Neural Mixture Density Processes

Supplementary Material

A. Exchangeable Stochastic Processes

Definition 1 (Exchangeable Stochastic Process) *Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space defined over functions. For a finite index set $\{x_1, \ldots, x_n\} \subset \mathcal{X}$, let $\nu_{x_1,\ldots,x_n}$ represent the corresponding probability measure on $\mathbb{R}^d$. A probabilistic model ρ is termed an exchangeable stochastic process $\mathcal{S} : \mathcal{X} \times \Omega \to \mathbb{R}^d$ if it satisfies $\nu_{x_1,\ldots,x_n}(\mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n) = \mathbb{P}(\mathcal{S}_{x_1} \in \mathcal{Y}_1, \ldots, \mathcal{S}_{x_n} \in \mathcal{Y}_n)$, and adheres to the principles of exchangeability and marginalization consistency.*

For clarity, we denote the exchangeable stochastic process ($\mathcal{SP}$) in **Definition** 1 by $\rho_{x_{1:n}}(y_{1:n})$, where $x_{1:n}$ represents a collection of indices in the domain $\mathcal{X}$ and $y_{1:n}$ the corresponding random variables in $\mathcal{Y}$. According to the Kolmogorov extension theorem [41], the process must satisfy the following fundamental consistency conditions:

- *Exchangeable Consistency.* Given any n data points and any permutation operator $\pi : [1, \ldots, n] \to [\pi_1, \ldots, \pi_n]$, the marginal distribution $\rho_{x_{1:n}}(y_{1:n})$ is invariant:

$$\rho_{x_{1:n}}(y_{1:n}) = \rho_{x_{\pi(1:n)}}(y_{\pi(1:n)}). \tag{12}$$

- *Marginalization Consistency.* Given any numbers of data points n and $n + m$, the distribution $\rho_{x_{1:n+m}}(y_{1:n+m})$ retains its marginal form after integral as:

$$\int \rho_{x_{1:n+m}}(y_{1:n+m}) dy_{n+1:n+m} = \rho_{x_{1:n}}(y_{1:n}). \tag{13}$$

Building on these definitions and consistency principles, we now introduce the Neural Process framework and its associated inference formulation.

B. Probabilistic Generative Process in NMDP

Here we translate the generative process of meta-learning mixture density networks into the following mathematical way:

$$\begin{aligned} \tau \sim p(\mathcal{T}), \quad z \sim \mathsf{Dir}(z; \alpha_\tau), \quad l_i \sim \mathsf{Cat}(z) \\ x_i \sim p(x), \quad y_i \sim p_{\psi_l}(y|x_i) \quad \forall i \in \{1, 2, \ldots, n+m\} \end{aligned} \tag{14}$$

Meanwhile, we provide a fine-grained probabilistic graphical models in Fig. 7.

C. EM/MM-Style Optimization with Importance Sampling

In this section, we provide additional derivation details for the optimization objective used in the main paper. We first review the optimization objective in ELBO-based variational methods, and then present the EM/MM-style surrogate objective adopted by the NMDP. Finally, we derive its self-normalized importance sampling approximation and the corresponding gradient estimators.

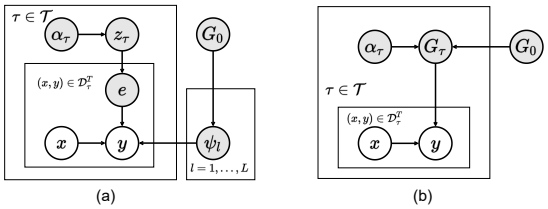


Figure 7. Probabilistic Graphical Models of NMDP from a Dirichlet Process Perspective. (a) describes the probabilistic dependencies in a Bayesian mixture model framework, with e a categorical latent variable from $\mathsf{Cat}(e; z_\tau)$ to select the probabilistic component p_{ψ_l} in generation. (b) is from a probability measure perspective and G_τ defines the mixture density distribution for the task τ.

C.1. Prior and Proposal Distributions

Since fast adaptation is achieved in an amortized way, the context set is encoded by a permutation-invariant neural network. For example, to parameterize a Dirichlet distribution $\mathsf{Dir}(z; \alpha_\tau(\mathcal{D}_\tau^C))$ from the input $\mathcal{D}_\tau^C = \{[x_i, y_i]\}_{i=1}^n$, the amortized network can take the following form:

$$\begin{aligned} r_i = g_1\left([x_i, y_i]\right), \quad [\hat{\alpha}, \beta] = g_2\left(\bigoplus_{i=1}^n r_i\right), \\ \alpha_\tau = \mathsf{Softmax}(\hat{\alpha}) \cdot \mathsf{Softplus}(\beta), \end{aligned} \tag{15}$$

where $\bigoplus$ denotes a permutation-invariant aggregation operator.

Approximate Posterior Distribution. In standard neural processes [14, 25], an approximate posterior $q_\phi(z|\mathcal{D}_\tau^T)$ is introduced as a proxy for the intractable true posterior.

Prior Distribution. In the NMDP, the context-conditioned prior is

$$p(z|\mathcal{D}_\tau^C; \eta) = \mathsf{Dir}(z; \alpha_\tau(\mathcal{D}_\tau^C; \eta)). \tag{16}$$

Unlike the approximate prior used in standard neural processes, this prior is directly parameterized by the context set.

Proposal Distribution. In our method, the proposal distribution used for importance sampling is chosen to be the current context-conditioned prior:

$$r_k(z) = p(z|\mathcal{D}_\tau^C; \eta_k). \tag{17}$$

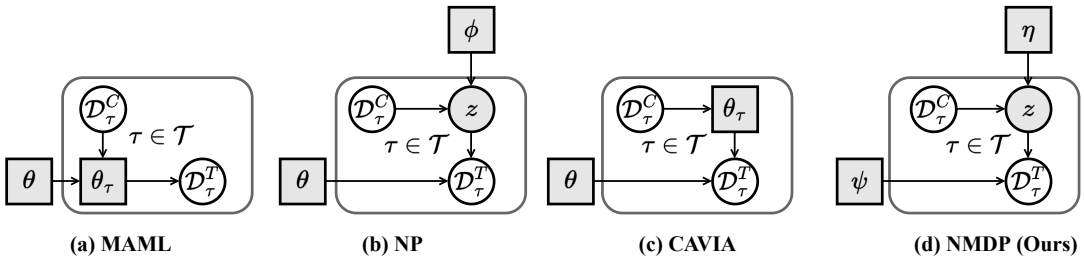


Figure 8. **Probabilistic Graphical Models of the Typical Meta-Learning Methods during meta-testing.** Unobserved variables are marked in grey, and random variables are in circles. For MAML and CAVIA, θ_τ is the model parameter after gradient based fast adaptation. For the NP and the NMDP, z is the latent variable representing the meta-learning task.

This choice is consistent with the optimization procedure in the main text, where latent particles are sampled from the prior inferred from the context set.

C.2. Approximation Gap in Variational Inference

In typical variational autoencoder-related models, the optimization objective is an evidence lower bound (ELBO). The prior distribution $p_\theta(z|\mathcal{D}_\tau^C)$ can be either a fixed distribution, *e.g.* a standard Gaussian $\mathcal{N}(z; 0, \mathcal{I})$, or an amortized prior as in neural processes [14, 25]. Given an auxiliary variational distribution $q_\phi(z)$, one has

$$\begin{aligned}\ln p_\theta(\mathcal{D}_\tau^T|\mathcal{D}_\tau^C) &= \mathbb{E}_{q_\phi(z)}\left[\ln p_\theta(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C) - \ln q_\phi(z)\right] \\ &\quad + D_{KL}\left[q_\phi(z) \parallel p_\theta(z|\mathcal{D}_\tau^T, \mathcal{D}_\tau^C)\right] \\ &\geq \mathbb{E}_{q_\phi(z)}\left[\ln p_\theta(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C) - \ln q_\phi(z)\right] \\ &= \text{ELBO}.\end{aligned} \tag{18}$$

Recall the notation in the main paper:

$$p_\theta(\mathcal{D}_\tau^T|z) = \prod_{i=1}^{n+m} p_\theta(y_i|x_i, z). \tag{19}$$

When the target dataset is used to parameterize the variational posterior, this corresponds to $q_\phi(z) = q_\phi(z|\mathcal{D}_\tau^T)$.

Compared with ELBO-based training, our method does not introduce an additional amortized target-conditioned proposal network. Instead, it directly constructs an EM/MM-style surrogate objective for the exact posterior under the current model parameters and approximates it with self-normalized importance sampling.

C.3. Proof of Improvement Guarantee using Variational Expectation Maximization

Definition 2 (Surrogate Function).[1] *Given an objective function $f(\Theta)$ to maximize, a surrogate function $g(\Theta; \Theta_k)$ is said to minorize $f(\Theta)$ if*

$$g(\Theta; \Theta_k) \leq f(\Theta), \qquad g(\Theta_k; \Theta_k) = f(\Theta_k). \tag{20}$$

When the original objective is difficult to optimize directly, a surrogate function provides a tractable proxy objective. If one updates $\Theta_{k+1} = \arg\max_\Theta g(\Theta; \Theta_k)$, then

$$f(\Theta_{k+1}) \geq g(\Theta_{k+1}; \Theta_k) \geq g(\Theta_k; \Theta_k) = f(\Theta_k). \tag{21}$$

Thus, exact surrogate maximization leads to the standard monotonic-improvement property.

For the NMDP, the ideal EM surrogate for a batch of tasks $\mathcal{T}$ can be written as

$$\mathcal{L}(\Theta; \Theta_k) = \sum_{\tau \in \mathcal{T}} \mathbb{E}_{p(z|\mathcal{D}_\tau^T, \mathcal{D}_\tau^C; \Theta_k)}\left[\ln p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C; \Theta)\right]. \tag{22}$$

Under exact posterior evaluation and exact maximization of the surrogate, the corresponding EM/MM-style procedure enjoys the standard monotonic-improvement property for the marginal log-likelihood. In practice, however, the posterior expectation in Eq. (22) is intractable, and the surrogate is only optimized approximately via gradient-based updates. For this reason, the actual training algorithm in the main text should be understood as an approximate EM/MM-style procedure rather than exact EM.

C.4. Importance Sampling in a Variational EM Algorithm

The conditional marginal likelihood $p(\mathcal{D}_\tau^T|\mathcal{D}_\tau^C; \Theta_k)$ is generally intractable. To approximate the posterior expectation in Eq. (22), we employ importance sampling using the current prior $r_k(z) = p(z|\mathcal{D}_\tau^C; \eta_k)$ as the proposal distribution.

[1] Here the surrogate function is defined in the Minorize-Maximization (MM) sense for a maximization problem.

Then

$$p(\mathcal{D}_\tau^T|\mathcal{D}_\tau^C;\Theta_k) = \int p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta_k)\,dz$$
$$= \int r_k(z)\,\frac{p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta_k)}{r_k(z)}\,dz$$
$$= \int p(z|\mathcal{D}_\tau^C;\eta_k)\,p(\mathcal{D}_\tau^T|z;\psi_k)\,dz$$
$$\approx \frac{1}{B}\sum_{b=1}^{B} p(\mathcal{D}_\tau^T|z^{(b)};\psi_k), \tag{23}$$
$$z^{(b)} \sim p(z|\mathcal{D}_\tau^C;\eta_k). \tag{24}$$

Since the proposal distribution is chosen as the current prior, the unnormalized importance weight for particle $z^{(b)}$ simplifies to

$$\omega_k^{(b)} = \frac{p(\mathcal{D}_\tau^T, z^{(b)}|\mathcal{D}_\tau^C;\Theta_k)}{p(z^{(b)}|\mathcal{D}_\tau^C;\eta_k)} = p(\mathcal{D}_\tau^T|z^{(b)};\psi_k). \tag{25}$$

The corresponding normalized importance weights are

$$\hat{\omega}_k^{(b)} = \frac{\omega_k^{(b)}}{\sum_{b'=1}^{B}\omega_k^{(b')}}. \tag{26}$$

The target-set likelihood for a sampled latent variable $z^{(b)}$ is

$$p(\mathcal{D}_\tau^T|z^{(b)};\psi_k) = \prod_{i=1}^{n+m}\left(\sum_{l=1}^{L} z_l^{(b)} p_{\psi_l^k}(y_i|x_i)\right), \tag{27}$$

where $\psi^k = \{\psi_l^k\}_{l=1}^L \subseteq \Theta_k$.

Using the above importance weights, the intractable surrogate objective can be approximated as follows:

$$\mathcal{L}_{\mathsf{EM}}(\Theta;\Theta_k) = \mathbb{E}_{p(z|\mathcal{D}_\tau^T,\mathcal{D}_\tau^C;\Theta_k)}\left[\ln p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta)\right] \tag{28a}$$
$$= \int \frac{p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta_k)}{p(\mathcal{D}_\tau^T|\mathcal{D}_\tau^C;\Theta_k)} \ln p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta)\,dz \tag{28b}$$
$$= \frac{1}{p(\mathcal{D}_\tau^T|\mathcal{D}_\tau^C;\Theta_k)} \int p(z|\mathcal{D}_\tau^C;\eta_k)\,p(\mathcal{D}_\tau^T|z;\psi_k) \ln p(\mathcal{D}_\tau^T, z|\mathcal{D}_\tau^C;\Theta)\,dz \tag{28c}$$
$$\approx \sum_{b=1}^{B} \hat{\omega}_k^{(b)} \ln p(\mathcal{D}_\tau^T, z^{(b)}|\mathcal{D}_\tau^C;\Theta) \tag{28d}$$
$$= \mathcal{L}_{\mathsf{IW\text{-}MC}}(\Theta;\Theta_k,B). \tag{28e}$$

Eq. (28d) is the self-normalized importance sampling estimator of the EM surrogate in Eq. (28a).

We can further expand the joint distribution as

$$p(\mathcal{D}_\tau^T, z^{(b)}|\mathcal{D}_\tau^C;\Theta) = p(z^{(b)}|\mathcal{D}_\tau^C;\eta)\,p(\mathcal{D}_\tau^T|z^{(b)};\psi)$$
$$= \mathsf{Dir}(z^{(b)};\alpha_\tau(\mathcal{D}_\tau^C;\eta))\prod_{i=1}^{n+m}\left(\sum_{l=1}^{L} z_l^{(b)} p_{\psi_l}(y_i|x_i)\right). \tag{29}$$

The probability density function of the Dirichlet distribution is

$$\mathsf{Dir}(z_\tau;\alpha_\tau) = \frac{\Gamma(\sum_{l=1}^{L}\alpha_{\tau,l})}{\prod_{l=1}^{L}\Gamma(\alpha_{\tau,l})}\prod_{l=1}^{L} z_{\tau,l}^{\alpha_{\tau,l}-1}, \tag{30}$$

where $\{\alpha_{\tau,l}\}_{l=1}^L$ is a set of concentration parameters satisfying $\alpha_{\tau,l} > 0$, and

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx. \tag{31}$$

The exact posterior under Θ_k is

$$p(z|\mathcal{D}_\tau^T, \mathcal{D}_\tau^C;\Theta_k) = \frac{p(z|\mathcal{D}_\tau^C;\Theta_k)\,p(\mathcal{D}_\tau^T|z;\Theta_k)}{\int p(z|\mathcal{D}_\tau^C;\Theta_k)\,p(\mathcal{D}_\tau^T|z;\Theta_k)\,dz}. \tag{32}$$

The denominator is generally intractable, which makes direct posterior sampling infeasible and motivates the importance sampling approximation above.

C.5. Optimization Objective with Self-Normalized Importance Sampling

By factorizing the joint distribution, the Monte Carlo surrogate in Eq. (28d) can be rewritten as

$$\mathcal{L}_{\mathsf{IW\text{-}MC}}(\Theta;\Theta_k,B) = \sum_{b=1}^{B}\hat{\omega}_k^{(b)}\left[\ln p(z^{(b)}|\mathcal{D}_\tau^C;\eta) + \ln p(\mathcal{D}_\tau^T|z^{(b)};\psi)\right]. \tag{33}$$

This is the particle-based objective used in the main text. It consists of a prior term and a generative likelihood term, both weighted by self-normalized importance weights computed from the previous iterate Θ_k.

C.6. Gradient Estimates in Variational EM

In practice, the normalized importance weights are computed using the previous parameters Θ_k and are treated as constants during the current parameter update. Accordingly, the gradient with respect to η is

$$\frac{\partial\mathcal{L}_{\mathsf{IW\text{-}MC}}(\Theta;\Theta_k,B)}{\partial\eta} = \sum_{b=1}^{B}\hat{\omega}_k^{(b)}\frac{\partial\ln p(z^{(b)}|\mathcal{D}_\tau^C;\eta)}{\partial\eta}, \tag{34}$$

and the gradient with respect to ψ is

$$\frac{\partial\mathcal{L}_{\mathsf{IW\text{-}MC}}(\Theta;\Theta_k,B)}{\partial\psi} = \sum_{b=1}^{B}\hat{\omega}_k^{(b)}\frac{\partial\ln p(\mathcal{D}_\tau^T|z^{(b)};\psi)}{\partial\psi}. \tag{35}$$

Averaging the above gradients over a batch of tasks gives the update rule in Algorithm 1 of the main paper.

C.7. Expectation Maximization

The training procedure used in this work can be summarized as an approximate EM/MM-style iterative algorithm.

E-step. Using the previous parameters $\Theta_k = (\eta_k, \psi_k)$, sample particles $z^{(b)} \sim p(z|\mathcal{D}_\tau^C; \eta_k)$ and compute self-normalized importance weights $\{\hat{\omega}_k^{(b)}\}_{b=1}^B$.

M-step. Update the current parameters by approximately maximizing the surrogate objective:

$$\Theta_{k+1} \approx \arg\max_{\Theta} \mathcal{L}_{\text{IW-MC}}(\Theta; \Theta_k, B). \tag{36}$$

Therefore, the practical training algorithm should be viewed as an approximate EM/MM-style optimization procedure, where the posterior expectation is approximated by self-normalized importance sampling and the surrogate is optimized by gradient-based updates.

D. Experiments and Analysis

D.1. General Setting

Synthetic Regression We use the Gaussian process simulator to generate different stochastic functions. Three types of kernels are used to formulate diverse Gaussian processes. The following kernel functions specify different Gaussian processes in this paper.

- RBF kernel:

$$k(x, x^{'}) = s^2 \exp\left(-\frac{(x - x^{'})^2}{2l^2}\right) \tag{37}$$

  with $s \sim U\,[0.1, 1.0]$ and $l \sim U\,[0.1, 0.6]$
- Matern $-\frac{5}{2}$ kernel :

$$k(x, x') = s^2 \left(1 + \frac{\sqrt{5}d}{l} + \frac{5d^2}{3l^2}\right) exp\left(-\frac{\sqrt{5}r}{l}\right) \tag{38}$$

  with $d = 4|x - x'|$, $s \sim U\,[0.1, 1.0]$ and $l \sim U\,[0.1, 0.6]$
- Periodic kernel:

$$k(x, x') = s^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right) \tag{39}$$

  with $s \sim U\,[0.1, 1.0]$, $l \sim U\,[0.1, 0.6]$ and $p \sim U[0.1, 0.5]$

For each training batch, we independently sampled the numbers of context and target observations, $\mathcal{D}_\tau^C$ and $\mathcal{D}_\tau^T$, from a discrete uniform distribution over the integers 3 to 50. Conditional on these counts, we first draw a function realization from the underlying stochastic process, a Gaussian process with one of the aforementioned kernels, then draw $\mathcal{D}_\tau^C$ context and $\mathcal{D}_\tau^T$ target input locations independently and uniformly from the interval $[-2, 2]$, obtaining outputs by evaluating the sampled function at these inputs.

Image Completion We evaluate image completion according to the standard episodic Neural Process protocol [17], representing each image as a continuous function $f : [-1, 1]^2 \to [0, 1]^c$ that maps 2D pixel coordinates to normalized intensity values. We conduct experiments on four widely-adopted benchmark datasets: CIFAR-10 [29], SVHN [38] (3-channel RGB images), and MNIST [31], EMNIST [5] (1-channel grayscale images), where $c = 3$ for RGB or $c = 1$ for grayscale. Episodes are formed by sampling a context set comprising a random 5-20% subset of pixels, while the target set is the entire image; query coordinates are drawn uniformly from $[-1, 1]^2$ and responses are the corresponding normalized intensity values.

Multi-input-output Regression To assess the model's performance on complex, real-world problems, we evaluate it on several standard UCI datasets for multi-output regression. This setting tests the model's ability to capture intricate dependencies between multiple input and output variables. The datasets used are as follows:

- **SARCOS**: A robotics dataset with 48,933 samples, predicting 7 joint torques of a robot arm from 21 corresponding motor features (positions, velocities, and accelerations).
- **Water Quality (WQ)**: An environmental dataset with 1,060 instances, predicting 14 biological species' populations from 16 chemical and physical water quality parameters.
- **SCM20D**: A supply chain management dataset with 8,966 samples, characterized by 61 input features and 16 output variables.

For all datasets, we perform a standard 80/20 split to create training and test sets. To ensure fair comparison and stable training, all input and output features are standardized to have zero mean and unit variance. The statistics (mean and standard deviation) for standardization are computed *solely* from the training set to prevent information leakage from the test set.

Following the meta-learning protocol, tasks are constructed by sampling mini-batches from the training data. During training, each batch is randomly partitioned into a context set (50% of the batch) and a target set (the remaining 50%). This forces the model to learn to make predictions for a subset of targets given another subset of observations from the same task. During evaluation, a fixed random split of the test set is used to measure the model's generalization performance on unseen data.

D.2. Neural Architecture

Our primary model is an attentive Neural Mixture Density Process (NMDP): an attentive mixture-of-experts Neural Process in which a Dirichlet latent variable $z \in \Delta^{L-1}$ governs the mixture weights. The architecture is composed of two main modules: the prior inference network (Fig. 1a) for inferring the Dirichlet parameters, and the generative network (Fig. 1b) for producing target predictions. Unless

Table 4. Hyperparameter configurations for different experimental settings. The latent dimension d_{model} is fixed to 128 for all experiments.

Hyperparameter	Description	1D Synthetic	Image (RGB)	Image (Gray)	Multi-Output
# Mixture Components (L)	Number of expert networks in the mixture	5	9	7	15
# Importance Particles (B)	Particles for IWAE objective estimation	20	30	30	35
# Context Points ($\lvert\mathcal{D}_\tau^C\rvert$)	Size of the context set for adaptation	$\mathcal{U}[3, 50]$	$\mathcal{U}[50, 200]$ (5-20% pixels)	$\mathcal{U}[50, 200]$ (5-20% pixels)	50% of training data
# Target Points ($\lvert\mathcal{D}_\tau^T\rvert$)	Size of the target set for prediction	$\mathcal{U}[3, 50]$	1024 (full image)	1024 (full image)	50% of training data
Optimizer	Optimization algorithm	Adam			
Learning Rate	Initial learning rate	5×10^{-4} (with cosine annealing)			
Batch Size	Number of tasks per batch	32	16	16	8
Encoder/Decoder Widths	Hidden units in MLPs	128			
Attention Heads	Heads in multi-head attention	8			
Input Domain (x)	Normalized range for input coordinates	$[-2, 2]$	$[-1, 1]^2$	$[-1, 1]^2$	Dataset-specific
Output Domain (y)	Normalized range for target values	$\mathbb{R}$	$[0, 1]^3$	$[0, 1]$	Dataset-specific

otherwise specified, we use $L = 5$–15 mixture components, hidden representation widths of 128–256, and $B = 15$–40 importance particles. Learning proceeds by maximizing a self-normalized importance-weighted surrogate objective that combines a Dirichlet log-prior term with a mixture likelihood term. We provide the detailed architectural specifications below.

Prior Inference Network (ContextNet)

The prior inference network, denoted as ContextNet in Fig. 1a, processes the context set $\mathcal{D}_\tau^C = \{(x_i, y_i)\}_{i=1}^n$ to parameterize the prior

$$p_\eta(z \mid \mathcal{D}_\tau^C) = \mathsf{Dir}(z; \alpha(\mathcal{D}_\tau^C)).$$

This network is permutation-invariant and consists of the following stages:

1. **Per-point Embedder (EmbNet):** Each context point (x_i, y_i) is independently encoded by a Multi-Layer Perceptron (MLP). The input dimension is $\dim(x) + \dim(y)$. The MLP consists of 3 hidden layers with 128 units each, separated by ReLU activations. The output is a $d_{\text{model}} = 128$-dimensional representation r_i for each point:
$$r_i = \text{EmbNet}([x_i, y_i]).$$
2. **Set Aggregator:** The set of representations $\{r_i\}_{i=1}^n$ is aggregated into a single global representation $R \in \mathbb{R}^{d_{\text{model}}}$ using a standard Transformer encoder block. This block contains a multi-head self-attention layer (with 8 heads), followed by a position-wise feed-forward network (two layers with 512 and 128 units). The output of the Transformer is mean-pooled across the n points to obtain the final permutation-invariant summary:
$$R = \text{MeanPool}(\text{Transformer}(\{r_i\}_{i=1}^n)).$$
3. **Dirichlet Parameter Head:** The global summary R is passed to a final MLP head to predict the concentration parameters $\alpha = [\alpha_1, \ldots, \alpha_L]$. This head has two hidden layers of sizes 128 and 64 with ReLU activations. The output layer has L units and uses a `Softplus` activation to ensure positivity:
$$\alpha = \text{Softplus}(\text{MLP}_\alpha(R)), \qquad \alpha_l > 0.$$

During training, the current prior $p_{\eta_k}(z \mid \mathcal{D}_\tau^C)$ is also reused as the proposal distribution for self-normalized importance sampling, consistent with the optimization procedure described in Sec 4.2 and Appendix C.

Generative Network (Density Experts)

The generative network, shown in Fig. 1b, takes a target location x^*, the context set $\mathcal{D}_\tau^C$, and a sampled latent variable z to produce expert-specific likelihood parameters, which are then combined into the predictive mixture distribution $p(y^* \mid x^*, \mathcal{D}_\tau^C, z)$.

1. **Target Embedder:** Each target coordinate x^* is embedded by an MLP with 3 hidden layers of 128 units each (ReLU activations), producing a query vector $q \in \mathbb{R}^{d_{\text{model}}}$:
$$q = \text{MLP}_q(x^*).$$
2. **Cross-Attention:** To incorporate context information, the query vector q attends to the context point representations $\{r_i\}_{i=1}^n$ computed by the context encoder. A standard multi-head cross-attention mechanism (with 8 heads) is used, where q serves as the query and $\{r_i\}_{i=1}^n$ serve as keys and values. This yields a context-aware query summary $c \in \mathbb{R}^{d_{\text{model}}}$:
$$c = \text{CrossAttention}(q, \{r_i\}_{i=1}^n).$$
3. **Mixture of Experts (Density Experts):** The model contains L expert networks, $\{\text{DensityExperts}_l\}_{l=1}^L$. Each expert is an MLP that takes the concatenation of the target coordinate x^* and the context-aware summary c as input. Each expert has 3 hidden layers of 128 units (ReLU activations) and outputs the parameters of a Gaussian likelihood. Concretely, each expert predicts a mean $\mu_l(x^*)$ and a scale parameter $\sigma_l(x^*)$:
$$[\mu_l, \sigma_l] = \text{DensityExperts}_l([x^*, c]).$$

A `Softplus` activation is applied to σ_l to ensure positivity. For image completion tasks, a `Sigmoid` activation is additionally applied to μ_l to match the normalized pixel range $[0, 1]$; for regression tasks, the predictive mean uses an identity output.

4. **Mixture Formation:** The final predictive distribution for a single target point is formed by mixing the outputs of all L experts using the latent weights z:

$$p(y^* \mid x^*, \mathcal{D}_\tau^C, z) = \sum_{l=1}^{L} z_l \, p_{\psi_l}(y^* \mid x^*),$$

where, in the Gaussian case,

$$p_{\psi_l}(y^* \mid x^*) = \mathcal{N}\big(y^*; \mu_l(x^*), \sigma_l^2(x^*)\big) \, .$$

Table 5. Target RMSE on 1D Synthetic Regression Tasks. Lower is better. Results are averaged over 5 runs.

Method	RBF	Matérn$-\frac{5}{2}$	Periodic
MAML	0.234$\pm$2e-3	0.283$\pm$4e-3	0.723$\pm$6e-3
CAVIA	0.212$\pm$3e-3	0.271$\pm$6e-3	0.697$\pm$2e-3
CNP	0.263$\pm$5e-3	0.256$\pm$4e-3	0.762$\pm$7e-3
ANP	0.227$\pm$2e-3	0.241$\pm$3e-3	0.709$\pm$2e-3
ConvCNP	0.198$\pm$1e-3	0.225$\pm$2e-3	0.695$\pm$2e-3
TNP	**0.177**$\pm$1e-3	0.222$\pm$1e-3	0.660$\pm$3e-3
DNP	0.195$\pm$1e-3	0.230$\pm$3e-3	0.674$\pm$5e-3
NMDP (Ours)	0.189$\pm$1e-3	**0.221**$\pm$2e-3	**0.656**$\pm$4e-3

Table 6. Target RMSE on Image Completion Tasks. Lower is better. Results are averaged over 5 runs.

Method	SVHN	CIFAR-10	MNIST	EMNIST
MAML	0.0118$\pm$6e-4	0.0115$\pm$7e-4	0.141$\pm$6e-3	0.134$\pm$3e-3
CAVIA	0.0099$\pm$5e-4	0.0060$\pm$5e-4	0.126$\pm$2e-3	0.127$\pm$6e-3
CNP	0.0114$\pm$3e-4	0.0061$\pm$2e-4	0.129$\pm$3e-3	0.133$\pm$1e-3
ANP	0.0097$\pm$7e-4	0.0057$\pm$3e-4	**0.121**$\pm$1e-3	0.127$\pm$5e-3
ConvCNP	0.0100$\pm$3e-4	0.0062$\pm$5e-4	0.126$\pm$1e-3	0.131$\pm$4e-3
TNP	0.0096$\pm$2e-4	0.0057$\pm$7e-4	0.119$\pm$2e-3	**0.122**$\pm$3e-3
DNP	0.0104$\pm$8e-4	0.0061$\pm$9e-4	**0.121**$\pm$5e-3	0.126$\pm$2e-3
NMDP (Ours)	**0.0094**$\pm$5e-4	**0.0055**$\pm$3e-4	0.123$\pm$3e-4	0.129$\pm$1e-4

For all experiments, we set the latent dimension $d_{\text{model}} = 128$. The number of mixture components L and importance particles B are task-specific and detailed in Table 4. All networks are trained end-to-end using the Adam optimizer with a learning rate of 5×10^{-4} and a cosine annealing schedule.

D.3. Additional Experiments

In the main text, we mainly established NMDP's strength in uncertainty quantification using log-likelihood. Here, we complement this by evaluating predictive accuracy from a more intuitive standpoint using Root Mean Squared Error (RMSE) on the target set. This metric directly assesses the model's generalization capability on unseen data. We present results for the 1D Synthetic Regression and Image Completion experiments.

1D Synthetic Regression

We evaluate the RMSE on the target set for 1D synthetic functions from three GP kernels. As shown in Table 5, NMDP achieves the lowest RMSE on two out of three kernels (Matérn$-\frac{5}{2}$ and Periodic) and remains highly competitive on the RBF kernel, where it is narrowly outperformed by TNP. This demonstrates NMDP's robust predictive accuracy across diverse function structures.

Image Completion

We next evaluate target RMSE for image completion on SVHN, CIFAR-10, MNIST, and EMNIST. Table 6 shows that NMDP achieves state-of-the-art results on the more complex, color-rich datasets (SVHN, CIFAR-10). On simpler grayscale datasets (MNIST, EMNIST), its performance is competitive with the strongest baselines. This indicates NMDP's particular strength in handling high-dimensional, unstructured data.

D.4. Hyperparameter Sensitivity Analysis

Sensitivity to Context Sparsity. To study robustness under limited observations, we vary the context ratio on CIFAR-10 image completion and report target predictive log-likelihood. Table 7 shows that NMDP consistently outperforms strong NP baselines across all context ratios, with the largest gains in the most sparse regimes (3% and 5%). This behavior is consistent with our modeling choice: under scarce or weakly diagnostic context, the context encoder tends to output a diffuse Dirichlet prior (high-entropy α), reflecting increased epistemic uncertainty about task identity. Consequently, the predictive distribution in Eq. 10 approaches a more averaged mixture over experts, yielding conservative predictions and avoiding overconfident reconstructions. We note that this also highlights an inherent limitation of context-conditioned gating: if the context contains insufficient information to disambiguate expert families, any gating mechanism will degenerate toward averaging. Overall, the results suggest that NMDP remains robust in sparse-context regimes while providing a principled uncertainty-aware response to ambiguous context.

Sensitivity to the Number of Experts. We further vary the number of experts L while fixing the context ratio to 10% on CIFAR-10. As shown in Table 8, increasing L improves performance initially and then saturates around $L \in \{9, 11\}$, suggesting that a moderate number of experts suffices to capture the dominant modes of the image completion task distribution. Very large L does not yield further gains and can

Table 7. Target Predictive Log-Likelihood (↑) under various context ratio on CIFAR-10 image completion tasks (10 runs).

Context Ratio	**3%**	**5%**	**10%**	**20%**
ConvCNP	-0.22$\pm$2e-2	1.05$\pm$1e-2	3.49$\pm$2e-2	4.25$\pm$1e-2
ANP	0.30$\pm$1e-2	1.14$\pm$2e-2	3.27$\pm$2e-2	4.19$\pm$1e-2
TNP	0.56$\pm$1e-2	1.75$\pm$1e-2	3.55$\pm$1e-2	4.57$\pm$1e-2
NMDP (Ours)	**0.88**$\pm$2e-2	**1.94**$\pm$2e-2	**3.63**$\pm$1e-2	**4.59**$\pm$1e-2

slightly degrade performance, consistent with the expected trade-off between model capacity and estimation difficulty.

Table 8. Target RMSE (↓) at 10% context on CIFAR-10 image completion tasks under various numbers of experts (10 runs).

L	5	7	9	11	13
NMDP (Ours)	0.0068$\pm$3e-4	0.0060$\pm$2e-4	**0.0058**$\pm$1e-4	**0.0058**$\pm$2e-4	0.0061$\pm$1e-4

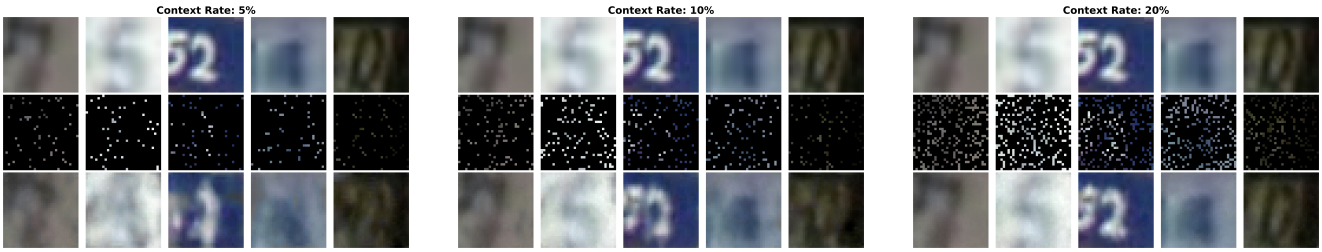
D.5. Computational complexity analysis

Our encoder adopts the same attention-based backbone as Attentive NPs, with computational cost $O(N^2 d)$. During training, the mixture decoder introduces an additional cost of $O(BL|\mathcal{D}^T|d^2)$ due to particle-based importance weighting. In practice, L is a small constant ($L \ll N$) and particles are computed in parallel on GPUs; the measured wall-clock overhead is approximately $2.1\times$ compared to standard NPs in our implementation. We also observe faster convergence in terms of optimization iterations: Fig. 6 shows that our importance-weighted objective (A5) reaches peak validation performance in fewer iterations than the ELBO-based stochastic gating variant (A4), partially offsetting the per-iteration overhead.

D.6. More Visualization Results

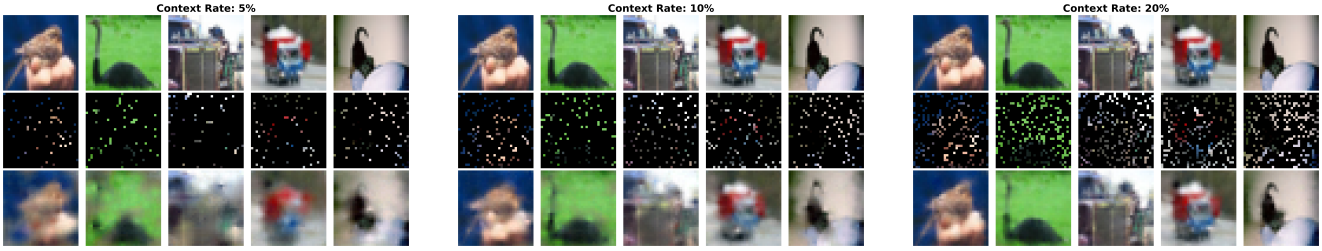
Here we include more visualized images, produced by our models, by varying the number of observed pixels. To demonstrate reconstruction performance at different sparsity levels, we show representative examples with $\{50, 100, 200\}$ context points, corresponding to approximately 5%, 10%, and 20% of the total 1024 pixels in 32×32 images.

As clearly evidenced by the visualization results across all four datasets, the reconstruction quality improves progressively as the number of context points increases. With only 50 context points (5% observation ratio), the model captures the coarse-grained structure and global patterns of the original images. When the context set expands to 100 points (10% observation ratio), the reconstructions exhibit significantly sharper edges and more accurate details. At the highest observation level of 200 context points (20% ratio), the reconstructed images achieve remarkable fidelity to the ground truth, with well-defined textures and precise boundaries.



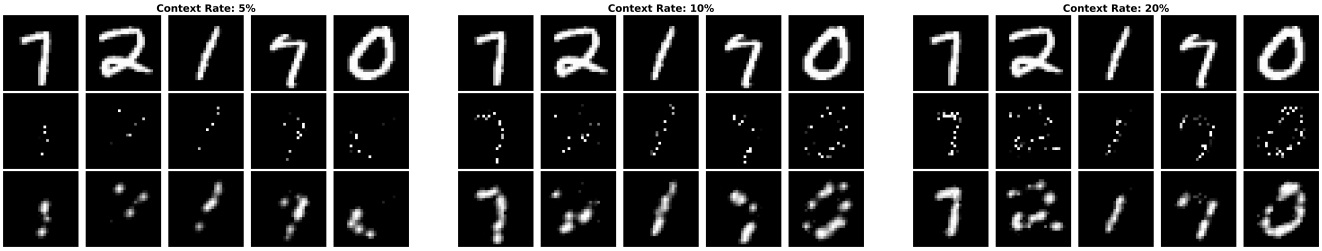
(a)5% pixels are context (b)10% pixels are context (c)20% pixels are context

Figure 9. Image completion results across different datasets. Each subplot from top to bottom shows the original image, context points and reconstructed output



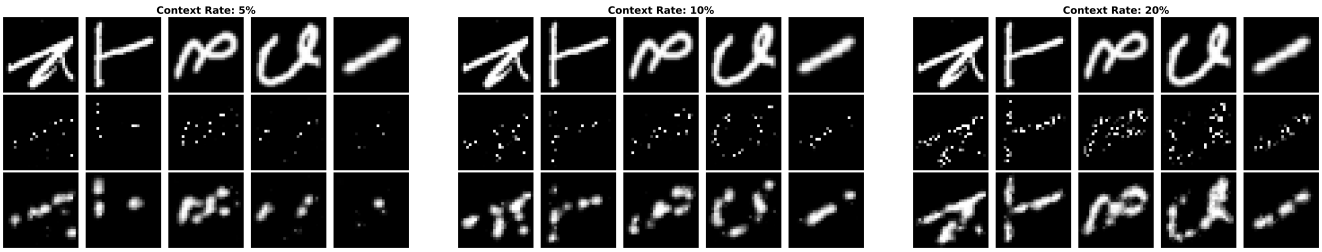
(a)5% pixels are context (b)10% pixels are context (c)20% pixels are context

Figure 10. Image completion results across different datasets. Each subplot from top to bottom shows the original image, context points and reconstructed output



(a)5% pixels are context (b)10% pixels are context (c)20% pixels are context

Figure 11. Image completion results across different datasets. Each subplot from top to bottom shows the original image, context points and reconstructed output



(a)5% pixels are context (b)10% pixels are context (c)20% pixels are context

Figure 12. Image completion results across different datasets. Each subplot from top to bottom shows the original image, context points and reconstructed output