 report results across three input feature configurations. DnNP achieves lower RMSE than the NP in all three settings and lower NLL in two of three, with the largest gains under (Lat, Lon, RH) and (Lat, Lon, Temp, RH), where humidity enters as a noisy input. In the (Lat, Lon, Temp) setting, the two models are statistically tied in NLL (-1.240 ± 0.032 vs. -1.239 ± 0.021), while DnNP still reduces RMSE.

7 Conclusion

In this paper, we introduced DnNP, a new member of the Neural Process family that explicitly handles noisy input coordinates through a variational denoising mechanism. We evaluated DnNP on 1D GP regression, MNIST image completion, and two real-world $PM_{2.5}$ and sea-level pressure spatial interpolation datasets, achieving lower or comparable NLL and lower RMSE across different settings, with up to 30% RMSE reduction on the air quality dataset. We further provided theoretical guarantees showing that the Bayes-optimal predictor under noisy inputs belongs to the DnNP model class, while plug-in models that treat noisy inputs as clean incur an irreducible KL divergence gap. Natural future directions include combining DnNP with methods that handle noisy labels [21], particularly relevant for air quality datasets where both input covariates and target measurements are corrupted, and extending the framework to noise that is correlated across observations.

Acknowledgments and Disclosure of Funding

This work was supported by the National Institutes of Health (NIH) under grant 5R01HL167072, the National Science Foundation (NSF) under grant 2414652, and the iSUPER Impact Engine at Northeastern University.

References

[1] Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P Bruinsma, Tom R Andersson, Michael Herzog, Nicholas D Lane, Matthew Chantry, J Scott Hosking, and Richard E Turner. End-to-end data-driven weather prediction. *Nature*, 641(8065):1172–1179, May 2025.

[2] R. James Cotton, Fabian H. Sinz, and Andreas S. Tolias. Factorized neural processes for neural processes: k-shot prediction of neural responses, 2020.

[3] Yann Dubois, Jonathan Gordon, and Andrew YK Foong. Neural process family. `http://yanndubs.github.io/Neural-Process-Family/`, September 2020.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

[5] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018.

[6] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. *CoRR*, abs/1807.01622, 2018.

[7] Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020.

[8] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in neural networks: A survey. *CoRR*, abs/2004.05439, 2020.

[9] Daniel Jenson, Jhonathan Navott, Piotr Grynfelder, Mengyan Zhang, Makkunda Sharma, Elizaveta Semenova, and Seth Flaxman. Scalable spatiotemporal inference with biased scan attention transformer neural processes. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

[10] Saurav Jha, Dong Gong, Xuesong Wang, Richard E. Turner, and Lina Yao. The neural process family: Survey, applications and perspectives, 2023.

[11] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *CoRR*, abs/1901.05761, 2019.

[12] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[13] Christos Louizos, Xiahan Shi, Klamer Schutte, and Max Welling. The functional neural process, 2019.

[14] Andrew Mchutchon and Carl Rasmussen. Gaussian process training with input noise. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[15] Meteostat. Meteostat Python library, version 2.1.4. `https://dev.meteostat.net`.

[16] Sumantrak Mukherjee, Mouad Elhamdi, George Mohler, David Antony Selby, Yao Xie, Sebastian Josef Vollmer, and Gerrit Großmann. Neural spatiotemporal point processes: Trends and challenges. *Transactions on Machine Learning Research*, 2025. Survey Certification.

[17] Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling, 2023.

[18] NOAA National Centers for Environmental Information. Surface weather station observations. https://www.ncei.noaa.gov.

[19] Shenghao Qin, Jiacheng Zhu, Jimmy Qin, Wenshuo Wang, and Ding Zhao. Recurrent attentive neural process for sequential data. *CoRR*, abs/1910.09323, 2019.

[20] QuantAQ, Inc. MODULAIR™ Product Specification Sheet. https://assets.quant-aq.com/downloads/spec-sheets/modulair.latest.pdf, 2025. Accessed: 2026.

[21] Chen Shapira and Dan Rosenbaum. Robust neural processes for noisy data, 2024.

[22] Aishwarya Venkataramanan and Joachim Denzler. Distance-informed neural processes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[23] Timon Willi, Jonathan Masci, Jürgen Schmidhuber, and Christian Osendorfer. Recurrent neural processes. *CoRR*, abs/1906.05915, 2019.

[24] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.

[25] World Meteorological Organization. *Guide to Meteorological Instruments and Methods of Observation*. Secretariat of the World Meteorological Organization, 1983.

A Limitations

DnNP assumes measurement noise that is independent across observations, encoded by the pointwise denoiser $q_\psi(x_i^* \mid x_i)$; spatially correlated sensor bias, documented in both of our application domains, is not captured, and a set encoder approximating the joint posterior over clean inputs is a natural extension. The method also relies on a user-specified noise model and prior over clean inputs, which reduces the purely data-driven character of NPs and opens the door to misspecification, although DnNP is robust to moderate misspecification of σ. Finally, DnNP inherits the capacity of its backbone; since the denoiser is architecture-agnostic, combining it with more expressive models such as TNPs is a promising direction.

B Broader Impact

DnNP is designed for settings where input measurements are inherently noisy, a common condition in real-world sensor networks. Our spatial interpolation experiments on urban air quality data demonstrate that explicitly modeling input noise improves prediction accuracy in environmental monitoring applications. Improved interpolation in this setting can support exposure assessment, environmental justice analyses, and public health decision-making, particularly in cities that rely on sparse, low-cost sensor deployments where measurement uncertainty is unavoidable. More broadly, principled handling of input uncertainty is relevant to many scientific and engineering domains that depend on noisy measurements, including geosciences and climate modeling. We recommend that practitioners deploying DnNP validate the assumed input-noise model against sensor specifications or held-out calibration data.

C Posterior Over Noise-Free Inputs

Here we provide the proofs for the posterior propositions including the Gaussian prior and a truncated uniform prior.

C.1 Proof of Proposition 5.1

Proof. By Bayes' rule,

$$p(x^* \mid x) \propto p(x \mid x^*)\, p(x^*).$$

Using the generative model from Section 4,

$$p(x \mid x^*) \propto \exp\Big(-\tfrac{1}{2}(x-x^*)^\top \Gamma^{-1}(x-x^*)\Big), \qquad p(x^*) \propto \exp\Big(-\tfrac{1}{2}(x^*-\mu)^\top \Sigma^{-1}(x^*-\mu)\Big).$$

Therefore,

$$p(x^* \mid x) \propto \exp\Big(-\tfrac{1}{2}\Big[(x^*-\mu)^\top \Sigma^{-1}(x^*-\mu) + (x^*-x)^\top \Gamma^{-1}(x^*-x)\Big]\Big).$$

Expanding the quadratic form in x^* yields

$$(x^*-\mu)^\top \Sigma^{-1}(x^*-\mu)+(x^*-x)^\top \Gamma^{-1}(x^*-x) = {x^*}^\top(\Sigma^{-1}+\Gamma^{-1})x^* - 2(\Sigma^{-1}\mu+\Gamma^{-1}x)^\top x^* + \text{const}.$$

Set

$$\Lambda_{\text{post}} := \Sigma^{-1} + \Gamma^{-1}, \qquad \Sigma_{\text{post}} := \Lambda_{\text{post}}^{-1}, \qquad h := \Sigma^{-1}\mu + \Gamma^{-1}x.$$

Then

$${x^*}^\top \Lambda_{\text{post}} x^* - 2h^\top x^* = (x^* - \mu_{\text{post}}(x))^\top \Lambda_{\text{post}}(x^* - \mu_{\text{post}}(x)) - \mu_{\text{post}}(x)^\top \Lambda_{\text{post}} \mu_{\text{post}}(x),$$

where

$$\mu_{\text{post}}(x) := \Sigma_{\text{post}} h.$$

Hence

$$p(x^* \mid x) \propto \exp\Big(-\tfrac{1}{2}(x^* - \mu_{\text{post}}(x))^\top \Sigma_{\text{post}}^{-1}(x^* - \mu_{\text{post}}(x))\Big),$$

which is the Gaussian density stated in Proposition 5.1. □

C.2 Bounded-Domain Extension

Proposition C.1 (Uniform box prior + Gaussian corruption)**.** *Let $x^* \in \mathbb{R}^d$ have a uniform prior on the box*

$$\mathcal{B} = [a_1, b_1] \times \cdots \times [a_d, b_d].$$

Under the Gaussian corruption model $p(x \mid x^*) = \mathcal{N}(x \mid x^*, \Gamma)$, the posterior is the truncated multivariate normal

$$p(x^* \mid x) = \frac{\mathbf{1}_{\mathcal{B}}(x^*)}{Z(\mathcal{B}; x, \Gamma)}\, \phi_d(x^*; x, \Gamma),$$

where

$$\phi_d(x^*; x, \Gamma) = \frac{1}{(2\pi)^{d/2} \det(\Gamma)^{1/2}} \exp\Big(-\tfrac{1}{2}(x^* - x)^\top \Gamma^{-1}(x^* - x)\Big)$$

and

$$Z(\mathcal{B}; x, \Gamma) = \int_{\mathcal{B}} \phi_d(u; x, \Gamma)\, du.$$

Proof. The prior density is constant on $\mathcal{B}$ and zero outside, so Bayes' rule gives

$$p(x^* \mid x) \propto \mathbf{1}_{\mathcal{B}}(x^*)\, p(x \mid x^*).$$

Substituting the Gaussian corruption density yields

$$p(x^* \mid x) \propto \mathbf{1}_{\mathcal{B}}(x^*)\, \phi_d(x^*; x, \Gamma).$$

Normalizing over $\mathcal{B}$ gives the stated form. □

D ELBO Derivation

We derive the evidence lower bound (ELBO) for DnNP in four steps: first the full-data ELBO, then the context–target training objective.

Step 1: Marginal Likelihood. We aim to maximize the log-likelihood of the observed labels conditioned on the noisy inputs. Starting from the joint model (5), we marginalize over the latent variables z and $x^*_{[n]}$:

$$p(y_{[n]} \mid x_{[n]}) = \iint p(z)\, p(x^*_{[n]} \mid x_{[n]}) \prod_{i=1}^{n} p(y_i \mid x_i^*, z)\; dz\; dx^*, \tag{21}$$

where $p(x^*_{[n]} \mid x_{[n]}) = p(x_{[n]} \mid x^*_{[n]})\, p(x^*_{[n]}) \,/\, p(x_{[n]})$ follows from Bayes' rule.

Step 2: Introducing the Variational Family. Since the integral in (21) is intractable, we introduce the structured variational posterior from (6) by multiplying and dividing inside the integral:

$$\begin{aligned}\log p(y_{[n]} \mid x_{[n]}) = \log \iint & \frac{q_\psi(x^*_{[n]} \mid x_{[n]})\, q_\phi(z \mid x^*_{[n]}, y_{[n]})}{q_\psi(x^*_{[n]} \mid x_{[n]})\, q_\phi(z \mid x^*_{[n]}, y_{[n]})} \\ & \times\; p(z)\, p(x^*_{[n]} \mid x_{[n]}) \prod_{i=1}^{n} p(y_i \mid x_i^*, z)\; dz\; dx^*. \end{aligned} \tag{22}$$

Recognizing the double integral as an expectation under $q_\psi q_\phi$ and applying Jensen's inequality ($\log \mathbb{E}[\cdot] \geq \mathbb{E}[\log \cdot]$):

$$\log p(y_{[n]} \mid x_{[n]}) \geq \mathbb{E}_{q_\psi q_\phi}\left[\log \frac{p(x^*_{[n]} \mid x_{[n]})}{q_\psi(x^*_{[n]} \mid x_{[n]})} + \log \frac{p(z)}{q_\phi(z \mid x^*_{[n]}, y_{[n]})} + \sum_{i=1}^{n} \log p(y_i \mid x_i^*, z) \right]. \tag{23}$$

Step 3: Rewriting as KL Divergences. The first two terms in (23) are negative KL divergences by definition:

$$\mathbb{E}_{q_\psi}\left[\log \frac{p(x^*_{[n]} \mid x_{[n]})}{q_\psi(x^*_{[n]} \mid x_{[n]})}\right] = -\mathrm{KL}\Big(q_\psi(x^*_{[n]} \mid x_{[n]}) \parallel p(x^*_{[n]} \mid x_{[n]})\Big), \tag{24}$$

$$\mathbb{E}_{q_\phi}\left[\log \frac{p(z)}{q_\phi(z \mid x^*_{[n]}, y_{[n]})}\right] = -\mathrm{KL}\Big(q_\phi(z \mid x^*_{[n]}, y_{[n]}) \parallel p(z)\Big). \tag{25}$$

Note that $p(x^*_{[n]} \mid x_{[n]})$ factors as $\prod_i p(x^*_i \mid x_i)$ under the assumed independence structure, so the KL decomposes point-wise. Furthermore, expanding $p(x^*_{[n]} \mid x_{[n]}) \propto p(x_{[n]} \mid x^*_{[n]})\, p(x^*_{[n]})$ and absorbing the $\log p(x_{[n]} \mid x^*_{[n]})$ term into the reconstruction objective yields the form stated in (7).

Step 4: Context–Target Split. For meta-learning we condition on the context set and predict targets. Splitting indices into context $C = \{1, \ldots, m\}$ and target $T = \{m+1, \ldots, n\}$, and replacing the prior $p(z)$ with the context posterior $q_\phi(z \mid x^*_C, y_C)$ as the baseline (following the NP training objective (2)):

$$\begin{aligned}\log p(y_T \mid x_{[n]}, y_C) \geq \mathbb{E}_{q_\psi(x^*_{[n]} \mid x_{[n]})}\Bigg[&\mathbb{E}_{q_\phi(z \mid x^*_{[n]}, y_{[n]})} \sum_{i \in T} \log p(y_i \mid x^*_i, z) \\ &- \mathrm{KL}\Big(q_\phi(z \mid x^*_{[n]}, y_{[n]}) \parallel q_\phi(z \mid x^*_C, y_C)\Big)\Bigg] \\ &- \mathrm{KL}\Big(q_\psi(x^*_{[n]} \mid x_{[n]}) \parallel p(x^*_{[n]})\Big) + \mathbb{E}_{q_\psi} \sum_{i=1}^{n} \log p(x_i \mid x^*_i),\end{aligned} \tag{26}$$

which matches (8) in the main text.

E Auxiliary Lemmas for Section 5

Here, we provide the lemmas that are used to prove the theorems and corollaries in Section 5.

Lemma E.1 (Gaussian cross-entropy)**.** *Let P be an absolutely continuous distribution on $\mathbb{R}$ with finite mean μ_P and variance v_P. For $Q = \mathcal{N}(m, \sigma^2)$ with density q, one has*

$$-\mathbb{E}_{Y \sim P}[\log q(Y)] = \frac{1}{2}\log(2\pi\sigma^2) + \frac{v_P + (\mu_P - m)^2}{2\sigma^2}. \tag{27}$$

Proof. Since

$$-\log q(Y) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{(Y-m)^2}{2\sigma^2},$$

taking expectation under P gives

$$-\mathbb{E}_P[\log q(Y)] = \frac{1}{2}\log(2\pi\sigma^2) + \frac{\mathbb{E}_P[(Y-m)^2]}{2\sigma^2}.$$

Using the bias–variance decomposition

$$\mathbb{E}_P[(Y-m)^2] = \mathrm{Var}_P(Y) + (\mathbb{E}_P[Y] - m)^2 = v_P + (\mu_P - m)^2$$

proves (27). □

Lemma E.2 (Entropy upper bound at fixed variance)**.** *Let P be an absolutely continuous distribution on $\mathbb{R}$ with finite mean μ_P and finite variance v_P. Then*

$$H(P) \leq \frac{1}{2}\log(2\pi e\, v_P), \tag{28}$$

with equality if and only if P is Gaussian with variance v_P.

Proof. Let $G = \mathcal{N}(\mu_P, v_P)$ with density g. Since $\mathrm{KL}(P\|G) \geq 0$,

$$-H(P) - \mathbb{E}_P[\log g(Y)] \geq 0.$$

Applying Lemma E.1 with $m = \mu_P$ and $\sigma^2 = v_P$ gives

$$-\mathbb{E}_P[\log g(Y)] = \frac{1}{2}\log(2\pi v_P) + \frac{v_P}{2v_P} = \frac{1}{2}\log(2\pi e\, v_P).$$

Hence

$$H(P) \leq \frac{1}{2}\log(2\pi e\, v_P).$$

Equality holds if and only if $\mathrm{KL}(P\|G) = 0$, i.e. $P = G$ almost everywhere. □

Lemma E.3 (Third-order Taylor expansion with Lipschitz Hessian)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice continuously differentiable and assume its Hessian is globally L-Lipschitz:*

$$\|\nabla^2 f(u) - \nabla^2 f(v)\| \leq L\|u - v\| \qquad \textit{for all } u, v \in \mathbb{R}^d.$$

Fix $u_0 \in \mathbb{R}^d$, and set $g = \nabla f(u_0)$ and $H = \nabla^2 f(u_0)$. Then for every $h \in \mathbb{R}^d$,

$$f(u_0 + h) = f(u_0) + g^\top h + \frac{1}{2}h^\top H h + r(h), \tag{29}$$

where

$$|r(h)| \leq \frac{L}{6}\|h\|^3. \tag{30}$$

Proof. By the integral form of Taylor's theorem,

$$f(u_0 + h) = f(u_0) + g^\top h + \int_0^1 (1 - t)\, h^\top \nabla^2 f(u_0 + th)\, h\, dt.$$

Subtracting $\frac{1}{2}h^\top H h = \int_0^1 (1-t) h^\top H h\, dt$ gives

$$r(h) = \int_0^1 (1 - t)\, h^\top \big(\nabla^2 f(u_0 + th) - H\big) h\, dt.$$

Hence

$$|r(h)| \leq \int_0^1 (1 - t)\, \|\nabla^2 f(u_0 + th) - H\|\, \|h\|^2\, dt \leq \int_0^1 (1 - t)\, Lt\, dt\, \|h\|^3 = \frac{L}{6}\|h\|^3.$$

□

Lemma E.4 (Gaussian moment bounds)**.** *Let $\Delta \sim \mathcal{N}(0, S)$ in $\mathbb{R}^d$ with $S \succeq 0$. For every integer $k \geq 1$,*

$$\mathbb{E}\|\Delta\|_2^k \leq C_{d,k}\, \|S\|^{k/2}, \tag{31}$$

where $C_{d,k} := \mathbb{E}\|G\|_2^k < \infty$ for $G \sim \mathcal{N}(0, I_d)$.

Proof. Write $\Delta = S^{1/2}G$ with $G \sim \mathcal{N}(0, I_d)$. Then

$$\|\Delta\|_2 = \|S^{1/2}G\|_2 \leq \|S^{1/2}\|\, \|G\|_2 = \|S\|^{1/2}\|G\|_2.$$

Raising both sides to the kth power and taking expectation yields

$$\mathbb{E}\|\Delta\|_2^k \leq \|S\|^{k/2}\, \mathbb{E}\|G\|_2^k = C_{d,k}\, \|S\|^{k/2}.$$

□

Lemma E.5 (Odd symmetry for centered Gaussians)**.** *Let $\Delta \sim \mathcal{N}(0, S)$ in $\mathbb{R}^d$. If $h : \mathbb{R}^d \to \mathbb{R}$ is measurable, integrable, and odd, i.e. $h(-u) = -h(u)$ for all u, then*

$$\mathbb{E}[h(\Delta)] = 0. \tag{32}$$

In particular, for every $g \in \mathbb{R}^d$ and every symmetric $H \in \mathbb{R}^{d\times d}$,

$$\mathbb{E}\big[(g^\top \Delta)(\Delta^\top H \Delta)\big] = 0. \tag{33}$$

Proof. Since Δ is centered Gaussian, $\Delta \overset{d}{=} -\Delta$. Therefore

$$\mathbb{E}[h(\Delta)] = \mathbb{E}[h(-\Delta)] = -\mathbb{E}[h(\Delta)],$$

which implies (32). For (33), note that

$$u \mapsto (g^\top u)(u^\top H u)$$

is odd because the linear factor changes sign and the quadratic factor does not. □

F Proofs for Section 5

In this section, we prove our theorems and corollaries.

F.1 Proof of Theorem 5.3

By the definition of the DnNP predictive class,

$$q^{\mathrm{DnNP}}_{\theta^\star,\psi^\star}(y \mid x, z) = \int \mathcal{N}\big(y \mid f_{\theta^\star}(x^*, z), \sigma_y^2\big)\, q_{\psi^\star}(x^* \mid x)\, dx^*.$$

Using the realizability assumptions (14),

$$q^{\mathrm{DnNP}}_{\theta^\star,\psi^\star}(y \mid x, z) = \int \mathcal{N}\big(y \mid f^\star(x^*, z), \sigma_y^2\big)\, p(x^* \mid x)\, dx^*.$$

The right-hand side is exactly the Bayes-optimal predictive conditional from Section 5, proving (15). The KL divergence therefore vanishes pointwise, and taking expectation over (x, Z) gives (16).

F.2 Proof of Theorem 5.2

Fix (x, z) and abbreviate

$$P := P_{x,z}, \qquad Q := \mathcal{N}(m(x,z), \sigma_y^2), \qquad \mu_\star := \mu_\star(x,z), \qquad v_\star := v_\star(x,z), \qquad m := m(x,z).$$

By the Bayes-optimal predictive conditional, P is a Gaussian mixture and is therefore absolutely continuous. Let q denote the density of Q.

By definition,

$$\mathrm{KL}(P\|Q) = -H(P) - \mathbb{E}_{Y\sim P}[\log q(Y)].$$

Applying Lemma E.1 gives

$$-\mathbb{E}_{Y\sim P}[\log q(Y)] = \frac{1}{2}\log(2\pi\sigma_y^2) + \frac{v_\star + (\mu_\star - m)^2}{2\sigma_y^2}.$$

Applying Lemma E.2 to P yields

$$H(P) \le \frac{1}{2}\log(2\pi e\, v_\star).$$

Combining the last two displays,

$$\begin{aligned}\mathrm{KL}(P\|Q) &\ge -\frac{1}{2}\log(2\pi e\, v_\star) + \frac{1}{2}\log(2\pi\sigma_y^2) + \frac{v_\star + (\mu_\star - m)^2}{2\sigma_y^2} \\ &= \frac{(\mu_\star - m)^2}{2\sigma_y^2} + \frac{1}{2}\left(\frac{v_\star}{\sigma_y^2} - 1 - \log\frac{v_\star}{\sigma_y^2}\right).\end{aligned}$$

The right-hand side is minimized over m by choosing $m = \mu_\star$, proving (13).

It remains to verify the strict-positivity claim. By (10),

$$v_\star = \sigma_y^2 + \mathrm{Var}\big(f^\star(X, z) \mid x\big).$$

If $\mathrm{Var}(f^\star(X,z) \mid x) > 0$, then $v_\star > \sigma_y^2$. Define

$$\phi(u) := u - 1 - \log u, \qquad u > 0.$$

Then

$$\phi'(u) = 1 - \frac{1}{u}, \qquad \phi''(u) = \frac{1}{u^2} > 0.$$

Hence ϕ is strictly convex and attains its unique minimum 0 at $u = 1$. Therefore $\phi(u) > 0$ for every $u \ne 1$, and in particular

$$\frac{1}{2}\left(\frac{v_\star}{\sigma_y^2} - 1 - \log\frac{v_\star}{\sigma_y^2}\right) > 0 \qquad \text{whenever } v_\star > \sigma_y^2.$$

This completes the proof.

F.3 Proof of Lemma 5.4, Theorem 5.5, and Corollary 5.6

For brevity, write $\bar{x^*} := \mu_{\text{post}}(x)$, $S := \Sigma_{\text{post}}$, and $f(\cdot) := f^\star(\cdot, z)$. By Proposition 5.1,

$$X \mid x \sim \mathcal{N}(\bar{x^*}, S).$$

Define

$$\Delta := X - \bar{x^*}, \qquad g := \nabla f(\bar{x^*}), \qquad H := \nabla^2 f(\bar{x^*}).$$

Then $\Delta \sim \mathcal{N}(0, S)$.

Since $\nabla^2 f$ is globally Lipschitz, Lemma E.3 gives

$$f(\bar{x^*} + \Delta) = f(\bar{x^*}) + g^\top \Delta + \frac{1}{2}\Delta^\top H \Delta + r(\Delta), \tag{34}$$

with

$$|r(\Delta)| \le C\|\Delta\|^3 \tag{35}$$

for a constant C independent of Δ.

Proof of (17)**.** Taking expectation in (34) yields

$$\mathbb{E}[f(X) \mid x] = f(\bar{x^*}) + \mathbb{E}[g^\top \Delta] + \frac{1}{2}\mathbb{E}[\Delta^\top H \Delta] + \mathbb{E}[r(\Delta)].$$

Because Δ is centered Gaussian,

$$\mathbb{E}[g^\top \Delta] = 0, \qquad \mathbb{E}[\Delta^\top H \Delta] = \operatorname{tr}(HS).$$

Moreover, by (35) and Lemma E.4,

$$|\mathbb{E}[r(\Delta)]| \le C\,\mathbb{E}\|\Delta\|^3 = O(\|S\|^{3/2}).$$

Therefore

$$\mathbb{E}[f(X) \mid x] = f(\bar{x^*}) + \frac{1}{2}\operatorname{tr}(HS) + O(\|S\|^{3/2}),$$

which proves (17).

Proof of (18)**.** Define the centered quadratic and remainder terms

$$Q(\Delta) := \frac{1}{2}\Big(\Delta^\top H \Delta - \operatorname{tr}(HS)\Big), \qquad R(\Delta) := r(\Delta) - \mathbb{E}[r(\Delta)].$$

Then

$$f(X) - \mathbb{E}[f(X) \mid x] = g^\top \Delta + Q(\Delta) + R(\Delta).$$

Hence

$$\begin{aligned}\operatorname{Var}(f(X) \mid x) = {} & \mathbb{E}\big[(g^\top \Delta)^2\big] + \mathbb{E}\big[Q(\Delta)^2\big] + \mathbb{E}\big[R(\Delta)^2\big] \\ & + 2\mathbb{E}\big[g^\top \Delta\, Q(\Delta)\big] + 2\mathbb{E}\big[g^\top \Delta\, R(\Delta)\big] + 2\mathbb{E}\big[Q(\Delta) R(\Delta)\big].\end{aligned} \tag{36}$$

We now bound each term.

First,

$$\mathbb{E}[(g^\top \Delta)^2] = g^\top S g.$$

Second, since

$$|Q(\Delta)| \le \frac{1}{2}|\Delta^\top H \Delta| + \frac{1}{2}|\operatorname{tr}(HS)| \le \frac{1}{2}\|H\|\,\|\Delta\|^2 + \frac{1}{2}|\operatorname{tr}(HS)|,$$

we have

$$\mathbb{E}[Q(\Delta)^2] = O\big(\mathbb{E}\|\Delta\|^4\big) + O(\|S\|^2) = O(\|S\|^2),$$

where Lemma E.4 was used in the last step.

Third, by (35),

$$|R(\Delta)| \le |r(\Delta)| + |\mathbb{E}[r(\Delta)]| \le C\|\Delta\|^3 + C\mathbb{E}\|\Delta\|^3.$$

Therefore

$$\mathbb{E}[R(\Delta)^2] = O\big(\mathbb{E}\|\Delta\|^6\big) = O(\|S\|^3)$$

by Lemma E.4.

Fourth, by Lemma E.5,

$$\mathbb{E}[g^\top \Delta\, Q(\Delta)] = \frac{1}{2}\mathbb{E}\big[(g^\top \Delta)(\Delta^\top H \Delta)\big] - \frac{1}{2}\mathrm{tr}(HS)\,\mathbb{E}[g^\top \Delta] = 0.$$

Fifth, by Cauchy–Schwarz,

$$\big|\mathbb{E}[g^\top \Delta\, R(\Delta)]\big| \le \Big(\mathbb{E}[(g^\top \Delta)^2]\Big)^{1/2}\Big(\mathbb{E}[R(\Delta)^2]\Big)^{1/2} = O(\|S\|^{1/2})\, O(\|S\|^{3/2}) = O(\|S\|^2).$$

Sixth, again by Cauchy–Schwarz,

$$\big|\mathbb{E}[Q(\Delta)R(\Delta)]\big| \le \Big(\mathbb{E}[Q(\Delta)^2]\Big)^{1/2}\Big(\mathbb{E}[R(\Delta)^2]\Big)^{1/2} = O(\|S\|)\, O(\|S\|^{3/2}) = O(\|S\|^{5/2}).$$

Substituting these bounds into (36) gives

$$\mathrm{Var}(f(X) \mid x) = g^\top S g + O(\|S\|^2),$$

which proves (18).

Proof of (19)**.** Let $m_0 := f(\bar{x^*})$. By Theorem 5.2,

$$\mathrm{KL}\big(P_{x,z} \,\|\, \mathcal{N}(m_0, \sigma_y^2)\big) \ge \frac{(\mu_\star - m_0)^2}{2\sigma_y^2} + \frac{1}{2}\left(\frac{v_\star}{\sigma_y^2} - 1 - \log\frac{v_\star}{\sigma_y^2}\right). \tag{37}$$

By (17),

$$\mu_\star - m_0 = \frac{1}{2}\mathrm{tr}(HS) + O(\|S\|^{3/2}),$$

so

$$\frac{(\mu_\star - m_0)^2}{2\sigma_y^2} = \frac{1}{8\sigma_y^2}\big(\mathrm{tr}(HS)\big)^2 + o(\|S\|^2). \tag{38}$$

Next, let $a := v_\star - \sigma_y^2$. By (18),

$$a = g^\top S g + O(\|S\|^2), \qquad a = O(\|S\|).$$

Write $u := a/\sigma_y^2$. Then $u = O(\|S\|)$, and

$$\frac{1}{2}\left(\frac{v_\star}{\sigma_y^2} - 1 - \log\frac{v_\star}{\sigma_y^2}\right) = \frac{1}{2}\big(u - \log(1+u)\big).$$

Since $u \to 0$, Taylor expansion of $\log(1+u)$ at 0 gives

$$u - \log(1+u) = \frac{u^2}{2} + O(u^3),$$

hence

$$\frac{1}{2}\left(\frac{v_\star}{\sigma_y^2} - 1 - \log\frac{v_\star}{\sigma_y^2}\right) = \frac{1}{4\sigma_y^4}(g^\top S g)^2 + o(\|S\|^2). \tag{39}$$

Combining (37), (38), and (39) proves (19).

Proof of (20) **and Corollary 5.6.** (20) follows directly from (13) and the expansion (39). If $S = \tau^2 I_d$, then

$$g^\top S g = \tau^2 \|g\|_2^2,$$

so the isotropic scaling stated in Corollary 5.6 follows immediately.

G Experimental Detail

All experiments were run on a single NVIDIA H200 GPU on an internal computing cluster. The training and inference time of DnNP is approximately the same as that of standard NPs, since the additional denoiser is a small MLP applied per input and contributes negligible overhead relative to the encoder and decoder networks.

Table 3: Overview of datasets used in experiments. d denotes the input dimensionality, K the number of training tasks (episodes), Task specifies the type of problem (regression, image completion, or spatial interpolation), and Noise indicates whether inputs are naturally corrupted. MNIST and 1D GP datasets do not inherently contain input noise, so we add noise synthetically to simulate noisy measurements.

DATASET	K	d	TASK	NOISE
1D GP	10,000	1	REGRESSION	NO
MNIST	70,000	2	IMAGE COMPL.	NO
AIR QUALITY	2,856	4	SPATIAL INTERP.	YES
NOAA CA 2026	90	4	SPATIAL INTERP.	YES

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The contributions are listed in Section 1; each is addressed in detail in the corresponding section of the paper.

   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: Limitations are discussed in Appendix A and computational efficiency in Appendix G.

   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: The assumptions are discussed in Section 5, and complete proofs are provided in the appendix.

   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: All information needed to reproduce our main experimental results, including dataset preprocessing, training hyperparameters, and evaluation protocols, is provided in Section 6 and Appendix G.

   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: The air quality dataset is publicly available at `https://hdl.handle.net/2047/D20881761`, and the NOAA data are publicly available via the Meteostat library. Code to reproduce all experiments is available at `https://github.com/iSUPER-ML/DnNP`.

   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: The information can be found in Section 6.

   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: Table 2 reports mean ± standard deviation over 5 seeds for the real-world experiments.

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: The information can be found in Appendix G.

   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:

   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: the information can be found in Appendix B.

    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:

    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:

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:

    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:

    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:

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:

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.