JOINT-DOMAIN IMPLICIT CONDITIONAL DENSITY ESTIMATION UNDER SPARSE STRUCTURED COVERAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Conditional density estimation (CDE) aims to estimate the full conditional distribution $p(y \mid x)$. We study CDE under sparse coverage of ordered or gridded conditioning domains. Structure-aware CDE approaches typically encode cross-condition sharing through prespecified analytic relational functions, such as kernels or covariance functions. While effective when well matched to the data, the chosen functional family can restrict the form of admissible interactions. We introduce Joint-domain Implicit Conditional Density Estimation (JICDE), which represents $\{p(y \mid x) : x \in \mathcal{T}\}$ as the location-wise marginals of a stochastic generator defined jointly over the conditioning domain. JICDE retains the geometry and locality of the conditioning domain as architectural inductive biases, while parameterizing the detailed cross-condition sharing rule through shared learnable local transformations, allowing the induced sharing behavior to adapt across different structural regimes. Across ordered and gridded structured CDE benchmarks, JICDE consistently compares favorably with representative CDE and structured probabilistic baselines.

1 INTRODUCTION

Conditional density estimation (CDE) aims to characterize the distribution of a response variable conditioned on an input, $p(y \mid x)$ Sugiyama et al. (2010b). Compared with regression methods that provide only point predictions, estimating the full conditional distribution allows a model to capture more complex predictive behaviors, including skewness and multimodality Gao & Hastie (2022). Such distributional information is therefore important for downstream tasks such as risk assessment and decision-making, where the shape of the predictive distribution itself is of interest.

In this work, we focus on a CDE setting in which training observations sparsely cover a structured conditioning domain with explicit geometry. Specifically, we consider ordered or gridded domains, such as one-dimensional temporal coordinates and two-dimensional spatial grids. In these settings, x is not merely an unstructured feature vector: different conditioning locations are related through neighborhood relations and geometric structure induced by the underlying coordinate domain Dutordoir et al. (2018). When observations are sufficiently dense, generic CDE models can rely on the available data to learn the mapping from conditioning locations x to their corresponding conditional distributions. As observations become increasingly sparse, the direct statistical evidence available around individual conditioning values diminishes, making it more difficult to learn the conditional distributions directly from limited data.

A natural way to address sparse conditioning coverage is to exploit local structural coherence across neighboring conditioning values, whereby nearby conditioning locations exhibit related conditional behavior. Existing structure-aware CDE methods make this assumption operational by specifying a particular mathematical form for how conditioning values are related. Kernel-CDE Bashtannyk & Hyndman (2001), for example, determines how observations are shared according to a prescribed distance-based weighting rule, while Gaussian-process CDE Dutordoir et al. (2018) uses a covariance function to specify how information is coupled across conditioning locations. Those methods may be learned from data, but cross-condition sharing is still mediated through an explicitly parameterized relational object, such as a similarity kernel or covariance function. Consequently, the admissible pattern of information sharing is tied to the chosen relational parameterization. Such mechanisms can be highly effective when the imposed relational structure is well aligned with the data; however, when the actual cross-condition structure cannot be adequately represented within that family, the resulting structural misspecification can degrade conditional density estimation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

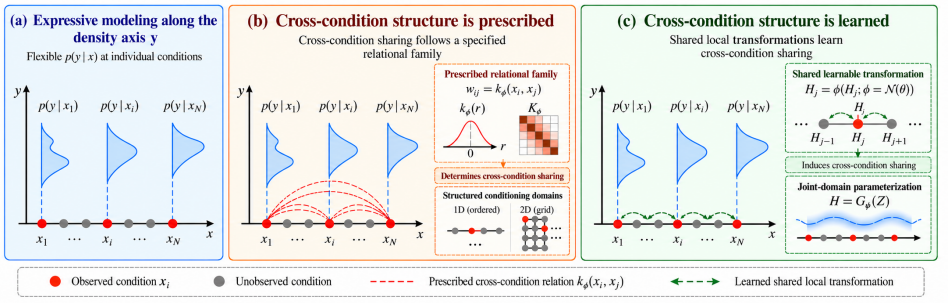

Figure 1: Illustration of the key challenge in sparse structured CDE. While existing methods often parameterize cross-condition sharing through an explicit analytic relational object, JICDE retains the conditioning-domain geometry as a structural scaffold and parameterizes the detailed sharing behavior through shared learnable local transformations, allowing the induced interaction pattern to adapt across different structural regimes.

This motivates the central question: *Can cross-condition sharing retain the geometric inductive bias of a structured conditioning domain while allowing its detailed functional form to adapt to different structural regimes, rather than being restricted to a prespecified analytic relational family?*

To answer this question, we propose Joint-domain Implicit Conditional Density Estimation (JICDE), an implicit CDE framework for sparse ordered and gridded conditioning domains. The key idea is to change how the conditional-density family is represented: rather than parameterizing $p(y \mid x)$ separately at each conditioning value, JICDE jointly parameterizes $\{p(y \mid x) : x \in \mathcal{T}\}$ with a stochastic generator defined over the conditioning domain. A single realization produces one response sample per conditioning location, while repeated realizations recover the model-implied conditional distribution at each x. This representation makes the structured conditioning domain an explicit axis along which cross-condition sharing can be parameterized. JICDE combines stochastic base noise with positional information and uses shared deconvolutional operations to propagate representations across neighboring conditions, providing a hierarchical local interaction scaffold over the conditioning domain. Within this scaffold, the detailed form of cross-condition sharing is represented end-to-end by learnable nonlinear transformations, allowing the induced sharing behavior to adapt across different structural regimes rather than being restricted to a single prespecified analytic relational family. The main contributions are summarized as follows:

- We introduce a joint-domain implicit parameterization for sparse structured CDE, representing the family of conditional distributions $\{p(y \mid x) : x \in \mathcal{T}\}$ as the location-wise marginals of a stochastic generator defined over the conditioning domain. Building on this representation, we propose JICDE which uses shared deconvolutional transformations to parameterize the cross-condition sharing rule within a structured local scaffold, allowing the induced sharing behavior to adapt across structural regimes.
- We evaluate JICDE on sparse structured CDE problems, measuring both conditional-distribution quality and point-prediction accuracy, and further demonstrate the utility of the resulting predictive distributions in downstream black-box optimization.

2 Joint-Domain Implicit Conditional Density Estimation

2.1 Structured Conditional Density Estimation under Sparse Observations

Let $\mathcal{T} = \{x_i\}_{i=1}^N \subset \mathbb{R}^{d_x}$ denote a finite conditioning domain equipped with an ordering or a grid-induced neighborhood structure. We observe responses only at a sparse subset of conditioning locations, $\mathcal{I}_{\text{tr}} \subset [N]$, $N_{\text{tr}} := |\mathcal{I}_{\text{tr}}| \ll N$, yielding the training set $\mathcal{D} = \{(x_i, y_i) : i \in \mathcal{I}_{\text{tr}}\}$, $y_i \in \mathbb{R}^{d_y}$. The statistical target is the family of conditional distributions

$$\{P^\star(\cdot \mid x_i) : i \in [N]\},$$

or their conditional densities $p^\star(y \mid x_i)$ when such densities exist. Under sparse sampling, however, local statistical evidence becomes limited, making each $P^\star(\cdot \mid x_i)$ difficult to estimate in isolation.

In this work, we exploit an additional structure that is available when the conditioning variable itself lies on an ordered or gridded domain. Rather than viewing the collection $\{P^\star(\cdot \mid x_i)\}_{i=1}^N$ as a set of unrelated conditional distributions, we regard it as a structured family indexed by $\mathcal{T}$. We do not assume that neighboring conditioning locations have identical conditional distributions. Instead, we assume only local structural coherence over the conditioning domain: neighboring conditioning locations exhibit related conditional behavior, so information from observed locations can inform estimation at nearby sparsely observed or unobserved locations. This motivates a joint-domain representation in which a stochastic generator produces a joint conditional sample whose component at x_i follows the model-implied conditional distribution $F_\theta(\cdot \mid x^i)$,

$$\widehat{Y}_\theta(\mathcal{T}; V) = \left[\widehat{Y}_\theta(x^1; V), \ldots, \widehat{Y}_\theta(x^N; V)\right], \qquad V \sim \zeta.$$

For each condition x^i, the model-implied predictive distribution is given by the marginal distribution

$$F_\theta(\cdot|x^i) = \text{Law}\left(\widehat{Y}_\theta(x^i; V)\right).$$

Thus, conditional density estimation is recovered through the location-wise marginal distribution of a generator defined jointly over the full conditioning domain. This joint-domain parameterization allows statistical information to be shared directly along the conditioning axis, making cross-condition coupling an explicit property of the model architecture. Let H_i denote an intermediate representation associated with x_i. Instead of using an independent transformation at every conditioning location, we consider a shared structured transformation

$$H_i' = \Psi_\theta\left(\{H_j : j \in \mathcal{N}(i)\}\right),$$

where $\mathcal{N}(i)$ denotes a neighborhood of x_i on the conditioning domain, and the same transformation Ψ_θ is reused across locations. Under this formulation, the key modeling requirement is that Ψ_θ be able to represent the structural relationships among neighboring conditioning locations. If these relationships can be captured, H_i' can incorporate information from neighboring conditions rather than being determined independently from H_i, thereby enabling cross-condition information sharing. The form of this interaction is left to the learnable transformation Ψ_θ, so that $\{F_\theta(\cdot \mid x_i)\}_{i=1}^N$ is represented as a coupled conditional-distribution family rather than a collection of unrelated location-wise distributions. In the next section, JICDE instantiates this principle using a deconvolution-based generator whose structured feature axis is aligned with the conditioning domain.

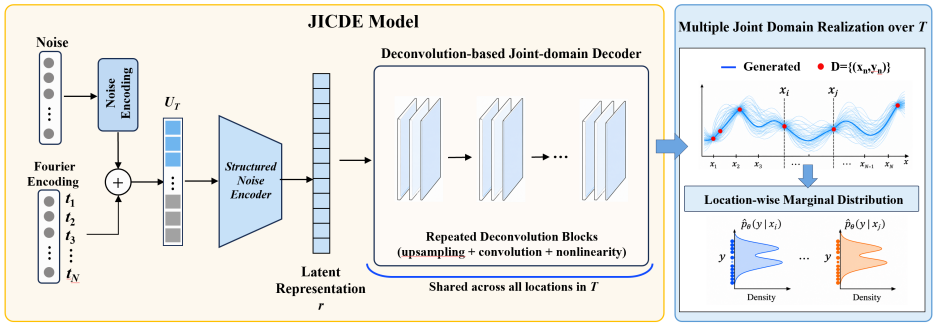

Figure 2: The model jointly produces one stochastic response at every conditioning location. The deconvolutional decoder couples neighboring condition-indexed representations through shared hierarchical transformations. Repeated realizations yield Monte Carlo samples from the location-wise conditional distributions $\hat{p}_\theta(y \mid x)$.

2.2 JICDE Model

Noise-structure input embedding JICDE separately constructs a stochastic noise representation and a positional representation of the structured conditioning domain. Specifically, for each generated sample, we draw a base-noise vector $Z \sim \mathcal{N}(0, I_{d_z}), Z \in \mathbb{R}^{d_z}$. The base-noise vector is first transformed by a learnable noise encoder h_{θ_z} to obtain

$$r_z = h_{\theta_z}(Z).$$

The base noise Z provides stochastic variation but contains no information about the geometry of the conditioning domain. Explicit positional information has been shown to facilitate coordinate-dependent generation by reducing the need for the network to infer spatial organization solely from

latent features Liu et al. (2018); Watters et al. (2019). We therefore augment the stochastic representation with a fixed structured random Fourier encoding $E = \phi(\mathcal{T}; \xi_0) \in \mathbb{R}^{d_e}$, where the entries of E are determined by the coordinates of $\mathcal{T}$ and their induced ordering or grid geometry through a random Fourier mapping. The resulting noise representation r_z is then concatenated with E:

$$U_{\mathcal{T}} = \mathrm{Concat}\left(r_z, E\right).$$

A subsequent fusion encoder h_{θ_h} combines the stochastic and structural information into a compact latent representation,

$$r = h_{\theta_h}(U_{\mathcal{T}}) \in \mathbb{R}^C.$$

At this stage, r is a single global latent representation and jointly contains stochastic information derived from the base noise and structural information derived from the conditioning domain. The deconvolutional decoder subsequently transforms this fused representation into structured feature maps and ultimately into a complete sample over $\mathcal{T}$.

Deconvolutional cross-condition coupling The fused latent representation r is transformed into a complete structured sample by the deconvolution-based decoder,

$$\widehat{X}(\mathcal{T}) = g_{\theta_g}(r).$$

The decoder g_{θ_g} is implemented as a stack of deconvolutional blocks. For an intermediate decoder feature map $H^{(\ell)}$, a generic block can be written as

$$H^{(\ell+1)} = \sigma_\ell \left(W_\ell * \mathrm{Up}_\ell \left(H^{(\ell)} \right) + b_\ell \right),$$

where $*$ denotes a convolution, Up_ℓ denotes the corresponding upsampling operation, and σ_ℓ denotes the nonlinear transformation used in the block.

JICDE exploits the built-in hierarchical spatial-correlation property of deconvolution to realize cross-condition information sharing. By aligning the structured axis of the deconvolutional decoder with the conditioning domain T, the local correlations induced by upsampling and convolution act directly across neighboring conditioning locations. Each generated response at a condition is therefore embedded in a structured representation whose neighboring entries are jointly transformed rather than modeled independently. Stacking such deconvolutional transformations organizes these local interactions hierarchically, allowing the generator to represent coherent variation of the conditional-distribution family across the conditioning domain. As a result, the fused representation is progressively transformed into a joint response realization over the conditioning domain. At each conditioning location x_i, the corresponding output component constitutes one sample from the model-implied conditional distribution $\hat{p}_\theta(y \mid x_i)$. Repeating this stochastic generation process therefore yields an empirical conditional distribution at every x_i.

Training JICDE is trained with a distributional objective on the sparse training set $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N_{tr}}$. Let $R_{\mathcal{D}} : \mathbb{R}^{N \times d_y} \to \mathbb{R}^{N_{tr} \times d_y}$ denote the selection operator that extracts the generator outputs at the conditioning locations $\{x_n\}_{n=1}^{N_{tr}}$. At each training iteration, we draw M independent realizations of the stochastic inputs, $Z^{(m)} \sim \mathcal{N}(0, I_{d_z}), \xi_0 \sim p_\xi, m = 1, \ldots, M$. Using the generator defined above, each stochastic realization jointly produces one conditional-response sample at every location in $\mathcal{T}$: $\widehat{X}^{(m)}(\mathcal{T})$. The predictive distributions at the training locations are represented by the empirical samples $\left\{ R_{\mathcal{D}} \widehat{X}^{(m)}(\mathcal{T}) \right\}_{m=1}^{M}$. Instead of minimizing a pointwise reconstruction loss, we optimize the continuous ranked probability score (CRPS), which is a proper scoring rule for distributional prediction. For scalar outputs, the sample-based CRPS at a training location x_n is estimated as

$$\mathrm{CRPS}_M\left(\widehat{F}_{\theta,x_i}, y_i\right) = \frac{1}{M} \sum_{m=1}^{M} \left| \widehat{X}^{(m)}(x_i) - y_i \right| - \frac{1}{2M^2} \sum_{m=1}^{M} \sum_{m'=1}^{M} \left| \widehat{X}^{(m)}(x_i) - \widehat{X}^{(m')}(x_i) \right|,$$

where $\widehat{F}_{\theta,x_n}$ denotes the empirical predictive distribution induced by the M generated samples at conditioning location x_n. For vector-valued outputs, we apply the same score componentwise and average over dimensions. The training objective is

$$\mathcal{L}_{\mathrm{CRPS}}(\theta) = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} \mathrm{CRPS}_M\left(\widehat{F}_{\theta,x_i}, y_i\right).$$

Minimizing L_{CRPS} trains the location-wise predictive distributions rather than only their conditional means. Although the score is evaluated only at observed locations, its gradients update decoder transformations that are reused throughout T. Together with the progressively expanding receptive fields of the deconvolutional stack, this provides a data-adaptive mechanism for cross-condition statistical sharing.

3 Related Work

Conditional Density Estimation Kernel-CDE assigns data-dependent weights according to proximity in the conditioning space, with bandwidth selection controlling the degree of smoothing Bashtannyk & Hyndman (2001); Hall et al. (2004). Related nonparametric approaches, such as least-squares conditional density estimation (LSCDE), directly estimate the density ratio underlying $p(y \mid x)$ Sugiyama et al. (2010b;a). These methods offer flexible density estimates with limited parametric assumptions, although their statistical and computational behavior can become increasingly sensitive to smoothing and dimensionality as the conditioning space grows. Neural CDE methods instead parameterize the conditional distribution through learned functions of x. Mixture Density Networks (MDNs) use a neural network to parameterize multimodal conditional densities through finite mixture model Bishop (1994). Conditional normalizing flows obtain more flexible densities through invertible transformations with tractable likelihoods and sampling Papamakarios et al. (2017); Trippe & Turner (2018); Durkan et al. (2019). Conditional latent-variable models, e.g., the conditional variational autoencoder, introduce stochastic latent variables and optimize a variational lower bound on the conditional likelihood Sohn et al. (2015). A closely related approach to JICDE is the Deconvolutional Density Network (DDN) Chen et al. (2022), which partitions the response domain into bins and uses deconvolution to impose hierarchical smoothness on the resulting density representation. Without an explicit mechanism for exploiting relationships among conditioning locations, neural CDE methods can struggle to estimate how $p(y \mid x)$ varies under sparse conditioning coverage. JICDE instead models the conditional-density family jointly over the conditioning domain, allowing dependencies among neighboring conditioning locations to be captured through shared transformations.

Cross-Condition Structure in CDE Cross-condition information sharing in CDE can be introduced explicitly through the geometry of the conditioning space. Kernel CDE provides the classical example, transferring information locally through similarity kernels and data-adaptive bandwidths. GP-CDE Dutordoir et al. (2018) offers a more flexible Bayesian alternative by combining latent variables with a Gaussian-process mapping, enabling non-Gaussian conditional densities while sharing structure across conditions through the GP covariance. Other approaches encode geometry through regularization, such as Wasserstein Laplacian Regularisation Graffeuille et al. (2022), which encourages nearby conditional distributions to vary smoothly over the input manifold, or through temporal structure, as in kernel-HMM De Gooijer et al. (2022) and FlexCodeTS Grivol et al. (2024). These mechanisms nevertheless encode particular assumptions about how information should be shared across conditions. When the imposed similarity, covariance, or geometric structure does not adequately reflect the actual cross-condition relationships, such structural misspecification may limit conditional density estimation Bashtannyk & Hyndman (2001); Wilson & Adams (2013); Beckers et al. (2018); Graffeuille et al. (2022). JICDE places structural bias at the architectural level, enabling the same learnable sharing mechanism to adapt across different structural regimes instead of following a fixed analytic relational family.

Structured Probabilistic Prediction Probabilistic prediction over structured domains has also been studied extensively beyond CDE. Gaussian processes MacKay et al. (1998) provide a classical approach by coupling observations and queries through covariance kernels, with extensions such as TDGP improving representation flexibility and RCGP improving robustness under model misspecification de Souza et al. (2023); Altamirano et al. (2024). Neural Processes offer a complementary amortized alternative: CNPs map context sets to predictive distributions, while ConvCNP/ConvNP incorporate translation-equivariant structure and richer predictive dependence, and recent variants such as DNP further introduce distance-aware local structure and uncertainty calibration Garnelo et al. (2018); Gordon et al. (2019); Venkataramanan & Denzler (2025); Foong et al. (2020). Although these methods are not CDE models in the same sense as the methods above, they address broader structured probabilistic prediction problems.

4 EXPERIMENT

We evaluate JICDE as a conditional density estimator on sparse ordered and gridded conditioning domains. Across all primary benchmarks, the data are represented as coordinate–response pairs $D = \{(x_n, y_n)\}$, and the prediction target is the conditional distribution of the response at held-out conditioning values. We consider one-dimensional temporal domains and two-dimensional spatial grids to evaluate whether the same cross-condition sharing mechanism can estimate $\hat{p}(y \mid x)$ under different structured geometries. We assess conditional-distribution quality together with pointwise summaries of the estimated distributions. Finally, we separately evaluate the downstream utility of JICDE's predictive distributions in black-box optimization.

We compare JICDE against two broad classes of baselines: representative conditional density estimation (CDE) methods and structured probabilistic prediction approaches. The CDE baselines include Kernel CDE Bashtannyk & Hyndman (2001), GP-CDE Dutordoir et al. (2018), and the DDN Chen et al. (2022). The probabilistic prediction baselines include standard Gaussian processes (GPs) Wilson et al. (2011), Thin and Deep Gaussian Processes (TDGP) de Souza et al. (2023), and ConvNP Foong et al. (2020). All methods are evaluated with the same sampling budget and protocol to ensure a fair comparison. More detailed descriptions of the experimental setup, algorithm configurations, and results are provided in the Appendix.

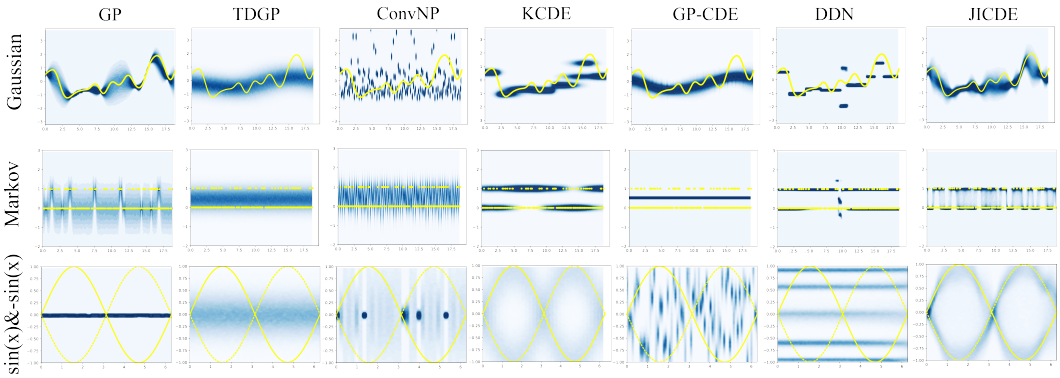

Figure 3: The results on synthetic data. The blue regions represent the estimating distribution of the methods, and the yellow lines indicate the ground truth.

4.1 SYNTHETIC STRUCTURED CDE UNDER SPARSITY AND STRUCTURAL MISSPECIFICATION

Can JICDE adapt across structurally different regimes under sparse supervision? To investigate this question, we construct three synthetic CDE benchmarks over an ordered conditioning domain. In each benchmark, supervision is available only at a sparse subset of conditioning values, and the model is required to estimate the conditional predictive distributions over the remaining conditioning domain. The three benchmarks are deliberately chosen to induce contrasting cross-condition structures: the Gaussian benchmark exhibits smooth stochastic dependence, the Markov benchmark contains sharper local transitions, and the double-sine benchmark introduces smooth multimodal structure. Together, these complementary regimes allow us to examine whether JICDE can adapt its sharing behavior across these regimes without relying on a single prespecified analytic relational family. We therefore visualize the predictive distributions over the full conditioning domain together with the held-out ground-truth responses, enabling direct comparison of conditional-density generalization and cross-condition sharing behavior. Figure 3 presents the qualitative results, with the corresponding quantitative evaluation reported in the Appendix.

The qualitative results reveal clear differences among the sharing mechanisms under sparse supervision. GP captures the smooth Gaussian benchmark well, but its smooth unimodal predictive structure is poorly matched to the abrupt Markov transitions and the bimodal double-sine target. ConvNP exhibits noisy and over-dispersed predictions, particularly in weakly supervised regions. KCDE performs comparatively well through local distance-weighted smoothing, although sharp changes can

Table 1: Performance comparison on the BIA, PDB, and ETT benchmarks. Lower is better.

	BIA ↓		PDB ↓		ETT ↓		Avg. Rank ↓
Method	NLL	MAE	NLL	MAE	NLL	MAE	
GP	2.77 ± 0.53	2.54 ± 1.07	2.48 ± 0.49	2.13 ± 0.58	4.30 ± 1.31	3.37 ± 0.63	5.00
TDGP	3.39 ± 0.07	6.99 ± 0.17	2.53 ± 0.52	3.33 ± 1.33	2.49 ± 0.32	2.15 ± 0.17	5.33
ConvNP	3.21 ± 0.58	4.08 ± 0.53	2.46 ± 0.32	3.18 ± 1.39	2.84 ± 0.50	3.73 ± 0.35	5.50
KCDE	2.27 ± 0.06	3.28 ± 0.26	1.84 ± 0.04	2.02 ± 0.08	2.35 ± 0.06	3.17 ± 0.16	3.50
GP-CDE	3.97 ± 0.55	2.83 ± 0.18	2.88 ± 0.57	1.81 ± 0.24	4.10 ± 0.82	$\mathbf{1.88 \pm 0.15}$	4.50
DDN	2.21 ± 0.29	2.48 ± 0.18	1.88 ± 0.15	1.89 ± 0.14	2.55 ± 0.24	2.76 ± 0.17	3.00
JICDE	$\mathbf{1.84 \pm 0.21}$	$\mathbf{1.81 \pm 0.36}$	$\mathbf{1.39 \pm 0.11}$	$\mathbf{1.16 \pm 0.15}$	$\mathbf{2.33 \pm 0.22}$	1.99 ± 0.13	**1.17**

be blurred. GP-CDE is more expressive than standard GP, but still fails to recover coherent structure on the Markov and double-sine benchmarks. DDN produces fragmented estimates along the conditioning axis because its deconvolution mainly regularizes the response-density representation rather than coupling neighboring conditions. In contrast, JICDE shows more consistent qualitative behavior across all three structural regimes, suggesting that its shared deconvolutional transformations can adapt cross-condition interactions from sparse observations.

4.2 CDE on Ordered Temporal Domains

We next evaluate sparse CDE on real one-dimensional ordered conditioning domains. Because only a sparse subset of conditioning locations is observed during training, direct statistical evidence for estimating $p(y \mid x)$ is unavailable at most query locations. The estimator must therefore transfer information across related temporal conditions while learning how the conditional distribution varies over the ordered domain. We evaluate this setting on BIA, PDB, and ETT. For each benchmark, 10% of the conditioning locations and their responses are used for training, and the remaining conditioning locations are held out for evaluating the estimated conditional distributions. Table 1 reports both NLL and MAE, assessing distributional quality and pointwise accuracy, respectively.

The baselines exhibit substantial variation across datasets and evaluation metrics. KCDE and DDN achieve relatively competitive NLL on BIA and PDB, but their pointwise errors remain consistently higher than those of JICDE. GP-CDE attains competitive MAE on PDB and ETT, while its NLL is noticeably weaker, indicating that accurate point predictions do not necessarily imply well-matched predictive distributions. TDGP performs well on ETT, but its performance degrades substantially on BIA and PDB. ConvNP is less competitive across both metrics in all three benchmarks. JICDE shows the most consistent performance across datasets and metrics. It achieves the lowest NLL and MAE on both BIA and PDB. On ETT, JICDE achieves the lowest NLL and the second-lowest MAE, behind GP-CDE. This consistency is also reflected in its average rank of 1.17. These results suggest that learning cross-condition interactions over the ordered conditioning domain provides an effective mechanism for estimating $p(y \mid x)$ under sparse conditioning coverage.

4.3 Grid-Structure CDE: Two-Dimensional

We next evaluate JICDE on two-dimensional gridded conditioning domains using image data. We formulate each image as a structured CDE problem: the conditioning variable $x = (u, v)$ is a pixel coordinate and the response y is its grayscale intensity or RGB value. A sparse subset of coordinate–response pairs is used as the CDE training set, and the model estimates the predictive conditional distribution $\hat{p}(y \mid x)$ at the remaining spatial locations. This yields a probabilistic image-completion task, in which uncertainty over each unobserved pixel is represented by its estimated conditional distribution rather than by a single reconstructed value. Figure 4 presents the visual results and Table 2 presents the quantitative results. Additional results are provided in Appendix B.3.

The image experiments reveal substantial differences in both reconstruction quality and predictive density estimation. GP produces fragmented or overly diffuse reconstructions, while GP-CDE becomes increasingly unstable as image complexity grows and fails markedly on ImageNet. TDGP improves over the standard GP on some cases, but its predictions remain visually blurred and often

Table 2: Quantitative evaluation on image completion. Lower is better.

Method	MNIST ↓		CIFAR ↓		ImageNet ↓		Avg. Rank ↓
	NLL	MAE	NLL	MAE	NLL	MAE	
GP	0.82 ± 0.005	0.39 ± 0.002	-0.12 ± 0.01	0.28 ± 0.001	-0.58 ± 0.01	0.15 ± 2e-4	5.67
TDGP	-1.82 ± 0.23	0.07 ± 0.01	-0.90 ± 0.20	0.12 ± 0.02	-1.12 ± 0.10	0.16 ± 0.02	3.83
ConvNP	1.47 ± 1.51	0.29 ± 0.09	0.52 ± 0.28	0.31 ± 0.04	0.78 ± 0.93	0.18 ± 0.12	6.50
KCDE	-1.23 ± 0.10	0.11 ± 0.003	-1.25 ± 0.05	0.06 ± 0.001	-1.75 ± 0.04	0.05 ± 3e-4	2.83
GP-CDE	0.08 ± 0.44	0.17 ± 0.03	-0.47 ± 0.32	0.08 ± 0.02	2.30 ± 0.02	0.27 ± 0.001	5.33
DDN	-2.36 ± 0.11	0.16 ± 0.01	-1.82 ± 0.03	0.09 ± 0.001	-1.98 ± 0.01	0.08 ± 0.001	2.83
JICDE	**-2.64 ± 0.35**	**0.05 ± 0.01**	**-1.84 ± 0.10**	**0.03 ± 0.001**	**-2.14 ± 0.04**	**0.04 ± 0.001**	**1.00**

exhibit pronounced color artifacts. ConvNP can produce locally sharp completions, particularly on the natural-image datasets, but these predictions are less consistent with the observed image content and are accompanied by substantially weaker NLL and MAE. KCDE provides relatively faithful reconstructions on CIFAR and ImageNet, although they remain noticeably smoothed. DDN achieves competitive NLL but tends to produce blurred or color-distorted outputs. JICDE shows the most consistent behavior across all three datasets, producing reconstructions that more closely preserve the underlying digit or object structure while retaining the observed color patterns. This qualitative advantage is consistent with the quantitative results: JICDE achieves the lowest NLL and MAE on MNIST, CIFAR, and ImageNet, with an average rank of 1.00. These results suggest that the same joint-domain CDE parameterization can effectively exploit spatial relationships among conditioning locations on two-dimensional grids under sparse observation.

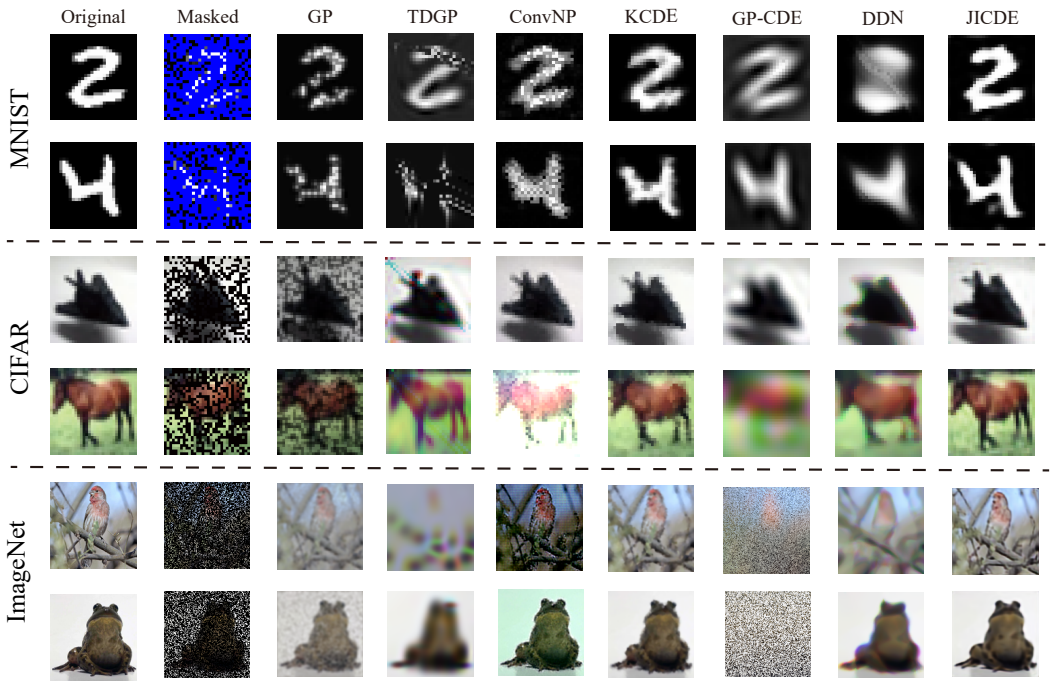

Figure 4: Image completion on the MNIST, CIFAR, and ImageNet. The "Masked" columns show the masked images, with unmasked pixels treated as known observations. For MNIST, the masked points are represented in blue, while for CIFAR and ImageNet, they are represented in black.

4.4 Downstream Utility: Black-Box Optimization

Having evaluated JICDE's conditional-distribution estimates directly on structured CDE benchmarks, we next ask whether these predictive distributions are useful as probabilistic surrogates for a downstream sequential decision problem. A natural downstream use of CDE is to serve as a probabilistic surrogate in Bayesian optimization, where acquisition functions balance predicted objective values against predictive uncertainty. We compare all methods under the same EI acquisition rule and evaluation budget on Schwefel and Rastrigin. Under this controlled setup, faster convergence and better attained objective values indicate that the surrogate provides more informative predictive estimates for identifying promising query locations.

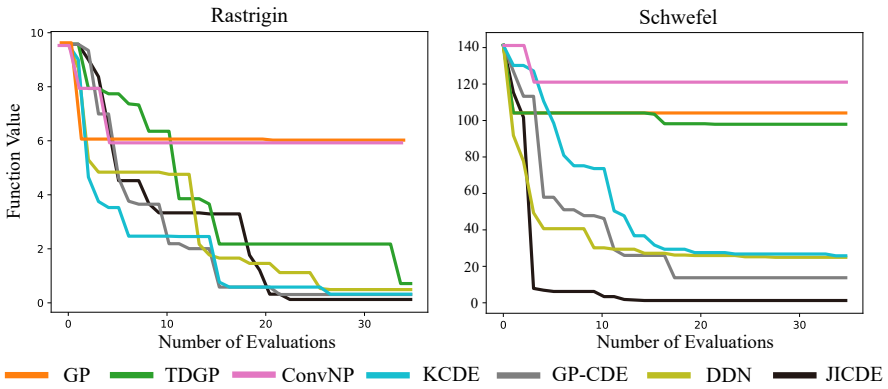

Figure 5: Optimization results (lower is better) on the Schwefel and Rastrigin problems, presented as averaged convergence curves, where JICDE outperforms the other methods.

Figure 5 shows that JICDE provides the strongest optimization performance on both objectives, with the advantage being particularly clear in sample efficiency. On Rastrigin, JICDE progressively reduces the best observed objective to near zero within roughly 20 evaluations and achieves the lowest final value. GP-CDE, KCDE, and DDN also improve substantially, but require more evaluations to reach comparably low objective values, while TDGP improves mainly at later stages and both GP and ConvNP plateau at considerably higher levels. The separation is even more pronounced on Schwefel: JICDE reaches approximately 10 within the first few evaluations and approaches 1 after about 15 evaluations, whereas GP-CDE improves more gradually to around 20, and the remaining methods terminate at noticeably higher objective values. Because all methods use the same acquisition rule and evaluation budget, these differences reflect the usefulness of the surrogate predictions supplied to the optimizer: more informative predictions lead EI toward promising regions earlier, whereas less effective surrogate estimates result in slower improvement or premature plateaus. The consistent advantage of JICDE across both landscapes provides an application-level indication that its predictive modeling is more effective for guiding sequential search under sparse evaluations.

5 Conclusion

We studied conditional density estimation under sparse coverage of ordered and gridded conditioning domains, where reliable estimation at weakly observed locations requires information sharing across structured conditions. We introduced JICDE, which represents the family of conditional distributions $\{p(y \mid x) : x \in \mathcal{T}\}$ as the location-wise marginals of a stochastic generator defined jointly over the conditioning domain. JICDE separates the structural scaffold from the detailed sharing behavior: the geometry and locality of the conditioning domain provide architectural inductive biases, while shared deconvolutional transformations parameterize a flexible nonlinear sharing operator whose induced behavior can adapt across different structural regimes. This avoids restricting cross-condition sharing to a single prespecified analytic kernel, covariance, or smoothing family. Across synthetic, temporal, and gridded CDE benchmarks, JICDE achieved consistently strong distributional and point-prediction performance, and its predictive distributions were further useful in downstream black-box optimization.

REFERENCES

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and conjugate gaussian process regression. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

David M Bashtannyk and Rob J Hyndman. Bandwidth selection for kernel conditional density estimation. *Computational Statistics & Data Analysis*, 36(3):279–298, 2001.

Thomas Beckers, Jonas Umlauft, and Sandra Hirche. Mean square prediction error of misspecified gaussian process models. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 1162–1167. IEEE, 2018.

Christopher M Bishop. Mixture density networks. 1994.

Bing Chen, Mazharul Islam, Jisuo Gao, and Lin Wang. Deconvolutional density network: Modeling free-form conditional distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6183–6192, 2022.

Jan G De Gooijer, Gustav Eje Henter, and Ao Yuan. Kernel-based hidden markov conditional densities. *Computational Statistics & Data Analysis*, 169:107431, 2022.

Daniel Augusto de Souza, Alexander Nikitin, ST John, Magnus Ross, Mauricio A Álvarez, Marc Deisenroth, João Paulo Gomes, Diego Mesquita, and César Lincoln Mattos. Thin and deep gaussian processes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13138–13148. Curran Associates, Inc., 2023.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Vincent Dutordoir, Hugh Salimbeni, James Hensman, and Marc Deisenroth. Gaussian process conditional density estimation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Steffen Finck, Nikolaus Hansen, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical report, Citeseer, 2010.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79, 2009.

Zijun Gao and Trevor Hastie. Lincde: Conditional density estimation via lindsey's method. *Journal of Machine Learning Research*, 23(52):1–55, 2022. URL `http://jmlr.org/papers/v23/21-0840.html`.

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 (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, 10–15 Jul 2018.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. *arXiv preprint arXiv:1910.13556*, 2019.

Olivier Graffeuille, Yun Sing Koh, Jörg Wicker, and Moritz K Lehmann. Semi-supervised conditional density estimation with wasserstein laplacian regularisation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 6746–6754, 2022.

Gustavo Grivol, Rafael Izbicki, Alex A Okuno, and Rafael B Stern. Flexible conditional density estimation for time series. *Brazilian Journal of Probability and Statistics*, 38(2):215–231, 2024.

Peter Hall, Jeff Racine, and Qi Li. Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99(468):1015–1026, 2004.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/60106888f8977b71e1f15db7bc9a88d1-Paper.pdf.

Minfang Lu, Shuai Ning, Shuangrong Liu, Fengyang Sun, Bo Zhang, Bo Yang, and Lin Wang. Opt-gan: A broad-spectrum global optimizer for black-box problems by learning distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 12462–12472, 2023.

David JC MacKay et al. Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166, 1998.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, 93(3):583–594, 2010a.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pp. 781–788, 2010b.

Brian L Trippe and Richard E Turner. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*, 2018.

Aishwarya Venkataramanan and Joachim Denzler. Distance-informed neural processes. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 12456–12485. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/124dde499d62b58e97e42a45b26d7369-Paper-Conference.pdf.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pp. 1067–1075. PMLR, 2013.

Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. Gaussian process regression networks. *arXiv preprint arXiv:1110.4411*, 2011.

APPENDIX

A ALGORITHM CONFIGURATIONS

Gaussian Process (GP). We use an exact Gaussian process regression model implemented with GPyTorch. The GP employs a constant mean function and an RBF covariance kernel wrapped by a scale kernel,

$$k(x, x') = \sigma_f^2 k_{\mathrm{RBF}}(x, x'),$$

where both the output scale and kernel hyperparameters are learned from the training data. Model parameters are optimized using Adam with a learning rate of 10^{-4} by maximizing the exact log marginal likelihood.

Kernel-Conditional Density Estimation (KCDE). Kernel-CDE estimates the conditional density using kernel smoothing in both the conditioning and response spaces. Gaussian kernels are used for both x and y, and the corresponding bandwidths are determined from the observed data using Silverman's rule of thumb. For a query condition x, the conditioning-space kernel assigns normalized weights to the observed samples according to their distances from x. These weights are then combined with Gaussian kernel smoothing over the observed responses to evaluate the conditional density on a response grid. The estimator is therefore nonparametric and does not require gradient-based optimization or variational inference.

Thin and Deep Gaussian Process (TDGP). We use the official implementation of Thin and Deep Gaussian Processes (TDGP) de Souza et al. (2023) (`https://github.com/spectraldani/thindeepgps`). TDGP constructs a hierarchical Gaussian-process model through input-dependent locally linear transformations of the conditioning space. Posterior inference is performed using a variational inducing-point approximation, with inducing variables introduced for both the final output GP and the latent transformation processes. We use the implementation and associated optimization procedure provided by the authors' reference code.

Deconvolutional Density Network (DDN). We use the public implementation of the Deconvolutional Density Network (DDN) Chen et al. (2022) (`https://github.com/NBICLAB/DDN`). Following the configuration described in the original work, the conditioning network contains two hidden layers with 64 tanh units per layer and batch normalization. For each response dimension, the density estimator consists of two Upsample–Conv–BatchNorm–LeakyReLU blocks followed by an Upsample–Conv output block. The response domain is discretized into 256 bins, and the variational layer uses 16 latent codes. For one-dimensional response tasks, the masking mechanism is disabled and the model is conditioned directly on the input.

Convolutional Neural Process (ConvNP). We implement ConvNP Foong et al. (2020) using the `Neural Processes` framework (`https://github.com/wesselb/neuralprocesses`). Observed context points are first mapped onto a regular internal discretization using set convolutions, after which the resulting functional representation is processed by a convolutional U-Net. We use an internal discretization density of 64 points per unit with a margin of 0.1. The U-Net contains six convolutional levels with 64 channels per level, kernel size 5, stride 2, and ReLU activations. The length scales of the encoder and decoder set convolutions are initialized to the inverse of the internal discretization density and are learned during training.

Gaussian Process Conditional Density Estimation (GP-CDE). We implement GP-CDE using a latent-variable sparse Gaussian process model. The conditioning variable is first transformed by a learnable linear map A, while a variational encoder infers the latent variable w from the observed pair (x, y). The encoder consists of three fully connected hidden layers with 128 units and ReLU activations, followed by separate linear heads that parameterize the mean and log-variance of the variational posterior over w. The sparse GP takes the concatenated representation $[Ax, w]$ as input. Its mean function is constant, and its covariance is parameterized by an RBF kernel with automatic relevance determination (ARD), allowing the transformed conditioning variable and latent variable to have separate length scales. The inducing-point locations are initialized using k-means clustering over the training inputs, and the variational distribution over the inducing variables is represented using a natural variational distribution. A second learnable linear map P transforms the GP output to the response space. Training is performed by maximizing an evidence lower bound (ELBO). The objective contains the expected data log likelihood together with the KL divergence between the

variational posterior of w and its standard normal prior, as well as the KL divergence associated with the GP inducing variables. The observation-noise variance is constrained to be positive using a Softplus transformation, with an additional lower bound of 10^{-4} for numerical stability.

JICDE configurations. We use a small set of architecture configurations shared across task families, rather than selecting a separate architecture for each dataset. The overall generator design is kept unchanged across experiments: a latent random vector is mapped to a low-resolution structured representation, combined with Fourier positional features, and progressively upsampled through convolutional decoder blocks. Architecture dimensions are varied only when required by the dimensionality or resolution of the conditioning domain. The Deconvolutional block is constructed by Upsample–Conv–BatchNorm–LeakyReLU. For the synthetic and double-sine experiments, we use a latent dimension of 20 and map the latent variable to an initial feature representation of shape 16×25. The one-dimensional Fourier positional encoding uses two frequencies with a maximum frequency of 2, resulting in five positional channels when the original coordinate is included. After concatenation with the 16 generated feature channels, the decoder uses channel dimensions $21 \to 4 \to 2 \to 1$. Three linear $2\times$ upsampling stages increase the structured resolution from 25 to 200; each stage is followed by a one-dimensional convolution with kernel size 5 and a LeakyReLU activation.

For the Finance and ETT experiments, we use an identical temporal configuration. The latent dimension is set to 10, and the latent vector is mapped to an initial representation of shape 8×45. Fourier positional encoding uses four frequencies with a maximum frequency of 5, yielding nine positional channels including the original coordinate. Concatenating these positional features with the eight generated feature channels gives 17 input channels to the decoder, whose channel dimensions are $17 \to 16 \to 8 \to 1$. Three linear $2\times$ upsampling stages map the structured resolution from 45 to 360, with a one-dimensional convolution of kernel size 5 applied at each stage. The same configuration is used for both datasets because they share the same one-dimensional output resolution and therefore admit the same decoder geometry.

For the image experiments, we use the same two-dimensional JICDE design. A 50-dimensional latent variable is first mapped to a low-resolution feature map with four feature channels. Two-dimensional Fourier positional features are projected to four channels by a 1×1 convolution before being concatenated with the generated feature map. The decoder then applies successive $2\times$ bilinear upsampling operations followed by 5×5 convolutions, BatchNorm, and LeakyReLU activations. A final convolution maps the decoder representation to the required number of image channels, followed by a sigmoid activation. The spatial size of the initial feature map and the number of upsampling stages are determined by the target image resolution. Specifically, MNIST starts from a 7×7 representation and reaches 28×28 after two upsampling stages, whereas CIFAR-10 and ImageNet start from 8×8 representations and use the number of $2\times$ upsampling stages required to reach 32×32 and 128×128, respectively. The output dimensionality is similarly fixed by the data modality (one channel for MNIST and three channels for RGB images). Thus, the image experiments share a common architectural template, with resolution-dependent changes arising from the geometry of the target domain rather than from a separate architecture choice for each dataset. We use the Adam optimizer with a learning rate of 1×10^{-3}. All code (and training/evaluation scripts) will be made publicly available upon acceptance.

B EXPERIMENT

We adopt a fixed-design view of structured CDE. The conditioning domain $T = \{x_i\}_{i=1}^N$ is fixed, while each condition x_i indexes a response random variable Y_i with marginal distribution $P_i^\star = P^\star(\cdot \mid x_i)$. Only one realization need be observed at a particular condition; as in fixed-design probabilistic regression and nonparametric CDE, estimation is made possible by sharing statistical strength across conditions through a structured model class rather than by estimating each $P_i^\star$ independently. In our setting this structural sharing follows the ordering or grid geometry of T. The estimand is the family of location-wise marginals $\{P_i^\star\}$; we do not claim to identify the full cross-location joint distribution from a single sparse realization.

For image task, we use the two-dimensional pixel field as a structured probabilistic-imputation benchmark under the same finite-design formulation. Reconstruction metrics evaluate point summaries, whereas proper distributional scores evaluate the model-implied predictive distributions. We

do not interpret generator-induced dispersion as a unique decomposition of aleatoric and epistemic uncertainty.

B.1 Quantitative Indicators

Since JICDE and several competing methods represent their predictive distributions through samples, we apply the same histogram-based density estimator to all methods for a consistent comparison. For each test case n and evaluation location i, we draw $S = 10{,}000$ predictive samples $\{\hat{y}_{i,s}^{(n)}\}_{s=1}^{S}$. We construct a piecewise-constant density using an equal-width histogram with $B = 64$ bins. The histogram range is determined adaptively from the predictive samples at each location,

$$\ell_i^{(n)} = \min_s \hat{y}_{i,s}^{(n)} - 10^{-6}, \qquad u_i^{(n)} = \max_s \hat{y}_{i,s}^{(n)} + 10^{-6},$$

with bin width

$$h_i^{(n)} = \frac{u_i^{(n)} - \ell_i^{(n)}}{B}.$$

If $C_{i,b(y)}^{(n)}$ denotes the number of predictive samples falling in the bin containing y, the corresponding histogram density estimate is

$$\hat{p}_i^{(n)}(y) = \frac{C_{i,b(y)}^{(n)}}{S\, h_i^{(n)}}.$$

The reported NLL is the average negative log-density assigned to the held-out ground-truth responses:

$$\mathrm{NLL} = -\frac{1}{N_{\mathrm{test}}|\mathcal{I}|} \sum_{n=1}^{N_{\mathrm{test}}} \sum_{i\in\mathcal{I}} \log\Big(\hat{p}_i^{(n)}(y_i^{*(n)}) + \varepsilon\Big),$$

where $\varepsilon = 10^{-8}$ is used for numerical stability.

The Mean Absolute Error (MAE) is computed at held-out evaluation locations by comparing the predicted values against the ground truth and averaging across trajectories. For the synthetic and time-series benchmarks, each trial randomly assigns 10% of the data for training and the remaining for evaluation; all experiments are repeated 10 times with independent splits. For all quantitative results, the synthetic, time-series, and black-box optimization tasks are repeated 10 times, whereas the image experiments are repeated 5 times due to their higher computational cost; we report the mean and standard deviation.

B.2 Quantitative results on synthetic benchmarks.

Table 3 complements the qualitative results by comparing distributional and pointwise predictive accuracy across the three synthetic benchmarks. On the Gaussian benchmark, GP achieves the lowest NLL (0.61 ± 0.73) and MAE (0.42 ± 0.06), while JICDE attains a nearly identical MAE (0.43 ± 0.04) with a slightly higher NLL (0.99 ± 0.56). This behavior is consistent with the Gaussian benchmark being well aligned with the covariance structure assumed by GP, providing a setting in which a prescribed relational model matches the underlying dependence well. On the Markov benchmark, however, the performance of GP degrades substantially, consistent with a mismatch between its smooth covariance-based inductive bias and the sharper cross-condition transitions. In contrast, JICDE remains competitive on the Gaussian benchmark while achieving the best performance on the Markov benchmark. Specifically, JICDE obtains the lowest NLL (-0.03 ± 0.74) and MAE (0.47 ± 0.03). This contrast suggests that JICDE can adapt the form of cross-condition interaction through learned shared transformations, rather than relying on a prespecified relational family whose effectiveness depends on how well it matches the underlying dependency structure.

Table 3: Quantitative evaluation on Synthetic Experiment (Lower is better)

Method	Gaussian↓		Markov↓		Double Sine↓		Avg. Rank ↓
	NLL	MAE	NLL	MAE	NLL	MAE	
GP	**0.61 ± 0.73**	**0.42 ± 0.06**	1.81 ± 1.17	0.60 ± 0.03	17.57 ± 1e-4	–	3.80
TDGP	1.22 ± 0.51	0.67 ± 0.04	0.71 ± 0.05	0.54 ± 0.01	4.30 ± 0.30	–	4.20
ConvNP	9.05 ± 2.75	0.92 ± 0.18	5.73 ± 3.93	0.61 ± 0.08	3.13 ± 0.56	–	6.00
KCDE	1.99 ± 1.16	0.60 ± 0.06	0.54 ± 0.90	0.57 ± 0.02	0.48 ± 0.007	–	3.60
GP-CDE	5.71 ± 6.42	0.59 ± 0.06	18.42 ± 2e-6	0.50 ± 7e-6	10.88 ± 1.55	–	4.80
DDN	10.91 ± 1.26	0.45 ± 0.09	3.13 ± 3.35	0.53 ± 0.04	1.41 ± 0.46	–	4.20
JICDE	0.99 ± 0.56	0.43 ± 0.04	**-0.03 ± 0.74**	**0.47 ± 0.03**	**0.40 ± 0.34**	–	**1.40**

Table 4: Quantitative evaluation on image completion using PSNR and SSIM (Hccigher is better)

Method	MNIST↑		CIFAR↑		ImageNet↑		Avg. Rank ↓
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
GP	6.33 ± 0.10	0.01 ± 0.00	10.57 ± 0.03	0.05 ± 0.00	10.97 ± 0.17	0.08 ± 0.00	6.67
TDGP	9.54 ± 0.64	0.41 ± 0.04	14.41 ± 0.46	0.42 ± 0.02	13.24 ± 0.37	0.16 ± 0.01	3.83
ConvNP	6.22 ± 2.08	0.03 ± 0.09	11.16 ± 0.41	0.17 ± 0.03	11.64 ± 0.87	0.19 ± 0.05	5.67
KCDE	12.47 ± 0.21	0.52 ± 0.03	19.33 ± 0.18	0.76 ± 0.01	19.77 ± 0.06	0.58 ± 0.00	2.00
GP-CDE	8.41 ± 0.51	0.28 ± 0.08	15.43 ± 1.36	0.60 ± 0.10	11.59 ± 0.28	0.06 ± 0.01	4.67
DDN	8.55 ± 0.12	0.15 ± 0.02	13.30 ± 0.10	0.37 ± 0.02	14.04 ± 0.03	0.42 ± 0.00	4.17
JICDE	**18.82 ± 1.90**	**0.87 ± 0.06**	**27.49 ± 0.73**	**0.95 ± 0.01**	**24.97 ± 0.21**	**0.78 ± 0.01**	**1.00**

DDN also obtains a competitive NLL (3.13 ± 3.35), whereas GP-CDE exhibits substantially larger variability across runs. These results are consistent with the qualitative observations that the sharper local transitions in the Markov setting are less well matched by a single prescribed smooth relational structure.

For the Double-Sine benchmark, we report NLL but omit MAE. The conditional distribution is multimodal, and a point-summary metric such as MAE does not measure whether the individual modes or their probability mass are recovered: in particular, a central prediction between two modes may attain a reasonable pointwise error despite assigning little probability to either mode. NLL is therefore more informative for this benchmark. JICDE achieves the lowest NLL (0.40 ± 0.34), closely followed by KCDE (0.48 ± 0.007), whereas GP degrades sharply to 17.57, indicating a severe mismatch between its predictive distribution and the multimodal target. Overall, JICDE obtains the best average rank (1.40) despite not being the best method on the Gaussian benchmark. More importantly, its performance remains consistently strong as the cross-condition structure changes from smooth Gaussian dependence to sharper Markov transitions and multimodal Double-Sine behavior. Together with the qualitative results, this suggests that learning the cross-condition sharing mechanism provides greater adaptability across heterogeneous structured regimes than relying on a single prespecified relational form.

B.3 Probabilistic Image completion

For an image, $x_i = (u_i, v_i)$ denotes a pixel coordinate and Y_i the corresponding grayscale or RGB response in a structured response field. The image experiment is therefore a two-dimensional fixed-design instantiation of the same location-indexed conditional-distribution problem. Only a sparse subset of coordinate-response realizations is observed during fitting. The estimand remains the family of location-wise conditional distributions; JICDE does not claim to identify the full joint random-field distribution across all pixels. We use 100 observed pixels for MNIST but increase this to 500 for CIFAR and 4800 for ImageNet, since the images are more complex and require more observations. Additional image results are provided in Fig. 8 (ImageNet), Fig. 7 (CIFAR) and Fig. 6 (MNIST). For all three datasets, images are randomly sampled from each class (i.e., no cherry picking).

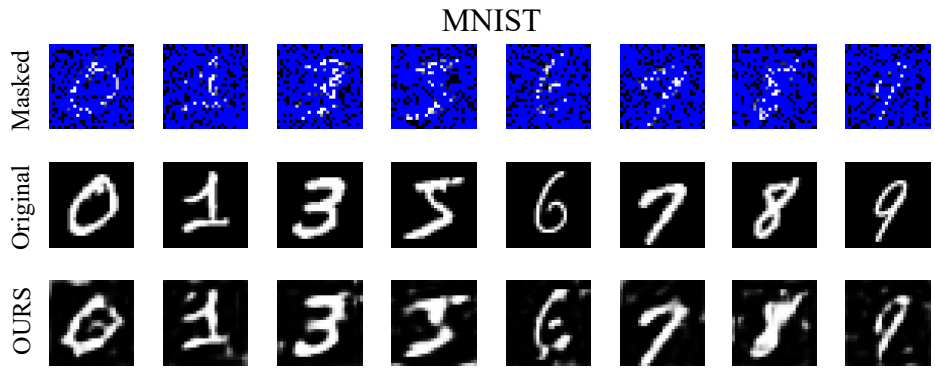

Figure 6: Image completion on MNIST dataset. The "Masked" columns show the masked images, with unmasked pixels treated as known observations. The masked pixels are represented in blue.

B.4 Ordered Temporal Domains

We consider two financial tasks and the ETTh1 electricity-transformer benchmark. Each benchmark is treated as a one-dimensional CDE dataset consisting of temporal conditioning values and scalar responses. The financial series contain daily closing prices of Shanghai Pudong Development Bank (PDB; 600000.SH) and Guangzhou Baiyun International Airport (BIA; 600004.SH). Each original financial series contains 746 daily observations, from which we extract a contiguous segment of 360 observations for the experiments. For each observation, the conditioning variable x represents the elapsed time from the beginning of the selected segment, and the response y is the corresponding daily closing price.

For the electricity benchmark, we use ETTh1.csv from the ETT-small dataset. ETTh1 is recorded at hourly resolution and contains six power-load variables together with transformer oil temperature (OT) as the prediction target. In our experiments, we retain only the OT series and extract a contiguous segment of 360 hourly observations. The conditioning variable x denotes the elapsed hours from the beginning of the selected segment, and y is the corresponding oil temperature. The dataset was obtained from the publicly available ETTh1 mirror on Kaggle.

For BIA, PDB, and ETT, x_i denotes a temporal coordinate and Y_i denotes the response indexed by that coordinate. The response variables across time need not be independent. The statistical target is not a forecasting distribution of the form

$$P(Y_{t+1} \mid Y_{\leq t}),$$

but the location-indexed conditional family

$$P^\star(\cdot \mid x_i).$$

Hence these experiments evaluate sparse CDE over an ordered conditioning domain rather than forecasting or extrapolation.

B.5 Black-Box Optimization

Black-box Optimization (BBO) Lu et al. (2023) is a task that aims to find the global optimum for a given black-box problem. Unlike traditional optimization methods, which rely on an explicit mathematical model or gradient information of the objective problem, BBO treats the objective problem as a "black box". This means that the internal workings of the problem are not directly accessible or observable. Instead, we can only infer its behavior by providing inputs and receiving corresponding outputs. The black-box nature of the problem makes it difficult, or even impossible to directly model the problem's internal mechanics. Mathematically, BBO problems are typically

formulated as follows:

$$\arg\min f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^n,$$

where $\boldsymbol{x}$ represents the decision variables, i.e., solutions, and Ω is the search domain in n-dimensional space, with n indicating the number of variables involved in the optimization. The "black-box" nature of the problem makes solving such problems particularly challenging.

To address black-box optimization problems, surrogate-based optimization methods Forrester & Keane (2009) construct a predictive model of the unknown objective function f and use it to guide subsequent evaluations. Bayesian optimization (BO) Shahriari et al. (2015) is a prominent example, traditionally using a Gaussian process MacKay et al. (1998) as the surrogate to provide both predictions of the objective value and associated predictive uncertainty. In this setting, the surrogate is fitted from only a small number of objective evaluations and is then used to determine where the objective should be queried next. A typical BO procedure consists of three components: the surrogate model approximates the objective landscape from the available evaluations; an acquisition function combines the surrogate predictions and predictive uncertainty to score candidate locations; and the most promising candidate is evaluated and added to the training set. The surrogate is then updated, and this procedure is repeated until the evaluation budget is exhausted.

Accurate surrogate modeling is therefore directly relevant to optimization performance. A surrogate that captures the objective landscape well can provide more informative acquisition values and guide evaluations toward promising regions, whereas inaccurate predictions may distort the acquisition function, misdirect subsequent queries, and waste the limited evaluation budget. Consequently, under the same acquisition strategy and evaluation budget, strong BBO performance provides application-level evidence that the underlying predictive model supplies useful information for sequential search. It therefore serves not only as a downstream evaluation of predictive modeling quality, but also as a practical demonstration of the model's ability to act as an effective surrogate in decision-making applications.

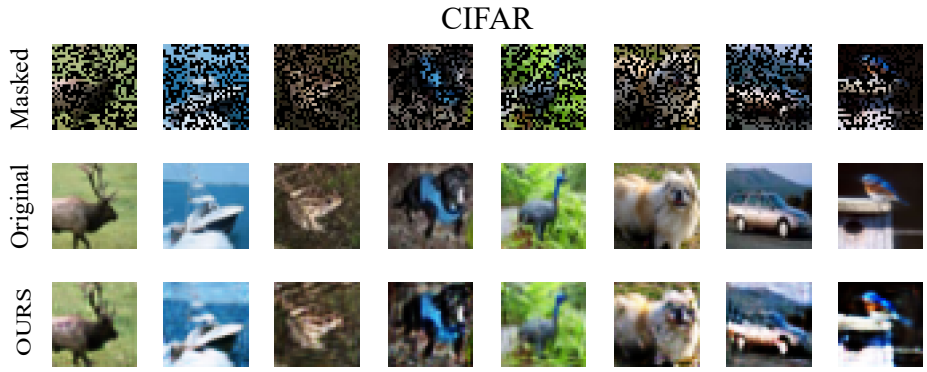

Figure 7: Image completion on CIFAR dataset. The "Masked" columns show the masked images, with unmasked pixels treated as known observations. The masked pixels are represented in black.

In this experiment, we use Gaussian Process-based Bayesian Optimization (GP-BO) as the baseline and replace the Gaussian process with TDGP, KCDE, DDN, ConvNP, GP-CDE, and JICDE to evaluate the performance of these algorithms from a black-box optimization perspective. The search domain is discretized into 200 candidate points. The initial points are drawn uniformly at random from these 200 candidates. At each iteration, we use Expected Improvement (EI) as the acquisition function and optimize EI by exhaustively evaluating it over all 200 candidates, selecting the maximizer. The evaluation budget is fixed at 35 objective evaluations. We validate the results on two widely used multimodal black-box optimization problems, the Schwefel and Rastrigin problems Finck et al. (2010).

The mathematical expressions for these problems are as follows:

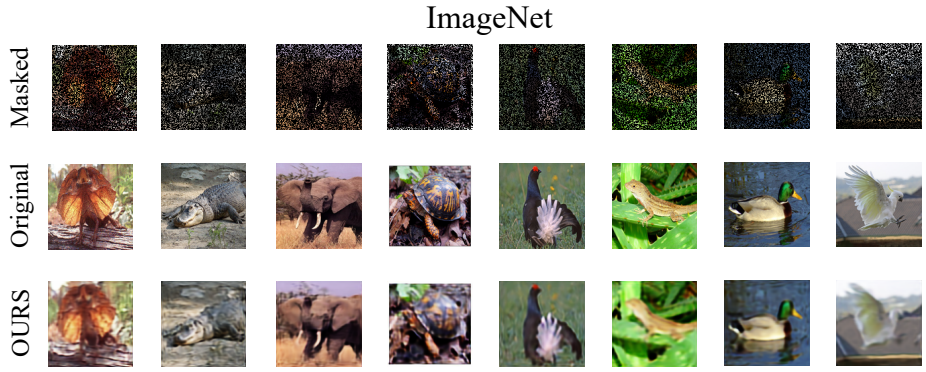

Figure 8: Image completion on ImageNet dataset. The "Masked" columns show the masked images, with unmasked pixels treated as known observations. The masked pixels are represented in black.

1. Schwefel problem:

$$f(\boldsymbol{x}) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

2. Rastrigin problem:

$$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$$

In the actual optimization process, we follow standard practices in the BBO field by intentionally ignoring any explicit expression information, treating the problem as a black-box, and optimizing without utilizing any inherent knowledge of the problem. For each experiment, we begin by randomly initializing three solutions, and in each iteration, we select the best candidate solution by optimizing EI.

B.6 Ablation Experiment

We compare JICDE with approximately parameter-matched variants that isolate different mechanisms for sharing information across conditions. The MLP removes explicit cross-condition coupling, while the Transformer replaces the local convolutional bias with globally learned interactions. To isolate whether JICDE benefits merely from local information sharing or specifically from learning the form of that sharing, we construct a FIXED_LOCAL variant that preserves the same local receptive-field structure while removing learnable spatial mixing. In FIXED_LOCAL, each learned $k = 5$ convolution in the decoder is replaced by a learned 1×1 channel projection, a fixed depthwise Gaussian convolution ($k = 5, \sigma = 1$), and a second learned 1×1 projection. The intermediate width is chosen to approximately match the number of trainable parameters of the original convolution. The two 1×1 projections retain learnable channel transformations, whereas the fixed depthwise Gaussian operator determines how information is propagated across neighboring conditioning locations. Thus, the comparison between JICDE and FIXED_LOCAL keeps the local inductive bias and model capacity approximately unchanged, while varying whether the local spatial interaction is learned from data or prescribed in advance.

FIXED_LOCAL achieves the best performance on the Gaussian benchmark, with JICDE ranking second, consistent with a prescribed smooth Gaussian operator being well matched to this setting. In contrast, JICDE performs best on the Markov benchmark and on both real-world datasets, PDB and ETT, yielding the best overall average rank of 1.25. This pattern suggests that a prescribed local operator can be highly effective when its assumed structure matches the data, whereas learning the local interaction provides greater adaptability across heterogeneous cross-condition structures. The MLP performs consistently worse, highlighting the importance of explicit information sharing under sparse coverage. The Transformer is competitive on PDB and ETT but degrades substantially on the

Table 5: Quantitative evaluation of Ablation Experiment (Lower is better)

Method	Gaussian ↓	Markov ↓	PDB ↓	ETT ↓	Avg. Rank ↓
OURS	0.99 ± 0.56	$\mathbf{-0.03 \pm 0.74}$	$\mathbf{1.39 \pm 0.11}$	$\mathbf{2.33 \pm 0.22}$	**1.25**
MLP	8.04 ± 0.77	8.04 ± 1.19	4.75 ± 0.90	12.60 ± 0.60	3.75
FIXED_LOCAL	$\mathbf{0.64 \pm 0.27}$	1.43 ± 1.39	1.56 ± 0.06	4.69 ± 0.64	2.00
TRANSFORMER	1.44 ± 0.71	16.52 ± 5.99	1.49 ± 0.13	6.06 ± 1.54	3.00

synthetic benchmarks, particularly Markov, indicating that globally learned interactions can be effective in some settings but are less consistent across structural regimes. Overall, the results support JICDE's combination of a local inductive bias with data-adaptive cross-condition interaction.

C Relation to the deconvolutional inductive bias in DDN

DDN motivates deconvolution by its built-in hierarchical spatial correlation: upsampling expands the structured feature map, while convolution with shared weights introduces spatial correlation, thereby encouraging coherent and smooth density representations. In DDN, this structure is imposed along the discretized response-density axis. JICDE reuses the same architectural principle on a different statistical axis. By aligning the decoder's structured dimension with the conditioning domain T, the hierarchical correlations induced by deconvolution act across neighboring conditioning locations. The decoder therefore provides a learnable structured prior over how the family $\{F_\theta(\cdot \mid x) : x \in T\}$ varies across the domain. The convolutional weights are optimized jointly from the sparse conditional observations, so the resulting cross-condition coupling is data-adaptive rather than specified by a predetermined numerical similarity or covariance function.

D Computational Resources and Cost

The experiments are conducted on several machines, including machine A: Intel(R) Core(TM) i7-14650HX CPU, NVIDIA GeForce RTX 4060 Laptop GPU, running on Windows 11 operating system, and machine B: Dual Intel(R) Xeon(R) CPU E5-2620v3 2.40GHz CPU, NVIDIA Tesla A100 GPU, running on Linux operating system. We benchmark the wall-clock training time of all methods on machine B for probabilistic image completion on MNIST, CIFAR, and ImageNet, using each method's original architectural settings. Table 6 reports the results. GP and TDGP require increasingly longer training times as the spatial resolution grows, increasing from less than one minute on MNIST to 7 and 8 minutes on CIFAR, and further to 22 and 24 minutes on ImageNet, respectively. ConvNP remains relatively efficient, requiring 2, 2, and 7 minutes across the three datasets, while GP-CDE requires 2, 5, and 15 minutes. DDN is fast on MNIST but scales less favorably to larger image domains, increasing from less than one minute on MNIST to 16 minutes on CIFAR and 100 minutes on ImageNet. In contrast, both KCDE and JICDE complete training within one minute on all three datasets. These results indicate that JICDE maintains low empirical training cost as the structured conditioning domain increases in spatial resolution, while retaining the predictive performance reported in the main experiments.

E Limitations and future directions

JICDE is currently designed for finite, regularly structured conditioning domains, such as ordered one-dimensional sequences and two-dimensional grids. The deconvolutional construction assumes that the conditioning locations can be organized on such a regular lattice, and therefore does not directly accommodate irregularly spaced observations, arbitrary query locations, or more general relational structures such as graphs. Extending the joint-domain formulation to irregular domains, for example through geometry-aware or graph-based structured operators, is an important direction for future work. The current formulation is also primarily transductive with respect to the conditioning domain: the model generates predictions over a predefined structured set of conditioning locations and is not intended for extrapolation beyond that domain or for direct transfer to previously unseen domain geometries or resolutions. Developing coordinate-continuous or resolution-

Table 6: Computational time of different algorithms. The unit is minutes. "< 1" indicates less than one minute, and "-" indicates that the data were not recorded.

	MNIST	**CIFAR**	**IMAGENET**
GP	< 1	7	22
TDGP	< 1	8	24
CONVNP	2	2	7
KCDE	< 1	< 1	< 1
GP-CDE	2	5	15
DDN	< 1	16	100
JICDE	< 1	< 1	< 1

independent variants could enable prediction at arbitrary conditioning values and improve transfer across domains. In addition, the effectiveness of JICDE relies on the presence of exploitable local structure across neighboring conditioning locations. When conditional behavior varies independently across nearby locations, or when the chosen neighborhood structure is poorly aligned with the underlying dependency structure, the local architectural bias may provide limited benefit and can potentially be restrictive. Learning the neighborhood geometry itself, or combining local and non-local interaction mechanisms, may broaden the range of structures that can be represented. Finally, although JICDE uses a joint stochastic generator over the conditioning domain, the present training objective targets conditional distributions through location-wise marginal scoring. Consequently, the cross-location dependence induced among generated samples should be viewed primarily as an architectural mechanism for information sharing, rather than as an identified model of the true joint stochastic process across conditioning locations. Explicitly learning and evaluating calibrated joint dependencies constitutes a separate extension of the current framework.

F THE USAGE OF LARGE LANGUAGE MODELS

We used Large Language Model (LLM) tools primarily for manuscript polishing, with additional use for translation, experimental-data organization, experimental analysis, suggesting titles and keywords, and reviewing the manuscript to identify potential issues and improve overall quality. All AI-assisted outputs were reviewed and verified by the authors. We did not use generative AI for synthetic-data generation, theoretical or conceptual development, mathematical claims or proofs, hypothesis formulation, experimental or methodological design, or implementation of the proposed methods. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Matias Altamirano, François-Xavier Briol, and Jeremias Knoblauch. Robust and conjugate gaussian process regression. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

David M Bashtannyk and Rob J Hyndman. Bandwidth selection for kernel conditional density estimation. *Computational Statistics & Data Analysis*, 36(3):279–298, 2001.

Thomas Beckers, Jonas Umlauft, and Sandra Hirche. Mean square prediction error of misspecified gaussian process models. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 1162–1167. IEEE, 2018.

Christopher M Bishop. Mixture density networks. 1994.

Bing Chen, Mazharul Islam, Jisuo Gao, and Lin Wang. Deconvolutional density network: Modeling free-form conditional distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6183–6192, 2022.

Jan G De Gooijer, Gustav Eje Henter, and Ao Yuan. Kernel-based hidden markov conditional densities. *Computational Statistics & Data Analysis*, 169:107431, 2022.

Daniel Augusto de Souza, Alexander Nikitin, ST John, Magnus Ross, Mauricio A Álvarez, Marc Deisenroth, João Paulo Gomes, Diego Mesquita, and César Lincoln Mattos. Thin and deep gaussian processes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13138–13148. Curran Associates, Inc., 2023.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Vincent Dutordoir, Hugh Salimbeni, James Hensman, and Marc Deisenroth. Gaussian process conditional density estimation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Steffen Finck, Nikolaus Hansen, Raymond Ros, and Anne Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical report, Citeseer, 2010.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79, 2009.

Zijun Gao and Trevor Hastie. Lincde: Conditional density estimation via lindsey's method. *Journal of Machine Learning Research*, 23(52):1–55, 2022. URL http://jmlr.org/papers/v23/21-0840.html.

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 (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1704–1713. PMLR, 10–15 Jul 2018.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. *arXiv preprint arXiv:1910.13556*, 2019.

Olivier Graffeuille, Yun Sing Koh, Jörg Wicker, and Moritz K Lehmann. Semi-supervised conditional density estimation with wasserstein laplacian regularisation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 6746–6754, 2022.

Gustavo Grivol, Rafael Izbicki, Alex A Okuno, and Rafael B Stern. Flexible conditional density estimation for time series. *Brazilian Journal of Probability and Statistics*, 38(2):215–231, 2024.

Peter Hall, Jeff Racine, and Qi Li. Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 99(468):1015–1026, 2004.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/60106888f8977b71e1f15db7bc9a88d1-Paper.pdf.

Minfang Lu, Shuai Ning, Shuangrong Liu, Fengyang Sun, Bo Zhang, Bo Yang, and Lin Wang. Opt-gan: A broad-spectrum global optimizer for black-box problems by learning distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 12462–12472, 2023.

David JC MacKay et al. Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166, 1998.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, 93(3):583–594, 2010a.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pp. 781–788, 2010b.

Brian L Trippe and Richard E Turner. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*, 2018.

Aishwarya Venkataramanan and Joachim Denzler. Distance-informed neural processes. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 12456–12485. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/124dde499d62b58e97e42a45b26d7369-Paper-Conference.pdf.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International conference on machine learning*, pp. 1067–1075. PMLR, 2013.

Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. Gaussian process regression networks. *arXiv preprint arXiv:1110.4411*, 2011.