STRUCTURE-AWARE EPISTEMIC UNCERTAINTY QUANTIFICATION FOR NEURAL OPERATOR PDE SURROGATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators (NOs) provide fast, resolution-invariant surrogates for mapping input fields to PDE solution fields, but their predictions can exhibit significant epistemic uncertainty due to finite data, imperfect optimization, and distribution shift. For practical deployment in scientific computing, uncertainty quantification (UQ) must be both computationally efficient and spatially faithful, i.e., uncertainty bands should align with the localized residual structures that matter for downstream risk management. We propose a structure-aware epistemic UQ scheme that exploits the modular anatomy common to modern NOs (lifting–propagation–recovering). Instead of applying unstructured weight perturbations (e.g., naive dropout) across the entire network, we restrict Monte Carlo sampling to a module-aligned subspace by injecting stochasticity only into the lifting module, and treat the learned solver dynamics (propagation and recovery) as deterministic. We instantiate this principle with two lightweight lifting-level perturbations, including channel-wise multiplicative feature dropout and a Gaussian feature perturbation with matched variance, followed by standard calibration to construct uncertainty bands. Experiments on challenging PDE benchmarks (including discontinuous-coefficient Darcy flow and geometry-shifted 3D car CFD surrogates) demonstrate that the proposed structure-aware design yields more reliable coverage, tighter bands, and improved residual–uncertainty alignment compared with common baselines, while remaining practical in runtime.

Keywords: Neural Operator, Uncertainty Quantification, Partial Differential Equations

1 INTRODUCTION

Uncertainty quantification (UQ) in supervised learning aims to characterize the distribution of model outputs conditioned on given inputs. In surrogate modeling of partial differential equations (PDEs), UQ can be as important as point prediction: well-calibrated uncertainty bands can indicate where and by how much a surrogate may deviate from a high-fidelity numerical solver, providing a practical notion of risk and enabling more reliable deployment in accelerated pipelines such as aerospace CFD design Ramezankhani et al. (2024); Feng et al. (2026), electromagnetic system simulation Peng et al. (2022); Hao et al. (2023), and safety-critical monitoring in the nuclear industry Gopakumar et al. (2023).

As fast, resolution-invariant surrogates that learn mappings between function spaces from data, neural operators (NOs) Lu et al. (2019); Li et al. (2021); Wu et al. (2024) are widely adopted because they can be deployed efficiently and reused across discretizations. However, with finite training data, limited optimization budgets, and potential distribution shift at inference time, NO predictions are inherently uncertain. In many PDE settings, the underlying simulator is effectively deterministic (i.e., aleatoric noise is negligible), so the dominant uncertainty in practical NO deployment is epistemic: uncertainty about the learned operator induced by limited data and imperfect training.

1.1 EPISTEMIC UNCERTAINTY IN NOS

An NO $\mathcal{G}[\cdot;\theta] : \mathbf{A} \mapsto \hat{\mathbf{U}}$ approximates a target operator $\mathcal{G}$ that maps discretized input fields $\mathbf{A} \in \mathbb{R}^{N\times d_a}$ to discretized solution fields $\mathbf{U} \in \mathbb{R}^{N\times d_u}$, using a finite dataset $\mathcal{D} = \{(\mathbf{A}_m, \mathbf{U}_m)\}_{i=1}^{M}$. In a Bayesian operator-learning view, the weights w are treated as random after observing $\mathcal{D}$, inducing a posterior $p(w \mid \mathcal{D})$ and a predictive distribution $p(\hat{U} \mid \mathbf{A}, \mathcal{D})$. Posterior predictive statistics derived from this distribution are then used to construct uncertainty bands Gal & Ghahramani (2016); Lakshminarayanan et al. (2017); Weber et al. (2024).

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.

When the underlying PDE solver is effectively deterministic, epistemic uncertainty is primarily manifested as the variability of $\mathcal{G}[\mathbf{A}; \theta]$ under plausible weights $\theta \sim p(\theta \mid \mathcal{D})$. In practice, this uncertainty is shaped by several factors, including (i) optimization heuristics that bias training toward a particular local optimum, (ii) finite compute budgets and regularization choices that restrict the set of reachable solutions, and (iii) out-of-distribution (OOD) inputs that probe under-constrained regions of the learned operator.

1.2 Previous Quantification Designs

Estimating the high-dimensional posterior $p(\theta \mid \mathcal{D})$ is intractable for NOs Wilson & Izmailov (2020), so existing approaches Gal & Ghahramani (2016); Lakshminarayanan et al. (2017); Weber et al. (2024); Ma et al. (2024) rely on practical sampling and approximation schemes to estimate posterior predictive quantities, yielding predictive statistical fields such as variances, quantiles, and credible intervals. Recent approaches include:

Deep Ensembles. Deep Ensembles Lakshminarayanan et al. (2017) train multiple models from different random initializations and statistically aggregate their predictions, treating the resulting variability as samples from an implicit distribution over plausible solution weights. They are often robust and empirically strong, but incur substantial training cost and still do not exploit which parts of the operator are responsible for uncertainty.

Laplace Approximation. Laplace methods Kristiadi et al. (2021); Magnani et al. (2025) approximate the posterior $p(\theta \mid \mathcal{D})$ locally around a trained solution $\hat{\theta}$ with a Gaussian whose covariance is given by (an approximation to) the inverse Hessian of the negative log posterior evaluated at $\hat{\theta}$. For overparameterized neural operators, full-curvature Laplace is typically computationally prohibitive and can be numerically fragile Ritter et al. (2018). A common simplification is last-layer Laplace Kristiadi et al. (2020), which is cheaper and more stable, but primarily captures uncertainty in the final linear readout (i.e., how to combine learned features) while neglecting uncertainty in the learned representations. However, since NO residuals can vary substantially across the output domain, epistemic error often stems from imperfect residual representations rather than merely mis-weighted features. This feature-level mismatch induces a calibration tension: achieving higher residual coverage often requires overly inflated (wide) intervals, whereas sharper intervals tend to suffer from under-coverage.

MCDropout. MCDropout Gal & Ghahramani (2016) injects randomness by sampling dropout masks at inference time and uses Monte Carlo forward passes to estimate posterior predictive moments. It is simple, scalable, and easy to integrate into existing neural operators. However, the induced perturbations in weight space are dictated by the dropout mechanism and therefore do not distinguish between parameter dimensions, effectively applying a uniform, unstructured stochastic perturbation across the network. When inserted naively, dropout can degrade predictive accuracy and lead to overly conservative, wide uncertainty bands: random masking may suppress channels or neurons that are critical for signal propagation in neural operators, producing samples in which essential intermediate features are partially removed. This can bias the Monte Carlo predictive mean and inflate predictive variance due to increased variability across sampled predictions Jiménez et al. (2025).

Table 1: Comparison of common epistemic UQ methods in neural operators. M is the ensemble size, T the number of stochastic passes.

Method	**Injection Site**	**Train / infer**
Deep Ensembles Lakshminarayanan et al. (2017)	all weights	M / M
Laplace Approximation Magnani et al. (2025)	readout $\theta_{\mathcal{Q}}$	1 / 1
MC Dropout Gal & Ghahramani (2016)	all layers	1 / T
Input perturbation Weber et al. (2024)	input field $a(x)$	1 / T
Ours	lifted field $\mathbf{V}_0$	1 / T

We summarize the computational cost of these epistemic UQ methods in Table 1 for deploying a neural operator (NO) with high-coverage uncertainty bands. Since NO inference is typically far cheaper than training, the overhead of tens to hundreds of stochastic forward passes for MCDropout is often modest relative to the additional training required by Deep Ensembles. For Laplace approximation, the dominant overhead usually arises from forming and solving with an (approximate) Hessian at the trained solution, which can be both computationally expensive for large NOs when applied beyond highly structured approximations.

1.3 Our Structure-aware Design

While previous structure-agnostic posterior approximations of $p(\theta \mid \mathcal{D})$ often spend most of their sampling budget exploring parameter dimensions that contribute little or even harm the alignment between uncertainty bands and residual structures, we propose a structure-aware UQ scheme for neural operators. Concretely, we perform Monte Carlo integration by sampling only a module-aligned subset of parameters, $\theta_{\mathcal{S}} \sim p(\theta_{\mathcal{S}} \mid \mathcal{D})$, where $w_{\mathcal{S}}$ corresponds to a designated operator module $\mathcal{S}$. This design is motivated by a key property of highly structured neural operators: parameters within the same module typically play similar functional roles and affect the final prediction through shared mechanisms. Ignoring such module semantics can make posterior sampling both inefficient and misleading. In particular, unstructured perturbations of sensitive parameters may produce samples with biased predictive means and large residuals, leading to overly conservative, inflated uncertainty bands; meanwhile, perturbing parameters whose induced features are (locally/globally) orthogonal to the residual field wastes sampling effort and provides little benefit for aligning uncertainty bands with residuals.

Contributions. Leveraging the distinct roles of the three core components in neural operators, Lifting, Propagation, and Recovering, we propose a UQ scheme that prioritizes structure-aligned subspaces, effectively attributing epistemic uncertainty throughout the end-to-end mapping (from input conditions to solution fields) to uncertainty in the generated lifted feature tensors. Empirically, this structure-aware design estimates residual structures between predicted and ground-truth fields with higher fidelity than existing UQ baselines. Our main contributions in this paper are summarized as follows:

- **Structure-aware epistemic UQ via lifting-subspace sampling.** We propose a module-aligned Monte Carlo UQ strategy for neural operators that leverages the lifting–propagation–recovering decomposition. Instead of injecting stochasticity into the full parameter space, we restrict sampling to the lifting subspace and keep the learned solver dynamics (propagation and recovering) deterministic. This design is efficient and structure-preserving: it models epistemic uncertainty as uncertainty over the lifted feature field that is then propagated by a fixed remainder operator, offering an operational interpretation akin to "feature-space initial-condition" uncertainty.
- **Two plug-and-play sampling mechanisms with matched statistics.** We instantiate this principle with two lightweight perturbations on the lifted features: (i) channel-wise multiplicative (dropout-style) feature noise, and (ii) Gaussian feature perturbation with variance matched to the inverted-dropout scale. Both are zero-mean around the deterministic embedding, require no retraining, and add only inference-time overhead via T stochastic forward passes.
- **Empirical evidence on challenging PDE benchmarks and OOD generalization.** We evaluate on discontinuous 2D Darcy flow and geometry-shifted 3D ShapeNet Car CFD surrogates, spanning representative NO backbones (e.g., spectral-propagation and attention-based variants). Across settings, the proposed method improves the coverage–bandwidth trade-off and remains competitive in runtime relative to widely used epistemic UQ baselines, including MC Dropout, Laplace-style approximations, deep ensembles, and input perturbation. Comprehensive comparisons and dropout-focused ablations further indicate that our structure-aware sampling yields stable uncertainty maps that better track residual fields attributable to epistemic uncertainty.

2 Problem Setup

2.1 Operator Learning

A PDE solution operator $\mathcal{G}$ maps an input function (e.g., initial conditions, forcing terms, or boundary conditions) $a(x) \in \mathcal{A}$ to a solution function $u(x) \in \mathcal{U}$, where $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$ are (suitable subsets of) Banach spaces over a spatial domain D. Here we consider vector-valued fields $a : D \to \mathbb{R}^{d_a}$ and $u : D \to \mathbb{R}^{d_u}$ with $x \in D$.

Neural operators parameterize $\mathcal{G}$ with weights θ, yielding a learned operator $\mathcal{G}_\theta$. In practice, we work with discretized fields on a grid $\{x_n\}_{n=1}^N \subset D$. The discretized input is a tensor $\mathbf{A} \in \mathbb{R}^{N \times d_a}$ with rows $\mathbf{A}_{n,:} = a(x_n)$, and the discretized solution is $\mathbf{U} \in \mathbb{R}^{N \times d_u}$ with rows $\mathbf{U}_{n,:} = u(x_n)$.

A typical neural operator can be decomposed into three components,

$$\mathcal{G}_\theta \;=\; \mathcal{Q}_{\theta_{\mathcal{Q}}} \circ \mathcal{M}_{\theta_{\mathcal{M}}} \circ \mathcal{P}_{\theta_{\mathcal{P}}}, \tag{1}$$

where: (i) the lifting operator $\mathcal{P}_{\theta_{\mathcal{P}}} : \mathbf{A} \mapsto \mathbf{V}_0 \in \mathbb{R}^{N \times d_v}$ embeds the input into a latent feature field; (ii) the propagation operator $\mathcal{M}_{\theta_{\mathcal{M}}} : \mathbf{V}_0 \mapsto \mathbf{V}_L$ iteratively transforms latent features via structured linear operators and nonlinear

activations, typically containing the majority of parameters and determining much of the approximation capacity Kovachki et al. (2021); and (iii) the recovering operator $\mathcal{Q}_{\theta_{\mathcal{Q}}} : \mathbf{V}_L \in \mathbb{R}^{N\times d_v} \mapsto \hat{\mathbf{U}}$ maps final latent features back to the output space, commonly implemented as a linear layer.

When explicit PDE priors are unavailable, neural operators are often trained in a purely data-driven manner by minimizing losses such as mean squared error (MSE) or relative L_2 error (defined in Appendix A). Throughout this paper, we focus on neural operators that follow the above architectural composition and are trained under this purely data-driven paradigm; operator-learning models outside this framework (e.g., DeepONet-style architectures Lu et al. (2019)) are not considered.

2.2 Epistemic Uncertainty Quantification

From a Bayesian perspective, epistemic uncertainty in neural operators is reflected in how well we can characterize the posterior over weights and translate it into calibrated predictive statistics. Given an input a, let $\hat{u}(x)$ denote the predicted solution and define the residual field

$$r(x) := (\hat{u} - u)(x). \tag{2}$$

We consider a nonnegative uncertainty-band field $\mathrm{band}(x) \geq 0$ that defines the pointwise prediction interval

$$\hat{u}(x) \pm \mathrm{band}(x). \tag{3}$$

For vector-valued outputs, we apply the band channel-wise and compute coverage per channel; a location x_n is counted as covered only if all channels are covered.

Prior works Weber et al. (2024); Ma et al. (2024); Magnani et al. (2025) evaluate UQ methods using (i) coverage rate (C.R.) and (ii) average bandwidth (Avg. B.W.). For a test set $\{(a_m, u_m)\}_{m=1}^{M_{\text{test}}}$ and grid points $\{x_n\}_{j=1}^{N}$, considering one channel $d_u = 1$, define the indicator $\mathbb{I}[\cdot] \in \{0, 1\}$. The case-wise coverage rate for a test case m is

$$\text{C.R.}_m := \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[\, \|\hat{u}_m(x_n) - u_m(x_n)\|_2 \leq \mathrm{band}_m(x_n)\,], \tag{4}$$

and the average case-wise coverage over the test set is

$$\text{Avg. C.R.} := \frac{1}{M_{\text{test}}}\sum_{m=1}^{M_{\text{test}}} \text{C.R.}_m. \tag{5}$$

The overall coverage aggregated over the entire test set and spatial grid is

$$\text{Total C.R.} := \frac{1}{M_{\text{test}}N}\sum_{m=1}^{M_{\text{test}}}\sum_{n=1}^{N} \mathbb{I}[\, \|\hat{u}_m(x_n) - u_m(x_n)\|_2 \leq \mathrm{band}_m(x_n)\,]. \tag{6}$$

Note that for under uniform grids, N is the same and thus Avg. C.R. and Total C.R. coincide.

(ii) The average (half) bandwidth is

$$\text{Avg. B.W.} := \frac{1}{M_{\text{test}}N}\sum_{m=1}^{M_{\text{test}}}\sum_{n=1}^{N} \mathrm{band}_m(x_n). \tag{7}$$

In the idealized limit, a band that exactly matches the pointwise residual magnitude attains 100% coverage with the smallest possible width. Accordingly, better UQ methods should achieve higher coverage with tighter bands. However, in practice, even methods with high coverage and relatively low bandwidth can still fail to capture the spatial structure of the residual field (Section 1.2). Such misalignment yields unnecessarily conservative uncertainty in regions where the bands greatly exceed the true residuals, increasing downstream correction and design costs by triggering avoidable interventions in otherwise reliable regions.

3 Method

Epistemic uncertainty is typically modeled via stochastic inference over model parameters. A widely used tool is Monte Carlo (MC) integration, whose convergence rate scales as $\mathcal{O}(T^{-1/2})$ and is largely insensitive to the ambient

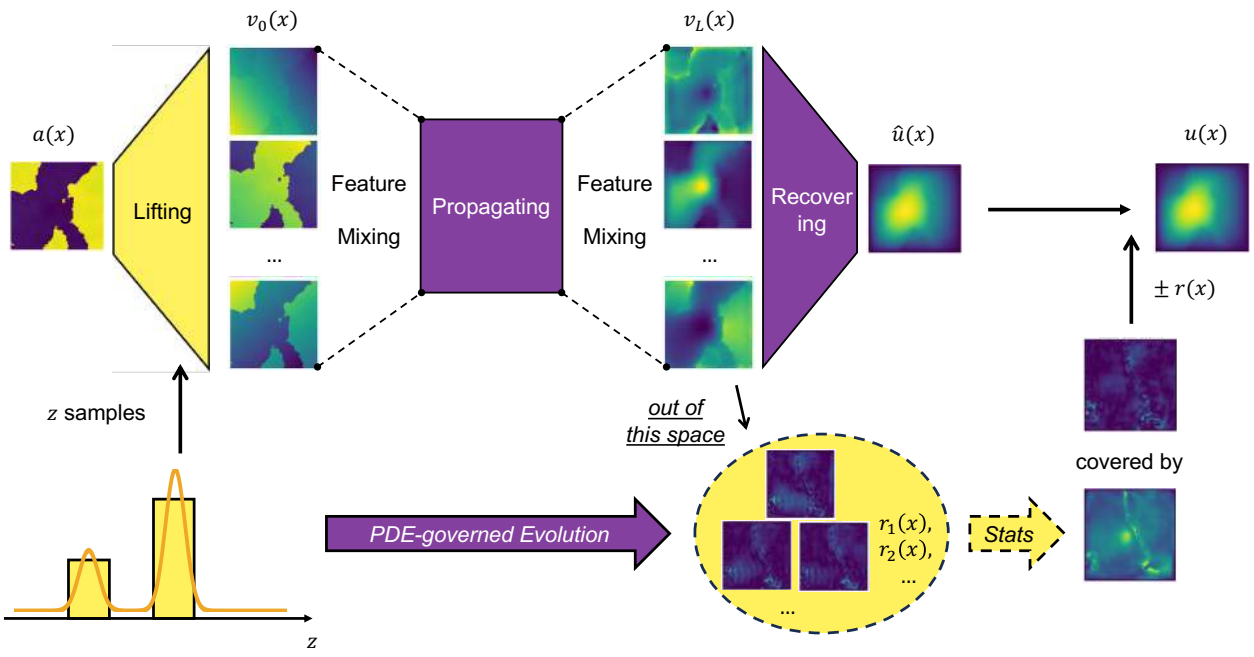

Figure 1: Overview of neural operator architectures and our structure-aware uncertainty quantification (UQ) scheme.

dimension. Its main drawback is statistical noise: accurate estimates of posterior expectations may require a large number of samples.

As illustrated in Figure 1, our goal is to approximate posterior statistics under $p(\theta \mid \mathcal{D})$ efficiently. We exploit the modular structure common to neural operators and restrict sampling to a module-aligned subspace. This structure-aware restriction targets the degrees of freedom that most directly control the epistemic residual field, yielding more faithful and sample-efficient uncertainty estimates than perturbations of other weights.

3.1 Structure-aware Subspace Sampling

To build a generic UQ method that respects the typical anatomy of neural operators, we group parameters into three modules (Section 2.1): lifting ($\mathcal{P}$), propagation ($\mathcal{M}$), and recovering ($\mathcal{Q}$). Crucially, not all parameter dimensions (or modules) are equally informative for epistemic UQ. Sampling an inappropriate set of parameters can lead to either (i) overly diverse residual samples that drift away from the typical residual structure (e.g., the outlier fields highlighted by the dashed region in Figure 2), producing conservative and inflated bands, or (ii) samples that miss key residual modes, yielding miscalibration (e.g., under-coverage unless bands are widened). In short, effective epistemic UQ benefits from identifying which parameters control which residual structures.

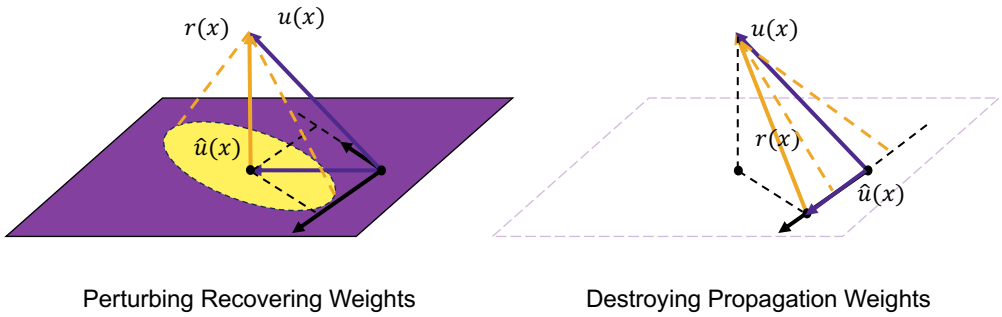

Figure 2: Sampling from other parameter groups can lead to uninformative and low-quality samples.

Why sample the lifting module? As demonstrated in Figure 2, sampling the recovering module $\mathcal{Q}$ (often a linear readout) primarily changes how latent feature channels are linearly combined at the output. Empirically, for a trained operator near a local optimum, the learned output feature space may be only weakly aligned with the dominant residual directions. Consequently, perturbing $\mathcal{Q}$ can introduce unnecessary variability, producing outlier fields that do not match the spatial organization of residuals.

On the other hand, the propagation module $\mathcal{M}$ typically contains most parameters and implements complex nonlocal transformations with nonlinearities. Unstructured perturbations in $\mathcal{M}$ can easily degrade predictive accuracy and induce biased or overly conservative uncertainty estimates due to uncontrolled error amplification through nonlinear layers.

These considerations motivate a structured projection of posterior sampling onto the lifting subspace: we inject stochasticity only through the lifting module and treat the remaining mapping as deterministic. Intuitively, lifting mainly prepares the initial embedding tensors (often through near-linear mixing of input channels) and does not itself represent the PDE evolution operator Kovachki et al. (2021). In contrast, $\mathcal{M}$ and $\mathcal{Q}$ encode the learned solver dynamics and reconstruction. Therefore, perturbing lifting features can be viewed as injecting uncertainty into initial lifted feature fields while keeping the learned evolution operator fixed. This converts epistemic uncertainty over weights into uncertainty over initial conditions in feature space, which are then propagated by a deterministic, trained operator.

Formalization. Consider a trained neural operator

$$\mathcal{G}[\cdot;\theta] = \mathcal{Q}_{\theta_\mathcal{Q}} \circ \mathcal{M}_{\theta_\mathcal{M}} \circ \mathcal{P}_{\theta_\mathcal{P}},$$

with optimized parameters $\hat{\theta}$. Given an input a, the prediction $\hat{u}(x)$ may deviate from the (deterministic) PDE solution $u(x)$ due to epistemic uncertainty, measured by the residual field $r(x)$ in Eq. equation 2. We define the deterministic remainder operator

$$\mathcal{T} \;:=\; \mathcal{Q}_{\hat{\theta}_\mathcal{Q}} \circ \mathcal{M}_{\hat{\theta}_\mathcal{M}}, \tag{8}$$

and treat it as a fixed surrogate of the target PDE solver after training. Epistemic uncertainty is represented by randomness injected into the lifting module. Concretely, we map the random variable(s) z to perturbed lifting parameters $\theta_\mathcal{P}(z)$, and perform Monte Carlo integration in this structured subspace:

$$\hat{u}(x) \approx \frac{1}{T}\sum_{t=1}^{T} \mathcal{T}\Big[\mathcal{P}_{\theta_\mathcal{P}(\mathbf{z}^{(t)})}[a]\Big](x), \tag{9}$$

$$\mathrm{band}(x) \approx \left(\frac{1}{T}\sum_{t=1}^{T}\Big(\mathcal{T}\big[\mathcal{P}_{\theta_\mathcal{P}(\mathbf{z}^{(t)})}[a]\big](x)\Big)^2 - \hat{u}^2(x)\right)^{\frac{1}{2}}, \tag{10}$$

where $\{\mathbf{z}^{(t)}\}_{t=1}^{T}$ are i.i.d. samples and $\mathrm{band}(x)$ denotes the predictive standard deviation (a scalar half-width up to a calibrated multiplier).

3.2 Sampling Strategies

We propose two practical perturbation strategies that generate stochastic lifted feature fields around the deterministic features, both constructed to be zero-mean around the deterministic embedding.

A. Channel-wise multiplicative noise on lifted features. Let the lifted feature field be $\mathbf{V}_0 := \mathcal{P}[\mathbf{A};\theta_\mathcal{P}]$, with channel dimension d_v. For dropout rate $p \in (0,1)$, sample a channel-wise mask

$$z = (z_1,\ldots,z_{d_v}), \qquad z_k \overset{iid}{\sim} \mathrm{Bernoulli}(1-p), \tag{11}$$

and use the inverted-dropout multiplier $\xi_k := z_k/(1-p)$, so that $\mathbb{E}[\xi_k] = 1$ and $\mathrm{Var}(\xi_k) = \frac{p}{1-p}$. Broadcasting ξ over spatial tokens yields a stochastic feature field

$$V_0^{(\mathrm{drop})} = \mathbf{V}_0 \odot \xi, \tag{12}$$

i.e., each feature channel is multiplied by a random scalar shared across space. This makes dropout an explicit multiplicative perturbation in lifted feature space. Equivalently, it can be written as an additive zero-mean perturbation around the deterministic feature:

$$V_0^{(\mathrm{drop})} = \mathbf{V}_0 + E, \qquad E := \mathbf{V}_0 \odot (\xi - \mathbf{1}), \qquad \mathbb{E}[E] = \mathbf{0}. \tag{13}$$

Entrywise, $\mathrm{Var}(E_{i,j}) = V_{0,i,j}^2 \cdot \frac{p}{1-p}$, i.e., the induced perturbation is heteroscedastic and proportional to the feature magnitude.

Remark (noise-perturbation view as a lifting-weight subspace). If the lifting contains an affine map $\mathbf{V}_0 = \Phi(\mathbf{A})\,\mathbf{W}_0$ (absorbing bias for simplicity), channel-wise feature dropout is algebraically equivalent to perturbing the columns of the lifting weights:

$$\mathbf{V}_0^{(\text{drop})} = \Phi(\mathbf{A})\,(\mathbf{W}_0 \text{Diag}(\xi)). \tag{14}$$

Thus, our procedure can be interpreted either as sampling a structured subspace of lifting weights or as sampling stochastic initial feature fields, followed by deterministic propagation through $\mathcal{T}$.

B. Gaussian feature perturbation. A simple continuous alternative is to perturb lifted features with Gaussian noise:

$$V_0^{(\text{gauss})} = \mathbf{V}_0 + \mathbf{V}_0 \odot \epsilon. \tag{15}$$

If $\epsilon_k \overset{iid}{\sim} \mathcal{N}\left(0, \frac{p}{1-p}\right)$ (applied channel-wise and broadcast over space), then $\mathbb{E}[V_0^{(\text{gauss})}] = \mathbf{V}_0$ while providing smooth perturbations with variance matched to the inverted-dropout scale.

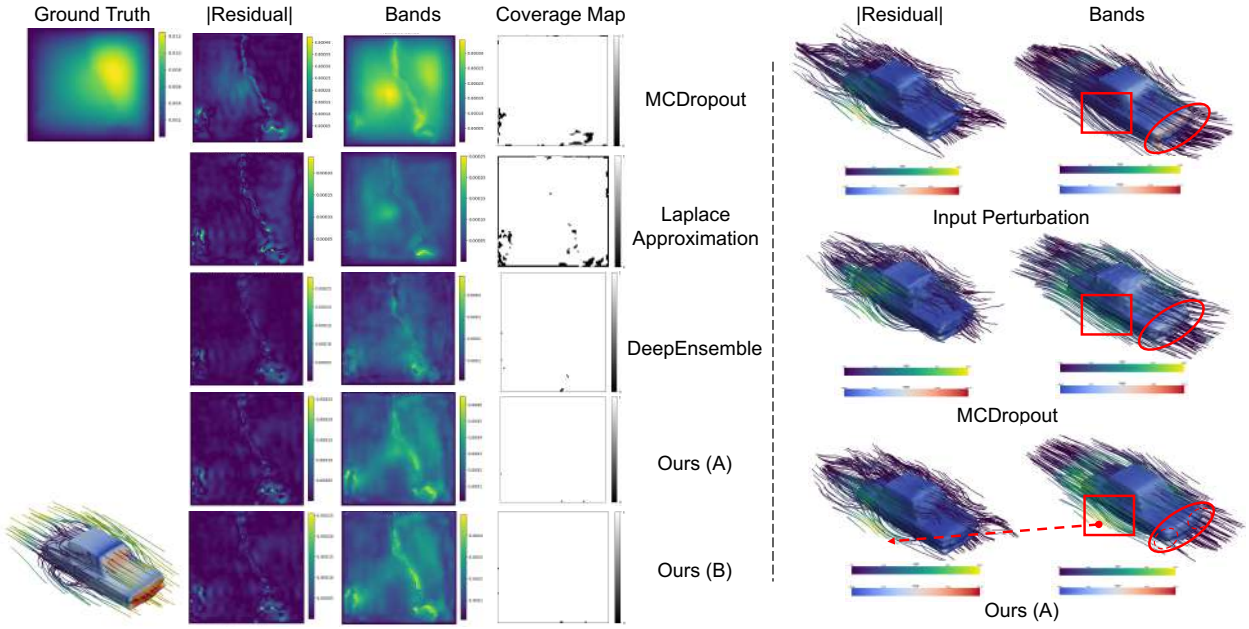

Figure 3: Visualizations of the residual and uncertainty band fields. Compared with other methods, our approach provides a more accurate characterization of the residual fields with reliable coverage.

4 Experiments

Data. We select two challenging datasets: 2D Darcy Flow Li et al. (2021) with a discontinuous coefficient field and 3D Shape-Net Car Umetani & Bickel (2018) with various car geometries. Since Neural Operators show more pronounced prediction errors/residual fields on discontinuous fields, we use 2D Darcy Flow to evaluate how different methods estimate the complex error distributions near discontinuities. Additionally, by altering the geometric features of 3D Shape-Net Car inputs, we assess the stability of epistemic uncertainty quantification (UQ) under geometry-shifted OOD inputs.

Models. We choose two representative neural operator networks: Fourier Neural Network (FNO) Li et al. (2021) and Transolver Wu et al. (2024), both of which possess the generic architecture described in Section 2.1. In the propagation layer, FNO transfers the signal into the spectral domain for propagation, with a residual MLP layer as a bias term, while Transolver uses multi-head attention and feed-forward layers to iteratively update tensor feature fields. These architectures represent common designs in current networks, with serial (FNO) and parallel (Transolver) propagation. Our UQ methods can run directly on these architectures and are adapted to the common structure and characteristics of neural operator approximations. Furthermore, the minimal parameter budget of their lifting layers is detailed in Appendix A. All experiments are conducted on a single *Nvidia GeForce RTX 4090* GPU.

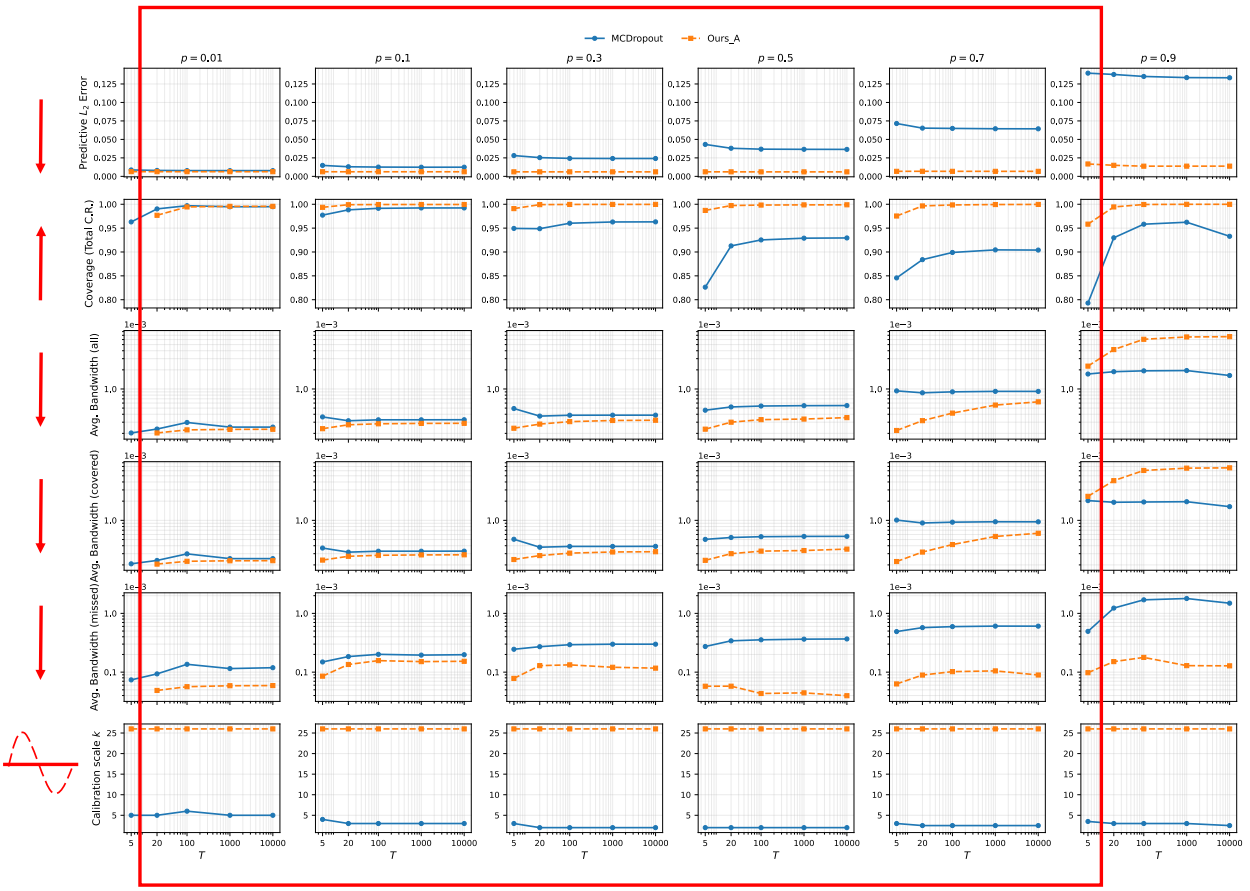

Figure 4: Comprehensive comparisons with dropout-based baselines. Notably, for each choice of p and T, MC Dropout must be recalibrated since the uncertainty bands are altered by the newly introduced stochasticity, whereas our method A can share a single constant calibration scale across all p and T settings. Note that, to obtain the case-wise Avg. B.W., one should multiply the reported values by the grid size N, which typically ranges from a few hundred to tens of thousands.

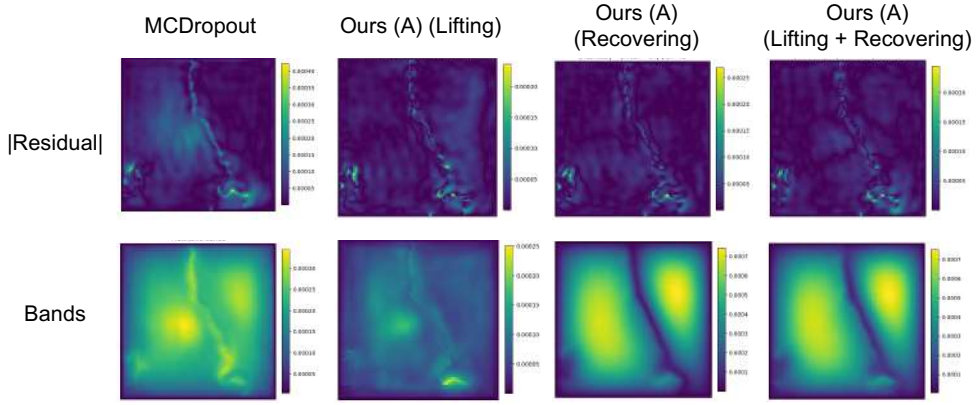

Figure 5: Demonstrations of sampling on the Lifting parameters v.s. Sampling on/with other additional choices

Quantification Results. Using the metrics from Section 2.2 and representative visualizations, we demonstrate the advanced performance and characteristics of our method in epistemic UQ for neural operators. Specifically, we first present statistics on the entire test set, including coverage rate (Equations 5 and 6) and bandwidth (Equation 7), which reflect the performance of UQ methods. As shown in Table 1, our method achieves a significantly better

compression of bandwidth while maintaining a high coverage rate, compared to the stable-performing DeepEnsemble Lakshminarayanan et al. (2017), which results in much faster UQ, providing strong support for rapid design and development.

Additionally, we visualize the calibrated bands to show the actual effects of various UQ methods. The left side of Figure 3 presents results on 2D Darcy Flow: MCDropout generates biased, conservative bands due to the loss of critical neurons, which leads to overestimation. Laplace Approximation introduces unnecessary bands in non-relevant regions, suggesting that sampling in feature spaces beyond residuals results in overestimation. DeepEnsemble is more robust than these two methods in capturing the spatial distribution of residuals. Our method, requiring significantly less time (see Table 1), achieves comparable even better coverage than DeepEnsemble.

The right side shows uncertainty quantification results for the 3D car wind-tunnel simulation (Appendix B). Input noise injection Weber et al. (2024) on the original input fails to capture the residual structure in the vector velocity field and yields inflated uncertainty on the pressure field. MCDropout is affected by biased mean predictions and redundant stochastic hypotheses; because dropout perturbs all neurons symmetrically, many samples become weakly informative and tend to average out localized turbulent patterns around the car body, resulting in globally widened and spatially smeared bands. In contrast, our method A injects stochasticity only at the lifted feature level by perturbing the learned input embeddings while keeping the trained operator fixed. This produces bands that better align with the spatial organization of residuals, with improved coverage in regions such as the headlamp pressure residuals and the side-flow velocity trends. Overall, our method provides designers with more reliable localization of epistemic uncertainty and appropriately conservative uncertainty estimates.

Comprehensive comparison with dropout-based baselines. We comprehensively compare our method A against MCDropout in Figure 4. Across a wide spectrum of dropout probabilities p and numbers of forward runs T, our method A achieves better prediction accuracy, higher (or comparable) coverage, lower bandwidth, and substantially reduced sensitivity to the calibration scale. The main exceptions occur in degenerate regimes, namely very large $p > 0.9$ or very small $T \leq 5$, where effective stochasticity can collapse (i.e., multiple stochastic forward passes become nearly identical), as illustrated in Figure 9. Overall, these results suggest that applying dropout indiscriminately to neural operators can introduce biased predictions and miscalibrated uncertainty (often manifesting as overly conservative or spatially smeared bands), whereas our structure-aware dropout sampling yields a more stable and reliable UQ mechanism for scientific computing.

Ablations. We perform an ablation study by changing the sampling module, based on our method A implementation, and applying dropout at different subspaces in the parameter space across different layers. As shown in Figure 5, sampling exclusively in the lifting subspace ensures faithful uncertainty quantification.

5 Conclusion

We presented a structure-aware approach to epistemic uncertainty quantification for neural-operator PDE surrogates, motivated by the observation that generic, structure-agnostic perturbations can be statistically inefficient and can distort residual structures that UQ is meant to capture. By sampling uncertainty only through the lifting module, interpretable as perturbing the initial lifted feature fields while keeping the learned evolution operator fixed, our method produces uncertainty bands that better track localized residual patterns, improving reliability for downstream scientific-computing workflows. Empirically, across representative benchmarks and geometrically OOD stress tests, this design improves calibration and spatial interpretability relative to standard MCDropout, Laplace-style approximations, and input perturbation, while remaining lightweight at inference time.

This work is primarily empirical and deployment-oriented rather than theoretical: our goal is a practical UQ mechanism that can be integrated into existing neural operators with minimal modification. Promising future directions include exploring richer choices of the sampling variable z (beyond the current feature-noise instantiations) and extending the structure-aware principle to coupled multi-physics and multi-field operators, where uncertainty must be propagated consistently across interacting state variables.

6 GenAI Disclosure

This manuscript was prepared with the assistance of a large language model (LLM) to improve the clarity, grammar, and overall fluency of sentences and paragraphs. The LLM was used solely for language polishing and stylistic refinement.

REFERENCES

Yiwei Feng, Kai Chen, Hao Lou, Lili Lv, Liang Xu, Xiaoguang Hu, and Bangcheng Ai. Geometry-informed neural operator for predicting surface flow field of 3d variable-geometry aerospace vehicles across a wide-speed range. *Aerospace Science and Technology*, pp. 111784, 2026.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Anima Anandkumar, and MAST Team. Fourier neural operator for plasma modelling. *arXiv preprint arXiv:2302.06542*, 2023.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems*, 2021.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pp. 12556–12569. PMLR, 2023.

Sebastián Jiménez, Mira Jürgens, and Willem Waegeman. Position: Epistemic uncertainty estimation methods are fundamentally incomplete. *arXiv preprint arXiv:2505.23506*, 2025.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pp. 5436–5446. PMLR, 2020.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence*, pp. 344–353. PMLR, 2021.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Ziqi Ma, David Pitt, Kamyar Azizzadenesheli, and Anima Anandkumar. Calibrated uncertainty quantification for operator learning via conformal prediction. *Transactions on Machine Learning Research*, 2024.

Emilia Magnani, Nicholas Krämer, Runa Eschenhagen, Lorenzo Rosasco, and Philipp Hennig. Approximate bayesian neural operators: Uncertainty quantification for parametric pdes. *Trans. Mach. Learn. Res.*, 2025, 2025. URL `https://openreview.net/forum?id=6WvIkYsMA8`.

Zhong Peng, Bo Yang, Yixian Xu, Feng Wang, Lian Liu, and Yi Zhang. Rapid surrogate modeling of electromagnetic data in frequency domain using neural operator. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023.

Milad Ramezankhani, Anirudh Deodhar, Rishi Yash Parekh, and Dagnachew Birru. An advanced physics-informed neural operator for dynamic, zero-shot, and near real-time simulation of aerospace composite material curing process. In *AI for Accelerated Materials Design-Vienna 2024*, 2024.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th international conference on learning representations, ICLR 2018-conference track proceedings*, volume 6. International Conference on Representation Learning, 2018.

Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

Tobias Weber, Emilia Magnani, Marvin Pförtner, and Philipp Hennig. Uncertainty quantification for fourier neural operators. In *ICLR 2024 workshop on AI4DifferentialEquations in science*, 2024.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *Forty-first International Conference on Machine Learning*, 2024.

A Neural Operators

Loss We aim to approximate the operator $\mathcal{G}$ in Section 2.1 by optimizing the model parameters $\theta \in \Theta$ through relative L_2 loss below:

$$\min_{\theta\in\Theta} \mathcal{L}(\theta) := \min_{\theta\in\Theta} \frac{1}{M} \sum_{m=1}^{M} \left[\frac{\|\mathcal{G}_\theta[a_m] - u_m\|_2}{\|u_m\|_2} \right] \tag{16}$$

$$= \min_{\theta\in\Theta} \frac{1}{M} \sum_{m=1}^{M} \left[\frac{\|\mathcal{G}[\mathbf{A}_m;\theta] - \mathbf{U}_m\|_F}{\|\mathbf{U}_m\|_F} \right], \tag{17}$$

where Θ, M, and $u_m = u_m(x)$ denote the parameter space, the number of function samples (in one batch), and the m-th output function in M samples, respectively.

Parameter allocation Since the lifting operator $\mathcal{P}[\cdot;\theta_\mathcal{P}]$ is typically implemented as a channel-wise fully-connected map, its parameter budget is negligible compared with the remaining blocks. This remains true even when positional encoding (e.g., concatenated absolute or reference coordinates) is included in the input channels.

For instance, on 2D Darcy, the lifting layer of an FNO contains only 128 parameters out of $1{,}196{,}801$ in total, accounting for just 0.0107% of the model. In contrast, architectures with heavier input embeddings (e.g., Transolver) allocate a larger yet still relatively small fraction to the lifting stage: Transolver uses $16{,}896$ lifting parameters out of $2{,}826{,}945$ total, i.e., 0.598%.

B Benchmarks

Here we select two challenging benchmarks: one with discretized fields on a uniform mesh, and another with fields defined on a non-uniform grid featuring out-of-distribution (OOD) car geometries.

2D Darcy Flow The 2D Darcy Flow is a second-order elliptic equation as follows:

$$-\nabla \cdot (a \nabla u) = f, \tag{18}$$

where $f = 1$. Our goal is to approximate the mapping operator from the spatially-distributed coefficient field $a(x)$ to the solution function $u(x)$ with zero Dirichlet boundary conditions Li et al. (2021). Models are trained on 1000 training samples, which are down-sampled to a size of (85×85), and tested on 100 samples.

3D ShapeNet Car The 3D ShapeNet Car benchmark targets aerodynamic performance prediction for vehicle geometries from the ShapeNet car subset, using CFD simulations conducted at a fixed inlet speed of 72 km/h Umetani & Bickel (2018). Given a car shape (represented on a fixed computational mesh), the task is to learn the surrogate operator that maps geometry-conditioned inputs to the corresponding 3D flow fields—specifically the velocity field and pressure field. The dataset contains 889 simulated samples, split the out of distribution 100 cases with unseen shapes for testing.

C Limitations and Ethical Considerations

Limitations. (i) Scope of uncertainty: We focus on epistemic uncertainty in largely deterministic PDE settings; the method does not explicitly model aleatoric uncertainty arising from internal numerical errors of the reference solver (e.g., discretization and iterative tolerance effects). (ii) Calibration transfer: The calibration scale is tuned on held-out

data; under severe distribution shift, calibrated bands may no longer maintain nominal coverage, so additional OOD detection, periodic recalibration, or conservative fallback rules may be required. (iii) Empirical focus: This work is driven by practical deployability and empirical evidence, rather than formal bounds under reasonable assumptions. A probabilistic theoretical exploration for general neural operators is left for future work.

Ethical considerations. We do not foresee explicit ethical issues specific to this work. The proposed method improves uncertainty reporting for scientific-computing surrogates and is intended to support more reliable deployment (e.g., flagging cases for additional high-fidelity simulation), rather than replacing established verification and validation practices.

D Epistemic Uncertainty Quantification

D.1 Showcases

MCDropout vs. Ours-A. MCDropout is implemented by inserting dropout layers before every linear layer / linear operator. We visualize the predictive mean and uncertainty band fields of MCDropout Gal & Ghahramani (2016) and our implementation of Method A (Section 3.2) under different numbers of samples T and dropout probabilities p in Figures 6–8 and Figures 9–11. The corresponding coverage maps are shown alongside each set of results.

Protocol. Bandwidth is only comparable at equal coverage, since any band can be inflated until it covers. For each method the global scale k is chosen on a calibration split, disjoint from the test split, by bisection so that the coverage of $k\,b(x)$ equals $\tau = 0.999$, and all methods share τ and the splits. Test coverage is 0.998–1.000 unless shown in parentheses. Last-layer Laplace is scaled by calibration likelihood rather than coverage, so it is listed with the coverage it achieves. Defaults are $p = 0.1$, $T = 200$, and $M = 5$. rel-L_2 is the error of each method's point prediction (the Monte Carlo mean of Eq. 9 for sampling methods). Values are not comparable with Table 2, which uses $T = 20$ and normalized bandwidths.

Additional architectures and benchmark. UNO Rahman et al. (2023) is a U-shaped operator with multi-resolution propagation and MWT Gupta et al. (2021) is a multiwavelet operator. Both share the lifting–propagation–recovering structure of Section 2.1. The *Airfoil* benchmark is steady compressible flow around NACA-0012 shapes morphed through eight control nodes with displacements $d \sim \mathcal{U}[-0.05, 0.05]$, and its Mach-number fields span 0.10–1.79, so small shape changes move shocks. The ablation rows apply the channel-wise dropout of Method A to $\mathcal{M}$ alone or to $\mathcal{Q}$ alone, with p, T, and calibration fixed. The $\mathcal{Q}$-only row is the "Ours (A) (Recovering)" setting of Figure 5.

Reading Table 3. (i) Our method gives the tightest single-model band in seven of eight architecture–benchmark pairs. The exception is UNO–Airfoil, where $\mathcal{M}$-only dropout is 2% tighter but 8% less accurate. (ii) It matches or beats deep ensembles in seven of eight pairs at one fifth of their training cost, while ensembles keep the lowest point-prediction error, as expected from averaging five models. (iii) MC Dropout widens the band 2–$5\times$ on Darcy and 1.5–$8\times$ on Airfoil and raises the point-prediction error up to $2\times$ (FNO–Darcy) and $4\times$ (FNO–Airfoil), because perturbing the solver biases the sample mean. $\mathcal{M}$-only dropout reproduces this, so the failure lies in the propagation module. $\mathcal{Q}$-only dropout tracks the magnitude of the final latent field rather than the residual (exactly so for a linear readout) and is either wider than ours or under-covers (MWT– and Transolver–Darcy). Laplace under-covers on every Darcy architecture and, on Airfoil, needs bands 28–$60\times$ wider than ours.

Misspecified remainder operator. Can a lifting-only perturbation signal errors that originate in the propagation module? We build hybrid operators $\mathcal{T}_{AB}$ whose lifting module is trained on the Darcy coefficient distribution A and whose remainder $\mathcal{Q}\circ\mathcal{M}$ is trained on a shifted distribution B (amplitude attenuation or amplification, spectral shift, or a change of coefficient distribution). Table 4 reports the hybrid's error and the alignment of our band (Method A) with the residual, as Spearman rank correlation and cosine similarity over the grid. Alignment is higher in every misspecified case than for the unmodified model and highest (0.84 Spearman, 0.90 cosine) for the largest misspecification: the band is propagated through the misspecified remainder and inherits the spatial structure of its errors. One scalar recalibration restores the target coverage in every case.

3D car flow: derived coefficients. Table 5 restates the field-level results of Table 2(B) for the Transolver surrogate on 100 held-out shapes, with the calibration scale transferred unchanged from in-distribution shapes, and adds the drag coefficient C_d and the skin-friction coefficient C_f. Our estimator keeps 0.95 total coverage under geometry shift, against 0.78 for MC Dropout and 0.55 for input perturbation, at comparable bandwidth. For C_d and C_f all methods

reach full coverage, and our bands are $4.7\times$ tighter for C_d and 4.7–$9.9\times$ tighter for C_f than those of MC Dropout. Inference adds 7 minutes for $T = 50$ to a 6-hour training run.

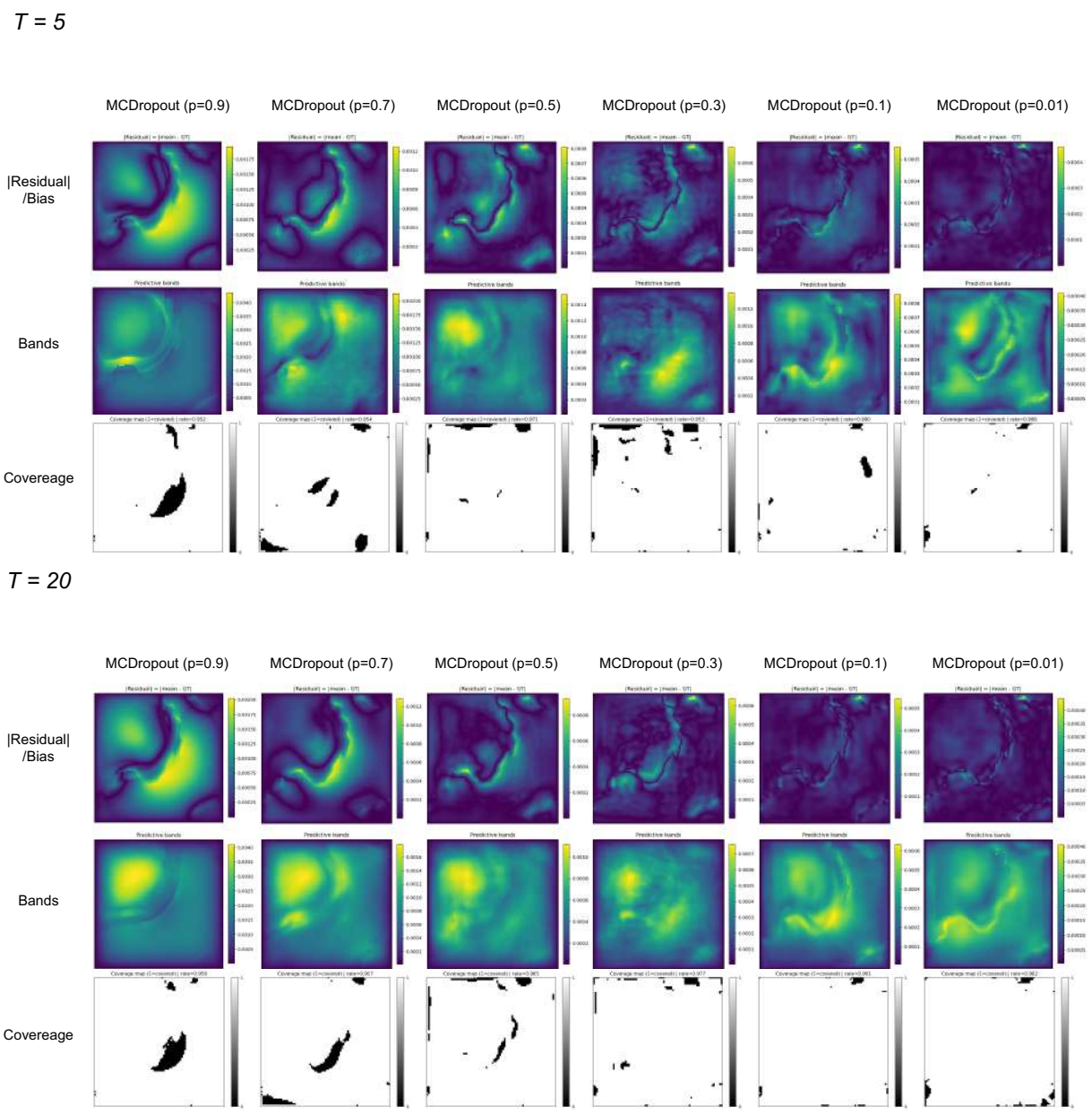

Figure 6: Naive MCDropout uncertainty for different dropout probabilities p and sample counts T. Higher p causes more biased predictions and lower coverage of the calibrated uncertainty bands.

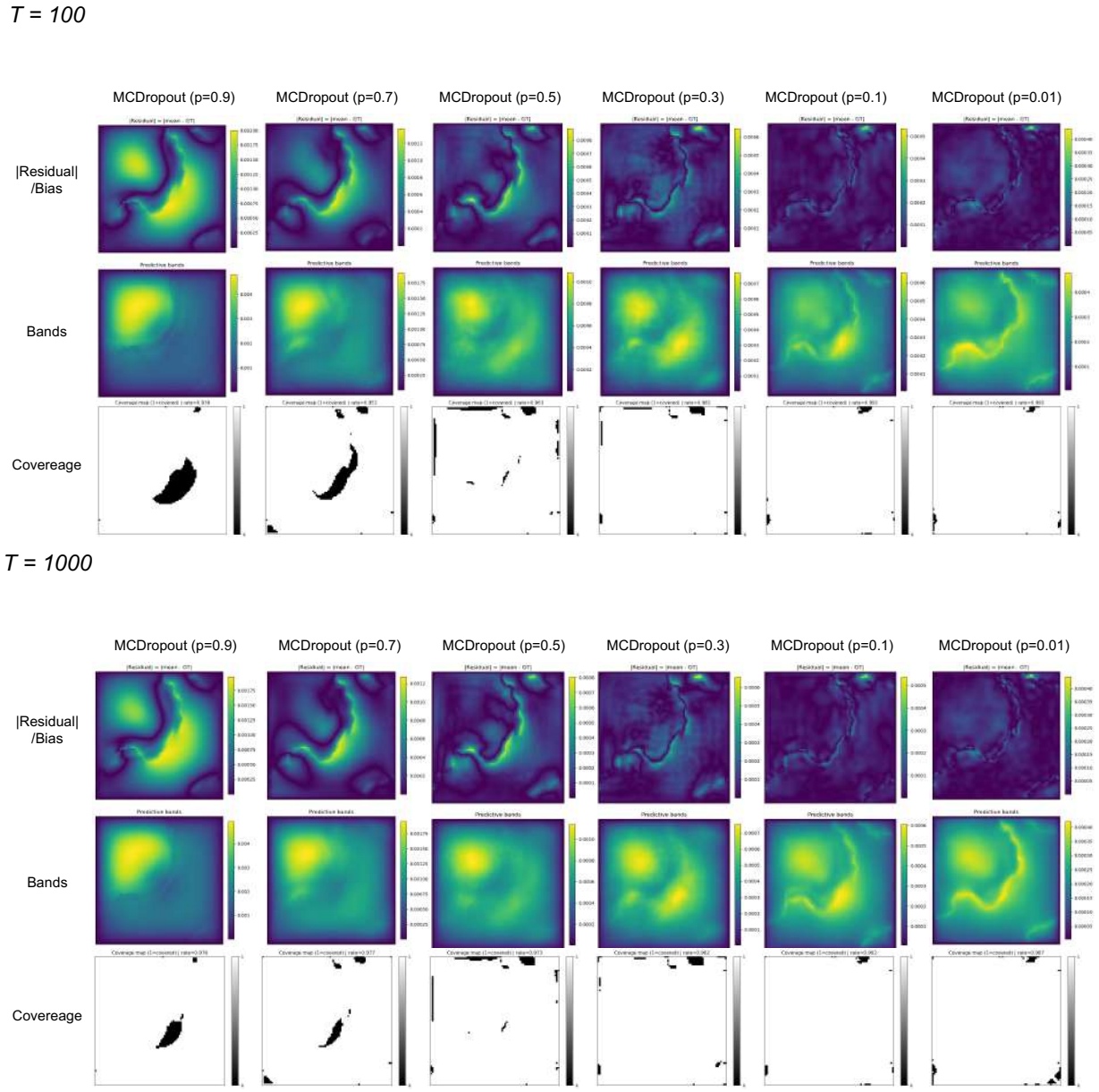

Figure 7: Naive MCDropout uncertainty for varying dropout probability p and sample counts T. Relative to the smaller-T results in Figure 6, increasing T yields a smoother, re-distributed uncertainty-band field.

Table 2: Uncertainty quantification (UQ) results on two challenging benchmarks.

(A) 2D Darcy Flow with Discontinuity

Metric	**MCDropout** ($T = 20$)	**Laplace Approx.**	**DeepEnsemble**	**Ours (A)** ($T = 20$)	**Ours (B)** ($T = 20$)
Total C. R. (Channel-wise C. R.)	0.990246	0.869001	<u>0.998399</u>	**0.999031**	<u>0.996810</u>
Normalized Avg. B.W. (all)	2.34	0.77	1.73	2.73	2.03
Normalized Avg. B.W. (covered)	2.35	0.82	1.73	2.73	2.03
Normalized Avg. B.W. (missed)	0.93	0.48	1.55	1.35	1.26
Time Cost of UQ	$\approx$ 10 min train + 1 min infer	$\approx$ 10 min train + 1 min Hess.	$>$ 100 min train	$\approx$ 10 min train + 0.5 min infer	$\approx$ 10 min train + 0.5 min infer

(B) 3D ShapeNet Car with Various Geometries

Metric	**Input Perturbation**	**MCDropout**	**Ours (A)** ($T = 50$)
Total C. R. (Channel-wise C. R.)	0.5516 / 0.5656 (P) / 0.9409 (V)	0.7761 / 0.7871 / 0.9852	**0.9514 / 0.9605 / 0.9905**
Normalized Avg. B.W. (all)	15.432636 / 57.799948 (P) / 1.310198 (V)	19.053221 / 71.173860 / 1.679674	17.702314 / 68.240521 / 0.856245
Time Cost of UQ	$\approx$ 6 h train + 4 min infer	$\approx$ 6 h train + 7 min infer	$\approx$ 6 h train + 7 min infer

Table 3: Comparison at matched coverage ($\tau = 0.999$) on 2D Darcy flow and transonic airfoil. Test coverage is 0.998–1.000 unless shown in parentheses. rel-L_2: error of the point prediction. B.W.: average band half-width (*lower is better*), $\times 10^{-4}$ for Darcy and $\times 10^{-2}$ for Airfoil. Best bandwidth per column among methods reaching the target coverage in **bold**. $p = 0.1$, $T = 200$, $M = 5$.

	FNO		UNO		MWT		Transolver	
Method	rel-L_2	B.W.	rel-L_2	B.W.	rel-L_2	B.W.	rel-L_2	B.W.
(a) 2D Darcy flow, discontinuous coefficient								B.W. $\times 10^{-4}$
MC Dropout	0.0122	13.5	0.0130	11.3	0.0205	54.5	0.0176	43.3
Laplace (last layer)	0.0060	2.03 (0.928)	0.0084	2.25 (0.897)	0.0164	1.29 (0.487)	0.0130	2.67 (0.863)
Deep ensembles (M=5)	0.0055	5.54	0.0076	7.87	0.0129	23.6	0.0111	21.5
Dropout in $\mathcal{M}$ only	0.0121	12.3	0.0156	10.5	0.0180	29.5	0.0168	31.2
Dropout in $\mathcal{Q}$ only	0.0069	57.7	0.0086	29.7	0.0167	17.4 (0.974)	0.0137	13.8 (0.985)
Ours (A)	0.0060	5.54	0.0089	**5.20**	0.0179	**21.2**	0.0136	**9.05**
Ours (B)	0.0060	**4.70**	0.0089	5.58	0.0179	24.1	0.0136	12.8
(b) Airfoil, transonic								B.W. $\times 10^{-2}$
MC Dropout	0.0302	13.9	0.0232	11.6	0.0361	21.8	0.0159	6.02
Laplace (last layer)	0.0097	121	0.0164	220	0.0285	126	0.0056	216
Deep ensembles (M=5)	0.0069	4.83	0.0118	**7.02**	0.0241	24.4	0.0047	4.33
Dropout in $\mathcal{M}$ only	0.0318	14.3	0.0200	7.63	0.0389	22.7	0.0113	4.07
Dropout in $\mathcal{Q}$ only	0.0183	17.3	0.0154	10.2	0.0243	36.7	0.0064	7.88
Ours (A)	0.0078	**4.02**	0.0186	7.76	0.0264	**2.64**	0.0089	**3.61**
Ours (B)	0.0078	4.22	0.0186	8.24	0.0268	2.79	0.0104	5.03

Table 4: Band alignment under a misspecified remainder operator (FNO, 2D Darcy, Method A). $\mathcal{T}_{AB}$: lifting trained on A, remainder on B. AA: unmodified model. Mean $\pm$ std over test cases, $T = 200$, $p = 0.1$.

Remainder trained on	rel-L_2	Spearman(band, $\lvert r\rvert$)	cosine(band, $\lvert r\rvert$)
AA (no misspecification)	0.006	0.433 ± 0.113	0.743 ± 0.044
AB: amplitude attenuation	0.087	0.622 ± 0.133	0.841 ± 0.046
AB: spectral shift	0.102	0.623 ± 0.131	0.787 ± 0.060
AB: amplitude amplification	0.124	0.499 ± 0.170	0.766 ± 0.089
AB: distribution shift	0.227	0.838 ± 0.080	0.904 ± 0.039

Table 5: Geometry-shifted 3D car flow (Transolver, 100 held-out shapes, $T = 50$). Left: total, pressure (P), and velocity (V) coverage with the calibration scale transferred from in-distribution shapes, and normalized average bandwidth as in Table 2. Right: coverage and band half-width for C_d ($\times 10^{-4}$) and C_f ($\times 10^{-5}$). Field-level metrics for Ours (B) are not included.

	Coverage			Norm. B.W.			C_d ($\times 10^{-4}$)		C_f ($\times 10^{-5}$)	
Method	all	P	V	all	P	V	Cov.	Band	Cov.	Band
Input perturbation	0.552	0.566	0.941	15.4	57.8	1.31	1.00	5.8	1.00	1.6
MC Dropout	0.776	0.787	0.985	19.1	71.2	1.68	1.00	5.6	1.00	8.0
Ours (A)	**0.951**	**0.961**	**0.991**	17.7	68.2	0.86	1.00	**1.2**	1.00	1.7
Ours (B)	–	–	–	–	–	–	1.00	**1.2**	1.00	**0.81**

T = 10000

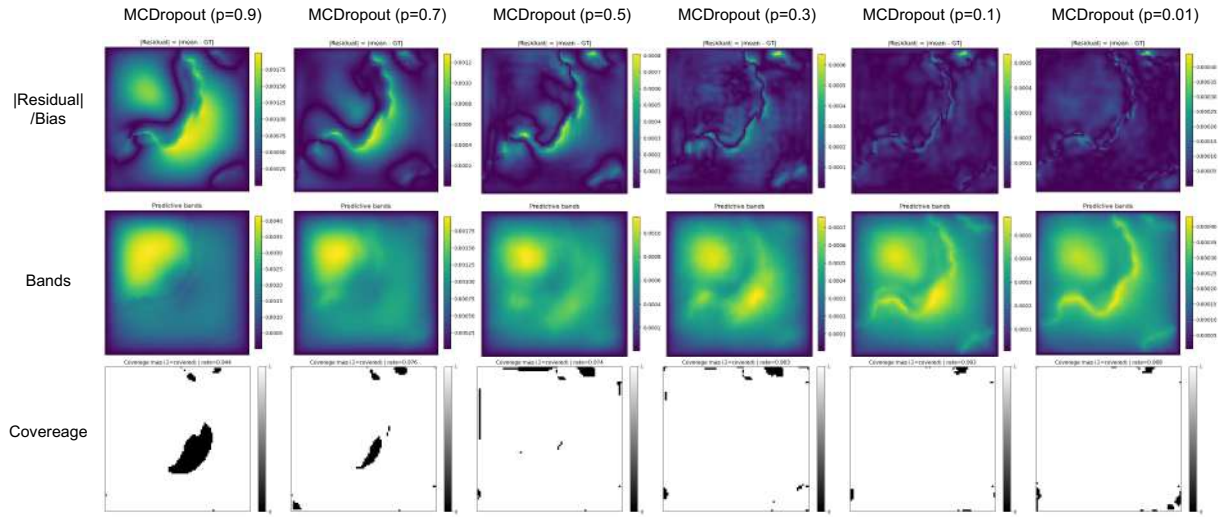

Figure 8: Naive MCDropout uncertainty for varying dropout probability p with a relatively large sample count $T = 10,000$. The near-identical predictive means, uncertainty bands, and coverage indicate that further increasing T has a negligible impact on the estimated mean and uncertainty band.

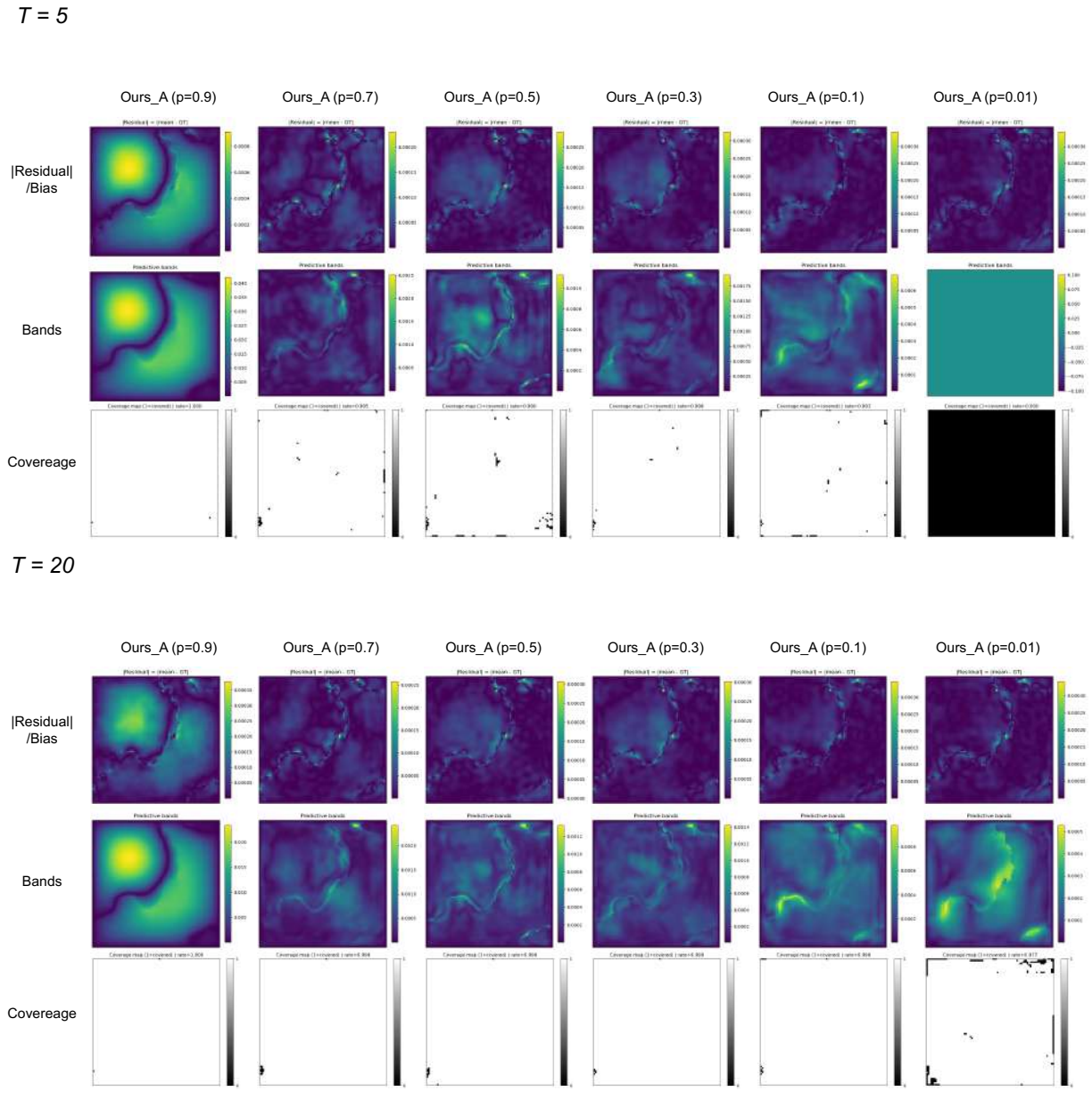

Figure 9: Quantified uncertainty visualizations of the A-implementation of our method for varying dropout probabilities p and sample counts T. With an appropriate dropout rate (i.e., neither extremely small nor large), our approach yields consistently far less biased predictions and higher coverage with calibrated uncertainty bands than the naive MCDropout results in Figure 6. Notably, the calibration scale k is shared across all p values in our method A, whereas naive MCDropout typically requires case-by-case tuning of k.

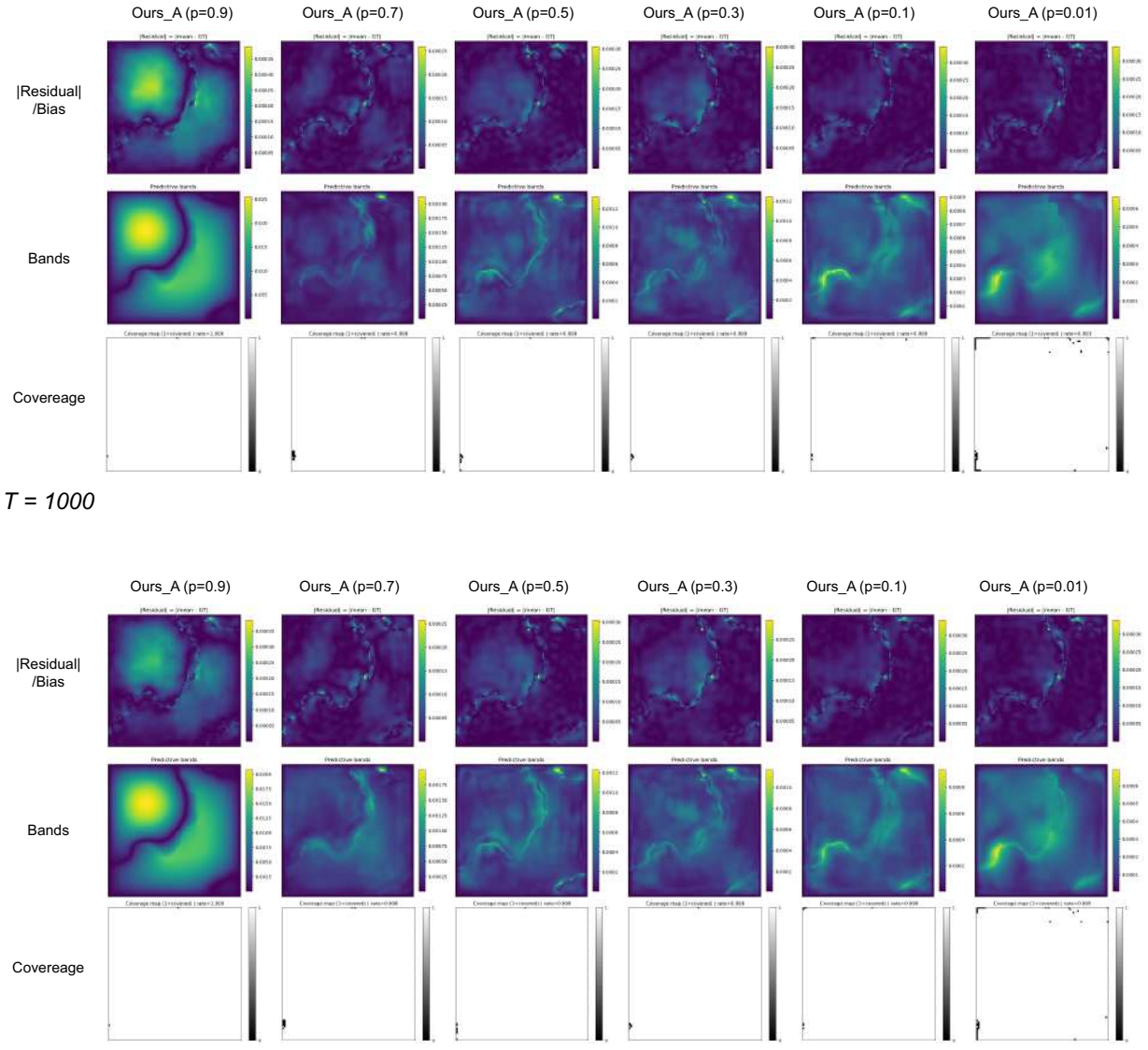

Figure 10: Quantified uncertainty visualizations of the A-implementation of our method for varying dropout probabilities p and sample counts T. An appropriate dropout rate (i.e., neither extremely small nor large) provides reliable uncertainty quantification even with small T shown in Figure 9. Increasing T to 100 or 1000 offers little benefit for normal p but helps reduce under-coverage when the dropout probability is very small ($p = 0.01$).

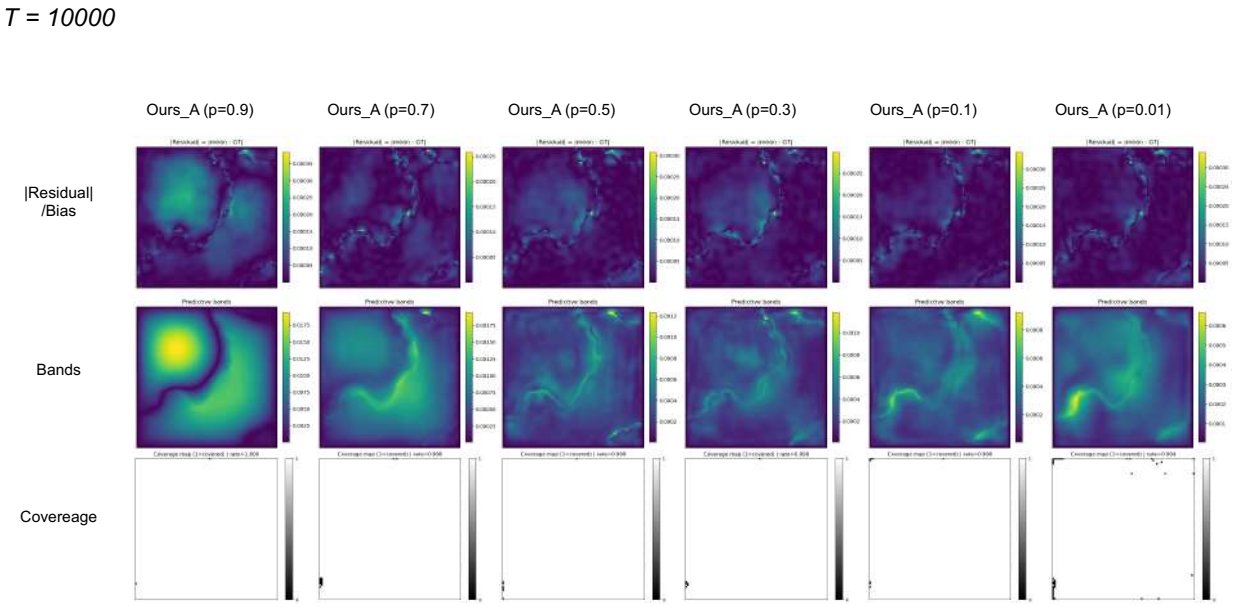

Figure 11: Quantified uncertainty visualizations of the A-implementation of our method for varying dropout probabilities p and sample counts T. Compared with the previously small T cases, further increasing T to 10000 has fewer benefits on coverage. However, compared with the uncertainty bands in Figure 8, the A-implementation of our method provides a more faithful estimation of the residual field and achieves consistently much higher coverage.

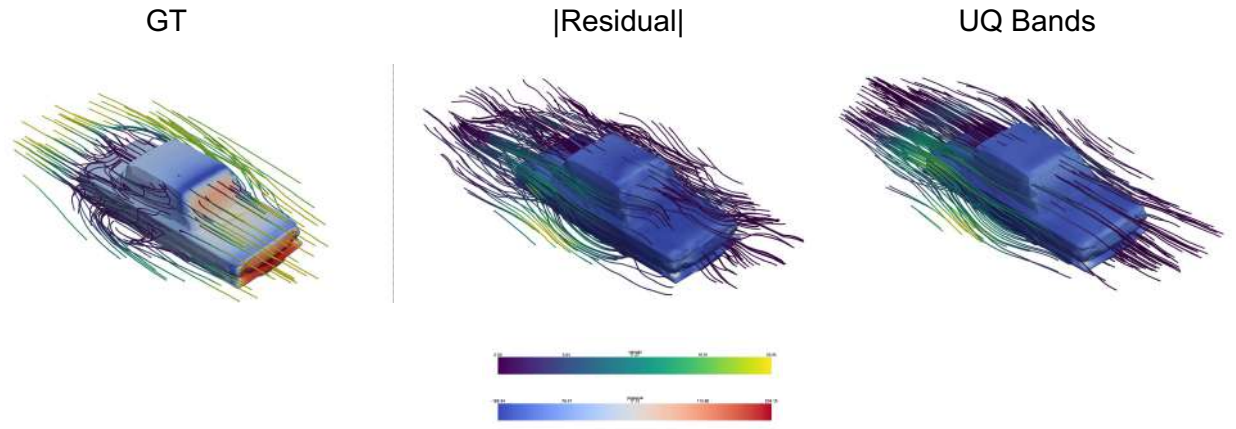

Figure 12: UQ visualizations of our method B on the *3D ShapeNet Car* with C.R.: 0.9162 (*All*) / 0.9248 (*Press*) / 0.9896 (*Velocity*) and Normalized Avg. B.W.: 17.530033 (*All*) / 0.970439 (*Press*) / 67.208816 (*Velocity*).