JENO: FULL-FIELD LATENT PREDICTION FOR SPARSE INVERSE PDE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Whole-field PDE recovery under partial observation requires inferring complete physical states from sparse and incomplete evidence. Deterministic neural operators such as FNO provide efficient one-pass inference, but often struggle under sparse observations because they primarily learn direct input-output regression rather than representations of the underlying physical state manifold. Probabilistic Diffusion-based approaches learn stronger generative priors for ill-posed PDE inference, but require expensive iterative sampling during inference. In this work, we propose **JENO**: **J**oint-**E**mbedding **N**eural **O**perators, a deterministic framework that learns latent operators from partial observations to representations of complete physical fields through joint-embedding predictive learning. Instead of directly regressing solutions in pixel space, JENO learns structured latent representations of the full state. As a result, JENO is not rewarded for just matching observed outputs, but explicitly encouraged to learn a globally consistent representation of the underlying physical state. Importantly, JENO combines the strengths of deterministic neural operators and probabilistic diffusion models. Extensive experiments show that JENO is nearly as efficient as standard neural operators while being more than 2× more accurate. At the same time, JENO is competitive with or surpasses diffusion models in accuracy while being four orders of magnitude faster than diffusion-based approaches. Overall, our results suggest that joint-embedding predictive learning offers a strong non-probabilistic alternative, establishing a new paradigm for PDE inference under partial observations.

1 INTRODUCTION

Partial differential equations (PDEs) describe physical systems across science and engineering (Evans, 2010; Quarteroni, 2009; Karniadakis et al., 2021). Inverse problems seek to recover coefficients, source terms, or initial states from measurements of a PDE solution (Kaipio & Somersalo, 2005; Tarantola, 2005; Stuart, 2010). In practice, these measurements often cover only a small fraction of the domain, whereas the desired output is a complete physical field. We study this setting with **paired complete fields available during training and sparse observations at inference** (Sec. 2.1). A motivating scenario is repeated inference for a system that can be simulated offline but is only partially measured during deployment. Sparse measurements introduce two related difficulties: **(i) ambiguity**, because distinct PDE instances may agree at the observed locations, and **(ii) sensitivity**, because small measurement perturbations can substantially change the inferred quantities (Kaipio & Somersalo, 2005; Stuart, 2010). Learning must therefore exploit regularities across complete training instances while making predictions from the limited evidence available at test time.

Neural operators provide an efficient approach to repeated PDE inference by learning mappings between functions (Li et al., 2021a; Lu et al., 2021; Kovachki et al., 2023). Their predictions require only a forward pass, but their accuracy under sparse observations depends on both the observation representation and the training procedure. Grid-based models can encode measured values together with masks and coordinates, while coordinate-based and neural-field operators can process sparse sensor data without constructing a completed input field (Lu et al., 2021; Serrano et al., 2023). These architectures offer different ways to represent the available evidence; none removes ambiguity arising from insufficient measurements. Diffusion-based methods, including DiffusionPDE and FunDPS, instead learn distributions of complete PDE fields and incorporate partial measurements through guided sampling (Huang et al., 2024; Yao et al., 2025). This is useful when multiple plausible

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.

solutions are needed, but iterative sampling can be expensive when the application requires a single dense estimate. For that setting, an important question is how to use complete training fields to improve deterministic prediction from sparse inputs.

We investigate a complementary learning strategy: *Can predicting representations of complete physical fields improve sparse PDE recovery while retaining one-pass inference?* We introduce **JENO**: **J**oint-**E**mbedding **N**eural **O**perators, adapting joint-embedding predictive learning to PDE inference (Assran et al., 2023; Baevski et al., 2022; Grill et al., 2020). During pretraining, a student encodes sparse measurements, observation masks, task information, and coordinates. A teacher processes the complete paired fields and supplies stop-gradient latent targets; its encoder tracks the student encoder through an exponential moving average. The student predicts these targets with a normalized latent objective that separately weights sampled unobserved positions and the remaining latent grid. This stage explicitly trains sparse-input features to predict representations of the complete PDE instance, without a dense-field reconstruction loss. The pretrained student encoder is then fine-tuned jointly with a task-specific readout using dense targets (Sec. 3). Thus, the distinction from direct supervised operators is an additional predictive representation-learning stage, not access to additional complete training fields. **At inference, the teacher and pretraining predictor are discarded**: the adapted encoder and readout recover the desired field from sparse observations in one deterministic pass.

Our contributions are threefold. **(1) Full-field predictive representation learning.** We introduce a two-stage framework for sparse PDE inference that combines complete-field teacher targets, spatially weighted latent prediction, and task-specific dense recovery. The framework supports both inverse recovery and forward prediction, with implementations using FNO, CNO, and U-Net backbones. **(2) Conditional error analysis.** For fixed student–teacher networks, we show how spatial supervision coverage controls full-grid normalized latent error. We then derive a conditional downstream field-error bound that accounts for teacher decoding quality, latent scale, transfer to the inherited encoder representation, and changes during fine-tuning (Sec. 3.2; Appendix A). **(3) Empirical evaluation of sparse PDE recovery.** Across four Partial-OBS PDE families, JENO has the lowest reported inverse errors and lower reported errors than the evaluated mask-aware deterministic baselines on all eight forward/inverse endpoints, under the respective protocols. Inference takes approximately $5.5\,\mathrm{ms}$ on the timed Poisson and Navier–Stokes tasks, more than $50{,}000\times$ faster than the evaluated DiffusionPDE and FunDPS samplers (Sec. 4). Additional experiments examine backbone and readout choices, observation density, and measurement-noise sensitivity (Appendix C).

2 Preliminaries and Related Work

2.1 PDE-based Inverse Problems and PDE Inference under Partial Observations

PDE-based inverse problems seek to recover an unknown coefficient, source, or initial state from measurements of the corresponding PDE solution (Kaipio & Somersalo, 2005; Tarantola, 2005; Stuart, 2010). Let $g \in \mathcal{G}$ denote the problem-defining field and $f \in \mathcal{F}$ its associated response, where $\mathcal{G}$ and $\mathcal{F}$ are appropriate function spaces. Depending on the task, f may be a steady-state solution or a state at a prescribed later time. We write $a = (g, f) \in \mathcal{A}$ for the complete paired PDE instance. The governing equations and their boundary or initial conditions are specific to each problem and are given in Appendix B.1.5.

Under *partial observations*, only measurements of one field are available. Let S select the observed field, T select the prediction target, and I describe the observation mechanism, including the selected field and sensor locations. A schematic observation model is

$$u = \mathcal{O}_I\big(S(a)\big) + \varepsilon, \qquad X = (u, I), \qquad Y = T(a), \tag{1}$$

where $\mathcal{O}_I$ is the observation operator and ε represents measurement noise. In the noiseless case, the **forward task** predicts f from $(\mathcal{O}_I(g), I)$, whereas the **inverse task** predicts g from $(\mathcal{O}_I(f), I)$. These are prediction tasks: sparse measurements need not determine a unique target.

Equation (1) is illustrative rather than a restriction on the analysis. More generally, we use the joint law of a and $X \sim P(\cdot \mid a)$, allowing random observation patterns and state-dependent noise; field-risk statements assume square-integrable targets and predictions. The main benchmark uses noiseless point measurements with known masks. Throughout, we study **partial-observation inference with**

paired complete-field training data: complete instances are available during training, but predictions at deployment use only X. Section 3.1 specifies the corresponding discretized inputs.

2.2 Related Work

Neural operators and sparse observations. Neural operators learn mappings between function spaces and provide efficient surrogates for PDE solution operators (Li et al., 2021a; Kovachki et al., 2023; Lu et al., 2021; Li et al., 2021b; Rahman et al., 2023). Their treatment of partial observations depends on the input representation and training procedure. Grid-based implementations, such as FNO, can receive zero-filled measurements together with binary masks and coordinates. The mask distinguishes unobserved entries from measured zeros; zero filling specifies an input representation rather than reconstructing a complete physical field. JENO uses this same type of mask-aware grid input. Other architectures need not construct such a grid: standard DeepONet encodes sampled input-function values and uses a coordinate trunk to evaluate the output (Lu et al., 2021), while CORAL uses coordinate-based neural fields for operator learning on general geometries (Serrano et al., 2023).

Methods have also been developed specifically for learning from incomplete data. LANO (Hou et al., 2026) combines mask-to-predict training with physics-aware latent autoregressive propagation to learn from partially observed training trajectories. This differs from our setting, which provides paired complete fields during training and evaluates sparse-to-dense forward and inverse recovery. Its POBench-PDE results are therefore relevant context, but are not directly interchangeable with Partial-OBS results. Our experiments include a standard sensor-coordinate DeepONet alongside grid-based controls (Table 3); comparisons are restricted to the evaluated implementations and protocols.

Generative PDE inference. A probabilistic formulation describes the distribution of physical fields compatible with the available measurements (Tarantola, 2005; Stuart, 2010). Diffusion-based inverse solvers combine learned priors with observation-dependent conditioning or guidance (Chung et al., 2023; Kawar et al., 2022). For PDE problems, DiffusionPDE (Huang et al., 2024) models the joint distribution of problem-defining and solution fields, while FunDPS (Yao et al., 2025) develops guided diffusion sampling in function spaces using discretization-agnostic neural operators. Related approaches incorporate physics-informed spectral representations (Gallon et al., 2026), PDE-residual-informed architectures (Sawhney et al., 2025), equivariance regularization (Tolooshams et al., 2025), or a decoupled coefficient prior and forward operator (Lin et al., 2026). These methods offer different ways to exploit complete training fields and impose measurement or physical constraints during inference.

Predictive representation learning. Joint-embedding methods learn representations by matching or predicting target-network features (Grill et al., 2020; Baevski et al., 2022; Assran et al., 2023). In particular, I-JEPA predicts target-block representations from an observed context rather than reconstructing raw pixels (Assran et al., 2023). JENO adopts this predictive principle for paired PDE fields: sparse measurements form the student input, and the complete instance supplies training-only teacher targets. The resulting encoder is subsequently adapted to dense field prediction. Thus, JENO uses complete-field supervision for representation learning; it does not learn solely from sparse training observations or replace downstream supervised recovery.

2.3 Deterministic Prediction and Conditional Sampling

Conditional sampling and deterministic prediction address different inference objectives. A conditional model aims to represent the distribution of Y given X, whereas a deterministic predictor returns a single estimate $\widehat{Y}(X)$. Sampling is useful when several physical states are compatible with the observations. When an application requires one dense estimate, however, two practical considerations arise: ① **Inference cost.** The diffusion-based PDE solvers considered here require repeated denoiser evaluations, often with additional guidance computations, rather than a single prediction pass (Huang et al., 2024; Yao et al., 2025). ② **Guidance approximation.** Some posterior samplers approximate a conditional likelihood by evaluating it at a denoised point estimate (Chung et al., 2023). This can introduce method-dependent approximation error, but does not imply a universal accuracy disadvantage for generative inference. Appendix C.9 gives the precise distinction.

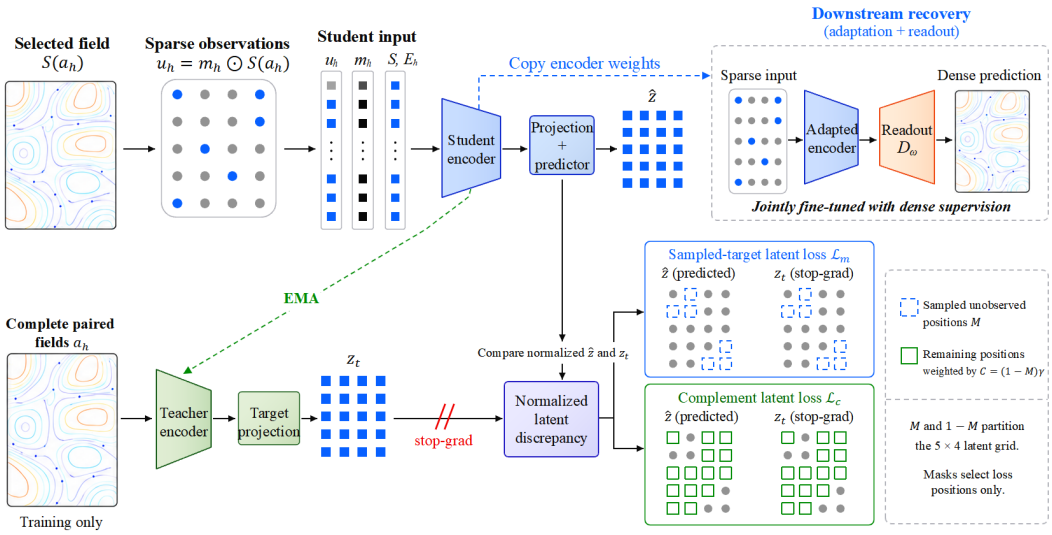


Figure 1: **Overview of JENO.** Predictive pretraining and downstream adaptation for sparse PDE inference.

Deterministic prediction remains meaningful even when sparse observations do not uniquely identify the target. Under squared field error, conditional expectation separates irreducible uncertainty from excess prediction error; the latter depends on the representation, model class, available training data, and optimization, rather than on representation alone (Eq. (20)). Consequently, poor results from operators trained on complete inputs and tested after masking should not be interpreted as evidence that deterministic inference is intrinsically unsuitable. Sparse-input training and observation-aware encodings must also be considered (Sec. 4.2).

Our focus is to use complete training fields to improve deterministic recovery through predictive representation learning. JENO pretrains a sparse-input encoder against full-field latent targets and then fine-tunes it with a task-specific readout, retaining one-pass inference (Sec. 3.1). The analysis in Sec. 3.2 relates the achieved latent objective to alignment and downstream error under explicit assumptions; it does not rank all deterministic and generative methods. The empirical comparison evaluates the resulting prediction accuracy and inference cost, rather than posterior calibration.

3 JENO: Full-Field Latent Prediction

JENO combines predictive representation learning with supervised field recovery. A sparse-input student first predicts latent targets supplied by a complete-field teacher. The pretrained student encoder is then adapted to the desired forward or inverse task. Figure 1 summarizes these two stages; we present the architecture and objective before analyzing their relation to field prediction.

3.1 JENO Architecture

Following Sec. 2.1, let $a_h = (g_h, f_h)$ be a complete PDE instance on a grid Ξ_h with $N = HW$ positions. The selectors S and T identify the observed and predicted fields. With observation mask m_h, the sparse measurement is $u_h = m_h \odot S(a_h)$ and the dense target is $r_h = T(a_h)$. The student and teacher receive

$$x_s = (u_h, m_h, S, \Xi_h), \qquad x_t = (a_h, \mathbf{1}, \Xi_h). \tag{2}$$

The student input records measured values, their locations, and the task direction. In the paired-channel implementation, the unobserved physical channel and its mask are zero; the teacher instead receives both complete fields with all-one masks. Complete fields provide training supervision but are unavailable to the deployed predictor.

The student comprises an encoder E_θ, projection head B_η, and predictor P_ϕ. The teacher supplies a stop-gradient target in the same latent space:

$$\begin{aligned} z_s &= B_\eta(E_\theta(x_s)), & \widehat{z} &= P_\phi(z_s) = J_\Theta(x_s), \\ z_t &= \mathrm{sg}[B_{\bar\eta}(E_{\bar\theta}(x_t))], & J_\Theta &= P_\phi \circ B_\eta \circ E_\theta. \end{aligned} \tag{3}$$

Here $\Theta = (\theta, \eta, \phi)$ and sg stops gradients through the target. The teacher encoder follows the student encoder through $\bar\theta \leftarrow \mu\bar\theta + (1-\mu)\theta$, with momentum $\mu \in (0,1)$. This update concerns the encoder; the target projection is treated separately, as described in Appendix B.1.1.

The main implementation uses an FNO encoder with width 64, six Fourier blocks, and 16×16 retained modes. Each block combines spectral and pointwise convolutions with normalization, a nonlinearity, and a residual connection. The encoder ends with GroupNorm, and a small convolutional head implements the student projection and prediction maps. The supervised FNO features retain the spatial grid, with final latents $\widehat{z}, z_t \in \mathbb{R}^{d_z \times N}$ and $d_z = 32$. We also evaluate CNO and U-Net backbones (Table 3).

We sample square blocks on the grid and intersect their union with the unobserved region to form a binary target mask M. The remaining positions receive weights $C = (1-M)\gamma$, where $1 \le \gamma \le 1+\alpha$ and $\alpha = 2$ in the main configuration. The weights emphasize spatial variation in the complete solution field; their construction and the empty-mask convention are given in Appendix B.3.1. Thus, **the context region is the target-mask complement**, containing both observed locations and unobserved locations outside the sampled targets. The auxiliary masks M and C select loss positions and do not enter either network. Their alignment with grid indices does not imply that a latent feature represents only one physical pixel: the encoder mixes information across locations. At each supervised level j, normalize latent vectors over channels using $\nu_\epsilon(v) = v/\max\{\|v\|_2, \epsilon\}$, with $\epsilon = 10^{-12}$, and define $e_{bi}^{(j)} = \|\nu_\epsilon(\widehat{z}_{bi}^{(j)}) - \nu_\epsilon(z_{t,bi}^{(j)})\|_2^2$. Here b indexes batch members and i indexes spatial positions. The two loss terms use batch-wide spatial reductions:

$$\mathcal{L}_m^{(j)} = \frac{\sum_{b,i} M_{bi} e_{bi}^{(j)}}{\max\{\sum_{b,i} M_{bi}, \epsilon_\ell\}}, \qquad \mathcal{L}_c^{(j)} = \frac{\sum_{b,i} C_{bi} e_{bi}^{(j)}}{\max\{\sum_{b,i} C_{bi}, \epsilon_\ell\}}, \tag{4}$$

where $\epsilon_\ell = 10^{-6}$. There is no additional division by latent dimension, and an empty term contributes zero. For the supervised levels $\mathcal{J}$, including the final prediction level, pretraining minimizes

$$\mathcal{L}_{\mathrm{JENO}} = \sum_{j \in \mathcal{J}} \left(\lambda_m \mathcal{L}_m^{(j)} + \lambda_c \mathcal{L}_c^{(j)} \right), \qquad \lambda_m = 1, \quad \lambda_c = 0.25. \tag{5}$$

The terms jointly supervise the full latent grid, while their separate normalization controls the relative contribution of sampled targets and their complement. The context term is also latent prediction, not a constraint that forces the output to match measured values. **No dense-field reconstruction loss is used during pretraining.**

We copy the pretrained student encoder and fine-tune it jointly with a task-specific readout using dense paired-field supervision (Appendix B.3.1); r_h is the requested output channel. Let $H_0(X) = E_\theta(x_s)$ denote the inherited representation and $H(X) = E_{\theta'}(x_s)$ its fine-tuned counterpart. The deployed prediction is

$$\widehat{r}_h = \widehat{Y}(X) = D_\omega(H(X), X). \tag{6}$$

Only the encoder is transferred from the pretraining student branch; the projection head, predictor, and teacher are not used for inference. Recovery therefore requires one deterministic encoder–readout pass from sparse observations. Readout architectures and training schedules are specified in Appendix B.3.1. The reported main models use the final checkpoints of those schedules, without validation-based checkpoint selection.

3.2 Analysis of the Implemented Latent Objective

We now relate the pretraining objective to full-grid latent alignment and then to the adapted field predictor. The analysis fixes the learned networks and evaluates them on fresh batches under a specified joint law of complete instances and sparse views. On the fixed grid, $Y = r_h$ denotes the physical target. For a grid field V, write $\|V\|_h^2 = N^{-1}\sum_i \|V_i\|_2^2$ and $\mathcal{E}(V) = (\mathbb{E}\|V\|_h^2)^{1/2}$. Single-example expectations refer to a uniformly selected member of the evaluation batch.

Let $\mathcal{B} = ((a_b, X_b))_{b=1}^{B}$ denote a batch and suppress the level index at the final prediction level. The corresponding loss is $\ell_B = \sum_{b,i} w_{bi} e_{bi}$, with

$$w_{bi} = \frac{\lambda_m M_{bi}}{\max\{\sum_{b',k} M_{b'k}, \epsilon_\ell\}} + \frac{\lambda_c C_{bi}}{\max\{\sum_{b',k} C_{b'k}, \epsilon_\ell\}}. \tag{7}$$

Define $L_{\text{final}} = \mathbb{E}\ell_B$ and $L_{\text{all}} = \mathbb{E}\mathcal{L}_{\text{JENO}}$, where expectations include auxiliary target-block sampling. Since other levels contribute nonnegative terms, $L_{\text{all}} \geq L_{\text{final}}$.

Theorem 3.1 (Mask coverage controls normalized latent error)**.** *Suppose the network outputs are independent of the auxiliary target-block draw conditional on $\mathcal{B}$. If, for some $\kappa > 0$, the effective weights satisfy $\mathbb{E}[w_{bi} \mid \mathcal{B}] \geq \kappa/(BN)$ almost surely for every b, i, then*

$$\mathbb{E}\|\nu_\epsilon(\widehat{z}) - \nu_\epsilon(z_t)\|_h^2 \leq \frac{L_{\text{final}}}{\kappa} \leq \frac{L_{\text{all}}}{\kappa}. \tag{8}$$

The main complement weighting satisfies this condition for every block draw: target positions have weight at least $\lambda_m/(BN)$, and other positions have weight at least $\lambda_c/((1+\alpha)BN)$. Hence $\kappa = \min\{\lambda_m, \lambda_c/(1+\alpha)\} = 1/12$. The proof and the observed-position context variant appear in Appendix A.2. This is a supervision-coverage guarantee, not a prediction of which weighting yields the best downstream model.

Normalization and target-dependent weighting also affect what the objective predicts. For the uniformly selected batch member, let $\beta_i = Bw_{bi}$. Where $\mathbb{E}[\beta_i \mid X] > 0$, the pointwise optimum over unit-ball-valued outputs is $\mu_i(X) = \mathbb{E}[\beta_i \nu_\epsilon(z_{t,i}) \mid X]/\mathbb{E}[\beta_i \mid X]$; over unit-norm outputs, it is the direction of μ_i when nonzero. Thus the objective does not generally target $\mathbb{E}[z_t \mid X]$. Appendix A.3 establishes this statement and distinguishes latent prediction from the physical uncertainty in Eq. (20).

Latent alignment is evaluated after the pretraining predictor, whereas the readout receives encoder features. To account for this interface, choose a fixed linear channel map A into the encoder-feature space, with $K_A = \|A\|_2$. This is a comparison map used in the analysis, not an inference module. Put $r_{s,i} = \max\{\|\widehat{z}_i\|_2, \epsilon\}$ and $r_{t,i} = \max\{\|z_{t,i}\|_2, \epsilon\}$, and define

$$\begin{aligned} \delta_{\text{int}} &= \mathcal{E}(H_0 - A\widehat{z}), & \delta_{\text{FT}} &= \mathcal{E}(H - H_0), \\ \delta_t &= \mathcal{E}(D_\omega(Az_t, X) - Y), & \delta_{\text{scale}} &= \Big(\mathbb{E}\frac{1}{N}\sum_i (r_{s,i} - r_{t,i})^2\Big)^{1/2}. \end{aligned} \tag{9}$$

The teacher residual uses the same final readout as the deployed model. The other terms measure the discrepancy between predicted latents and inherited features, the change during fine-tuning, and the latent-scale difference left uncontrolled by normalization.

Theorem 3.2 (Transfer to the implemented readout)**.** *Under the assumptions of Theorem 3.1, suppose $r_{s,i} \leq R$ almost surely for every position, the final readout D_ω is L_D-Lipschitz in its representation input in $\|\cdot\|_h$ uniformly over the relevant X, and all residuals in Eq.* (9) *are finite. Then*

$$\mathcal{E}(\widehat{Y} - Y) \leq \delta_t + L_D \left[\delta_{\text{FT}} + \delta_{\text{int}} + K_A \left(\delta_{\text{scale}} + R\sqrt{\frac{L_{\text{final}}}{\kappa}}\right)\right]. \tag{10}$$

The bound shows how small latent loss can contribute to accurate field recovery when teacher decoding, latent scale, representation transfer, and fine-tuning residuals are controlled. These residuals are not assumed to vanish. Appendix A provides the proof, finite weight-based constants for the stated FNO architecture and readouts, and extensions to mixed observation protocols and independent checkpoint evaluation. All quantities in the bound refer to the same observation law. The analysis relates the objective to prediction error under explicit conditions; performance advantages over direct supervision or alternative objectives remain empirical questions.

4 Experiments

We evaluate JENO on sparse-to-dense PDE inference, with inverse recovery as the primary task and forward prediction as a complementary test. We first compare prediction error and inference time, then examine backbone and readout choices, observation density, measurement noise, and representation diagnostics.

Table 1: **Forward and inverse errors on Partial-OBS (%).** Relative L_2 is used except for Darcy inverse (binary error rate). Bold/underline mark the lowest/second-lowest displayed errors, not significance. †NS (BCs) protocols differ; these columns are unranked. Source, test-set, and normalization differences are specified in Appendix B.1.6.

Method	Darcy Flow		Poisson		Helmholtz		NS		NS (BCs)†	
	Fwd	Inv	Fwd	Inv	Fwd	Inv	Fwd	Inv	Fwd	Inv
Deterministic neural PDE baselines (trained with full observations)										
FNO Li et al. (2021a)	28.2	49.3	100.9	232.7	98.2	218.2	101.4	96.0	82.8	69.6
PINO Li et al. (2021b)	35.2	49.2	107.1	231.9	106.5	216.9	101.4	96.0	81.1	69.5
DeepONet Lu et al. (2021)	38.3	41.1	155.5	105.8	123.1	132.8	103.2	97.2	97.7	91.9
PINN Raissi et al. (2019)	48.8	59.7	128.1	130.0	142.3	160.0	142.7	146.8	100.1	105.5
Deterministic neural PDE baselines (mixed-density mask-aware training)										
FNO	10.83	13.10	26.09	49.31	27.20	49.17	19.10	31.66	26.31	21.94
DeepONet	11.03	11.19	16.65	36.03	17.03	41.87	39.59	50.28	34.35	27.76
CNO	12.78	5.71	35.33	45.09	33.99	43.43	26.42	27.55	33.74	28.25
Generative / sampling-based sparse-inference methods										
DiffusionPDE Huang et al. (2024)	6.07	7.87	4.88	21.10	12.64	19.07	3.78	9.63	9.69	4.18
FunDPS Yao et al. (2025)	<u>2.88</u>	6.78	2.04	24.04	2.20	20.07	3.99	9.87	5.91	4.31
FunDPS+ Yao et al. (2025)	**2.49**	<u>5.18</u>	<u>1.99</u>	20.47	<u>2.13</u>	17.16	3.32	8.48	4.90	4.08
PRISMA-50 Sawhney et al. (2025)	3.11	6.80	—	—	—	—	—	—	—	—
Equi-FunDPS Tolooshams et al. (2025)	—	—	—	—	**2.12**	15.91	<u>3.06</u>	7.84	—	—
PISD Gallon et al. (2026)	—	—	3.08	13.81	3.47	12.76	—	—	—	—
DDIS Lin et al. (2026)	—	—	—	<u>12.32</u>	—	<u>12.20</u>	—	<u>7.81</u>	—	—
JENO (ours)	4.31	**2.35**	**1.80**	**9.72**	2.16	**10.10**	**2.78**	**7.36**	4.10	3.58

4.1 Experimental Setup

Data and tasks. We use the normalized Partial-OBS datasets released with FunDPS and derived from DiffusionPDE (Huang et al., 2024; Yao et al., 2025): Darcy flow, Poisson, Helmholtz, Navier–Stokes (NS), and Navier–Stokes with boundary conditions (NS (BCs)). The fields are represented on 128×128 grids. Forward prediction recovers f from sparse measurements of g, while inverse recovery predicts g from sparse measurements of f, following Sec. 2.1. The main JENO evaluation uses $1{,}000$ held-out instances per PDE and a nominal 3% observation rate; control evaluation sizes are specified in Appendix B.1.6. Dataset sizes, task definitions, and sensor-count conventions are given in Appendix B.1.

Training and evaluation. Paired complete fields are available during training, whereas deployment uses only the sparse view. JENO transfers the pretrained student encoder and jointly fine-tunes it with a task-specific readout, separately for the forward and inverse tasks. The main runs use training seed 42 and the fixed schedules in Table 11. We evaluate the **final checkpoints**, without validation-based checkpoint selection. Appendix B.3.1 distinguishes these models from the separate diagnostic checkpoints and specifies the inherited encoder, observation sampler, and fine-tuning budget.

Baselines and metrics. Table 1 separates published full-observation-trained deterministic baselines, our mask-aware FNO, DeepONet, and CNO controls, and literature-reported generative methods. Baseline sources and configurations are listed in Appendix B.1.6. The main-table DeepONet uses a convolutional grid-input branch; the non-convolutional sensor-coordinate variant is evaluated separately in Table 3. Errors are percentages: Relative L_2 for all endpoints except Darcy inverse, which uses binary error rate (Appendix B.1.7). Entries retain their source protocols, including different test sizes and error-normalization spaces (Appendix B.1.6); these are not common-mask evaluations.

4.2 Main Results and Inference Time

Inverse recovery. JENO achieves the lowest reported inverse error among the compared methods on Darcy, Poisson, Helmholtz, and NS, with errors of 2.35%, 9.72%, 10.10%, and 7.36%, respectively (Table 1). These comparisons include both the mask-aware deterministic controls and the reported

generative methods. Across these eight forward/inverse endpoints, JENO has lower displayed errors than all three mask-aware controls; Appendix B.1.6 specifies their evaluation differences.

Forward prediction and comparison scope. JENO also has the lowest reported forward error on Poisson (1.80%) and NS (2.78%), while its Helmholtz error of 2.16% is close to the lowest reported value of 2.12%. We retain the NS (BCs) results but do not rank them because the observation protocols are not matched. The substantially lower errors of the mask-aware FNO and DeepONet controls relative to the published full-input-trained rows also show why the latter alone are insufficient baselines for sparse-input inference. These comparisons evaluate the complete JENO training and adaptation procedure, rather than isolating the contribution of latent pretraining. Appendix C.9 discusses the backbone, readout, and component evidence.

Inference time. On the timed Poisson and NS tasks, JENO produces one dense prediction in 5.54 and 5.55 ms, respectively, on an NVIDIA H200 (Table 2). The evaluated DiffusionPDE and FunDPS samplers require approximately 5.5×10^4–8.2×10^4 times longer on the same GPU. JENO retains a low-latency, one-pass prediction path. The comparison concerns single-sample inference, not training cost or the cost of obtaining a calibrated posterior.

Table 2: Mean inference time over 200 single-sample runs on H200; speed relative to JENO (Appendix B.3.1).

Method	Runtime (s)	Speed
Poisson		
JENO	5.54×10^{-3}	1.00×
DiffusionPDE	4.35×10^2	$7.85 \times 10^4\times$ slower
FunDPS	3.07×10^2	$5.54 \times 10^4\times$ slower
Navier–Stokes		
JENO	5.55×10^{-3}	1.00×
DiffusionPDE	4.54×10^2	$8.18 \times 10^4\times$ slower
FunDPS	3.43×10^2	$6.18 \times 10^4\times$ slower

4.3 Backbone and Readout Dependence

At matched 3% downstream training and test observation densities, JENO is effective with FNO, CNO, and U-Net backbones on Poisson forward and NS inverse (Table 3). FNO gives the lowest Poisson error among the JENO variants (1.80%), whereas CNO and U-Net give lower NS inverse errors (6.29% and 6.33%) than FNO (7.36%). All three outperform the evaluated DeepONet with an MLP sensor-value/location branch and a query-coordinate trunk. The comparison supports applicability beyond the FNO backbone in these structured-grid tasks; it is not a controlled comparison of pretraining objectives (Appendix C.1).

Readout swaps provide a separate test of downstream architectural sensitivity. Starting from a shared task-specific base checkpoint and using equal 20-epoch fine-tuning budgets, the five readouts yield Poisson forward errors of 1.74–1.92% and NS inverse errors of 5.24–6.24% (Table 21). The preferred readout differs between the two tasks. These runs use the separate initialization and fixed-count masks described in Appendix C.8; they are not alternative readouts evaluated on an unchanged main-benchmark checkpoint.

Table 3: Matched 3% downstream train/test density across backbones. Mean Relative L_2 error (%); lowest values are bold.

Method	Poisson forward	NS inverse
JENO-FNO	**1.80**	7.36
JENO-CNO	2.04	**6.29**
JENO-U-Net	2.45	6.33
DeepONet (MLP sensor branch)	6.29	28.92

4.4 Sensitivity to Observations and Design Choices

Observation density. At fixed 128×128 resolution, we evaluate the 3%-fine-tuned models from 1% to 100% observations without retraining (Table 17). Increasing density from 1% to 20% reduces Poisson forward error from 5.16% to 0.50% and NS inverse error from 11.12% to 3.81%. The trend is not monotonic: both errors increase between 20% and 50%. The sweep measures transfer across sensor densities, not spatial resolutions or performance relative to a separately trained full-observation model. Appendix C.2 additionally reports FNO and convolutional DeepONet trained at 2% and tested at 2% and 5%.

Measurement noise. Using separate fixed checkpoints, we perturb observed values while holding sensor locations fixed across noise levels (Appendix C.4). At 1% noise, inverse-error amplification

is $3.186\times$ for Poisson, $3.589\times$ for Helmholtz, and $2.299\times$ for NS, compared with $1.010\times$, $1.009\times$, and $1.013\times$ in the corresponding forward tasks (Table 18). Thus, the strong clean-input results do not imply uniform robustness to measurement noise. This experiment characterizes the sensitivity of the trained predictors, not an architecture-independent condition number of the underlying PDEs.

Context weight and latent dimension. Table 15 reports independent sweeps over $\lambda_c \in \{0, 0.1, 0.25, 1.0\}$ and $d_z \in \{16, 32, 64\}$ on Poisson and NS. The effects depend on task and density, with no single setting best across all endpoints. This diagnostic initializes the downstream model from the EMA encoder, unlike the student-initialized main benchmark, and includes full recovery from sparse observations of both fields (Appendix B.4). Varying λ_c changes the spatial allocation of latent supervision; it does not remove full-field predictive pretraining or compare it against direct supervised training.

4.5 Representation Diagnostics

We evaluate representation alignment using two complementary diagnostics. In the shared-teacher comparison, JENO, FNO, and convolutional DeepONet are evaluated against the same complete-field teacher representation (Appendix B.2). JENO exhibits consistently lower latent mismatch and higher CKA across Poisson and Navier–Stokes over the tested observation densities. The comparison for FNO and DeepONet includes a full-to-partial input shift, since these controls are trained on complete inputs and evaluated after masking. We additionally compare partial-view and complete-view representations within sparse-trained models (Appendix C.6). Mismatch decreases overall, with non-monotonic intermediate changes. At full observation it is zero by construction because the inputs coincide; this endpoint is not independent evidence of pretraining success, and CKA need not increase monotonically.

5 Conclusion and Limitations

Conclusion. In this work, we introduced JENO, a joint-embedding neural operator framework for full-field PDE inference under partial observations. During latent pretraining, JENO aligns a student representation computed from sparse measurements and masks with a stop-gradient teacher representation of the complete PDE state. It then transfers the student encoder and jointly fine-tunes it with a task-specific readout using dense-field supervision. Inference uses only the adapted encoder and readout. The theory conditionally connects the achieved masked normalized loss to full-grid alignment and downstream error, with explicit teacher-quality, latent-scale, representation-transfer, and fine-tuning residuals.

Empirically, JENO shows that strong sparse-observation PDE inference does not require iterative probabilistic sampling. Across the first four Partial-OBS settings, JENO has lower reported errors than the mask-aware FNO, CNO, and DeepONet controls on all eight forward/inverse endpoints under the respective evaluation protocols. It also achieves competitive or superior accuracy compared with recent diffusion-based sparse-inference methods, while retaining deterministic one-pass inference. The representation diagnostics and sparse-trained controls support the evaluated protocols, not an architecture-independent ranking. Overall, these results suggest that joint-embedding predictive learning provides a strong non-probabilistic alternative for sparse PDE inference and a promising direction for representation learning in scientific machine learning.

Limitations. Despite these encouraging results, several limitations remain. First, our experiments are conducted on the standard Partial-OBS benchmark with structured 128×128 grids and controlled point-observation protocols. Although these benchmarks cover multiple PDE families, they do not fully capture the complexity of irregular geometries, adaptive meshes, multi-resolution discretizations, or real-world sensor networks. Second, observations are noiseless with known masks in the main benchmark; the added noise diagnostic establishes only task-dependent sensitivity. JENO needs paired complete training fields and returns point estimates. Alternative liftings, cross-resolution/3D tests, and irregular, biased, or asynchronous sensing remain open.

STATEMENT ON AI USAGE

Large language models and other AI tools were used to assist with language editing and polishing, retrieval and discovery of relevant literature, and research ideation and execution, including discussion of experimental design, analysis, and implementation strategies. All AI-assisted outputs were critically reviewed and verified by the authors, and the authors take full responsibility for the scientific content, methodology, results, citations, and conclusions presented in this work.

REFERENCES

Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. SIAM, 2016. doi: 10.1137/1.9781611974546.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pp. 1298–1312. PMLR, 2022. URL https://proceedings.mlr.press/v162/baevski22a.html.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2 edition, 2010. ISBN 978-0-8218-4974-3. doi: 10.1090/gsm/019.

Davide Gallon, Philippe von Wurstemberger, Patrick Cheridito, and Arnulf Jentzen. Physics-informed diffusion models in spectral space. *arXiv preprint arXiv:2602.09708*, 2026.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

Jingren Hou, Hong Wang, Pengyu Xu, Chang Gao, Huafeng Liu, and Liping Jing. Learning neural operators from partial observations via latent autoregressive modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):390–398, 2026. doi: 10.1609/aaai.v40i1.37001.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *Advances in Neural Information Processing Systems*, 2024.

Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*, volume 160 of *Applied Mathematical Sciences*. Springer, 2005. doi: 10.1007/b138659.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23593–23606, 2022.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021b.

Thomas Y. L. Lin, Jiachen Yao, Lufang Chiang, Julius Berner, and Anima Anandkumar. Decoupled diffusion sampling for inverse problems on function spaces. *arXiv preprint arXiv:2601.23280*, 2026.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Jian Luo, Jie Wang, Hong Wang, Huanshuo Dong, Zijie Geng, Hanzhu Chen, and Yufei Kuang. Neural krylov iteration for accelerating linear system solving. In *Advances in Neural Information Processing Systems*, volume 37, pp. 128636–128667, 2024.

Qitan Lv, Hong Wang, Zhongkai Hao, Wen Wu, Xuenan Xu, Bowen Zhou, Feng Wu, and Chao Zhang. AOT-POT: Adaptive operator transformation for large-scale PDE pre-training. *arXiv preprint arXiv:2605.15793*, 2026.

Alfio Quarteroni. *Numerical Models for Differential Problems*. MS&A: Modeling, Simulation and Applications. Springer Milan, Milan, 2009. ISBN 978-88-470-1070-3. doi: 10.1007/978-88-470-1071-0.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=j3oQF9coJd.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper/2023/hash/f3c1951b34f7f55ffaecada7fde6bd5a-Abstract-Conference.html.

Medha Sawhney, Abhilash Neog, Mridul Khurana, and Anuj Karpatne. Beyond loss guidance: Using PDE residuals as spectral attention in diffusion neural operators. *arXiv preprint arXiv:2512.01370*, 2025.

Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling PDEs on general geometries. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70581–70611, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/df54302388bbc145aacaa1a54a4a5933-Abstract-Conference.html.

Andrew M. Stuart. Inverse problems: a bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005. doi: 10.1137/1.9780898717921.

Bahareh Tolooshams, Aditi Chandrashekar, Rayhan Zirvi, Abbas Mammadov, Jiachen Yao, Chuwei Wang, and Anima Anandkumar. EquiReg: Equivariance regularized diffusion for inverse problems. *arXiv preprint arXiv:2505.22973*, 2025.

Hong Wang, Zhongkai Hao, Jie Wang, Zijie Geng, Zhen Wang, Bin Li, and Feng Wu. Accelerating data generation for neural operators via krylov subspace recycling. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided diffusion sampling on function spaces with applications to PDEs. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2505.17004.

A PROOFS FOR THE IMPLEMENTED LATENT OBJECTIVE

This appendix proves Theorems 3.1 and 3.2 for fixed student–teacher networks, and develops the accompanying statements on normalization, network constants, observation protocols, and finite-sample evaluation. Unless otherwise stated, the analysis concerns the final prediction level. All population expectations are conditional on the learned parameters and concern fresh evaluation batches under the specified joint law of complete fields and sparse observations. Other supervised levels contribute nonnegative terms, so the final-level loss may be bounded by the total pretraining objective. Fixing a checkpoint for analysis does not make the data used to train it independent evaluation data.

A.1 EXACT NORMALIZATION, REDUCTION, AND PROBABILITY SPACE

Let $\mathcal{B} = ((a_b, X_b))_{b=1}^{B}$ be a random batch of complete PDE objects and sparse views. Members may share a randomly selected observation rate or task, as in the minibatch sampler. Let A denote the auxiliary target-block draws conditional on $\mathcal{B}$. Network outputs $\hat{z}_b = \hat{z}(X_b)$ and $z_{t,b} = z_t(a_b)$ are deterministic functions of the examples at a fixed checkpoint, in evaluation mode. In particular, A only selects loss positions; it is not an extra input to either network. Define

$$\nu_\epsilon(v) = \frac{v}{r_\epsilon(v)}, \qquad r_\epsilon(v) = \max\{\|v\|_2, \epsilon\}, \qquad \|V\|_h^2 = \frac{1}{N}\sum_{i=1}^{N} \|V_i\|_2^2. \tag{11}$$

The channel normalization uses $\epsilon = 10^{-12}$; the spatial reduction uses the separate floor $\epsilon_\ell = 10^{-6}$. Thus $\|\nu_\epsilon(v)\|_2 \le 1$, including $v = 0$, and $v = r_\epsilon(v)\nu_\epsilon(v)$ exactly. The floor must not be replaced by an assumption that every normalized vector is a unit vector.

Put $f_{bi} = \nu_\epsilon(\hat{z}_{bi})$, $t_{bi} = \nu_\epsilon(z_{t,bi})$, and $e_{bi} = \|f_{bi} - t_{bi}\|_2^2$. For nonnegative masks/weights, the implemented loss is

$$w_{bi} = \frac{\lambda_m M_{bi}}{\max\{\sum_{b',j} M_{b'j}, \epsilon_\ell\}} + \frac{\lambda_c C_{bi}}{\max\{\sum_{b',j} C_{b'j}, \epsilon_\ell\}}, \tag{12}$$

$$\ell_B = \sum_{b,i} w_{bi} e_{bi}, \qquad \mathcal{L}_{\text{final}} = \mathbb{E}_{\mathcal{B},A}\ell_B. \tag{13}$$

Every $e_{bi} \le 4$, and $\sum_{b,i} w_{bi} \le \lambda_m + \lambda_c$, so $0 \le \ell_B \le 4(\lambda_m + \lambda_c)$. The weights contain the random batch denominators. No proof replaces an expectation of a ratio by a ratio of expectations. For statements about one example, choose b uniformly from the batch independently of everything else; this defines its marginal law even without within-batch independence. The physical target Y is the desired field for the chosen task, assumed square-integrable whenever a field-risk statement is used.

A.2 PROOF OF THEOREM 3.1 AND EXPLICIT COVERAGE CONSTANTS

Proof of Theorem 3.1. Conditional on $\mathcal{B}$, each error e_{bi} is fixed. The assumed lower bound on each expected weight therefore gives

$$\mathbb{E}_A[\ell_B \mid \mathcal{B}] = \sum_{b,i} e_{bi}\mathbb{E}_A[w_{bi} \mid \mathcal{B}] \ge \frac{\kappa}{BN}\sum_{b,i} e_{bi}. \tag{14}$$

Taking expectations over $\mathcal{B}$, the last batch average equals the expected grid error for a uniformly selected member. Dividing by $\kappa > 0$ proves the first inequality. Summation over levels adds nonnegative losses, proving the second. This argument does not require independent grid positions, a fixed sensor count, an approximation property of FNO, or a global optimizer. □

Observed-position context and sampled missing blocks. Let $O_{bi} \in \{0, 1\}$ be the observed union mask, fixed with $\mathcal{B}$, and set $C = O$. In each example draw $k \ge 1$ square blocks of integer side $1 \le s \le \min(H, W)$, independently and uniformly over valid top-left corners. Form their union U, intersect with $1 - O$, and, if that intersection is empty for an example, use its entire unobserved

region. This is the target-mask helper in Appendix B.3.1. For pixel (r, c), indexed from one, the numbers of covering vertical and horizontal origins are

$$\begin{aligned} n_H(r) &= \min(r, H - s + 1) - \max(1, r - s + 1) + 1, \\ n_W(c) &= \min(c, W - s + 1) - \max(1, c - s + 1) + 1. \end{aligned} \tag{15}$$

These integers are at least one. Writing $T = (H - s + 1)(W - s + 1)$, the union inclusion probability is

$$q_{rc} = 1 - \left(1 - \frac{n_H(r) n_W(c)}{T}\right)^k \geq q_{\min} = 1 - (1 - 1/T)^k > 0. \tag{16}$$

For an unobserved position, intersection preserves inclusion whenever a block covers it, and the fallback can only increase its inclusion probability. Since $\sum M \leq BN$ and a nonzero binary numerator has denominator at least one and at most BN, $\mathbb{E}_A[w_{bi} \mid \mathcal{B}] \geq \lambda_m q_{\min}/(BN)$ there. At any observed position, $\sum O \leq BN$ gives $w_{bi} \geq \lambda_c/(BN)$. Consequently

$$\kappa_{\text{obs}} = \min\{\lambda_c, \lambda_m q_{\min}\} \tag{17}$$

satisfies Theorem 3.1 when both coefficients are positive. If an example is fully observed its target term is zero but context covers every position; if fully unobserved, the target sampler covers each position with positive probability. An empty term contributes zero because its numerator is zero. The denominator bound BN remains valid under the fallback: a bound of Bks^2 would in general be invalid because the fallback can select more than ks^2 positions.

Gradient-weighted target complement. The main pretraining objective uses $C_{bi} = (1 - M_{bi})\gamma_{bi}$, where $1 \leq \gamma_{bi} \leq 1 + \alpha$. The weighting function is defined in Appendix B.3.1; its normalized gradient factor lies in $[0, 1]$ and its dilation factor lies in $[1/2, 1]$. These bounds hold even though the weights depend on the complete field and the auxiliary target-block draw. At $M_{bi} = 1$, the target term gives $w_{bi} \geq \lambda_m/(BN)$. At $M_{bi} = 0$, one has $C_{bi} \geq 1$ and $\sum_{b',k} C_{b'k} \leq (1 + \alpha)BN$, giving $w_{bi} \geq \lambda_c/((1 + \alpha)BN)$. Therefore, for every realized mask,

$$w_{bi} \geq \frac{\kappa_{\text{comp}}}{BN}, \qquad \kappa_{\text{comp}} = \min\left\{\lambda_m, \frac{\lambda_c}{1 + \alpha}\right\}. \tag{18}$$

No independence between the spatial weights and teacher targets is required. With $\lambda_m = 1$, $\lambda_c = 0.25$, and $\alpha = 2$, the main configuration gives $\kappa_{\text{comp}} = 1/12$. The context term covers every position outside the sampled targets, including unobserved positions. If one loss region is empty, its numerator is zero and the denominator floor makes its contribution zero; the other region still covers the grid. The observed-position context variant considered above instead uses its own coverage constant κ_{obs}.

A.3 The conditional target of normalized, weighted prediction

Proposition A.1 (Weighted projection and directional optimum)**.** *For the uniformly selected batch member, let $\beta_i = Bw_{bi}$, $t_i = \nu_\epsilon(z_{t,i})$, and $a_i(X) = \mathbb{E}[\beta_i \mid X]$. Where $a_i > 0$, define $\mu_i(X) = \mathbb{E}[\beta_i t_i \mid X]/a_i(X)$. For any measurable candidate $f_i(X)$, the conditional risk satisfies*

$$\mathbb{E}[\beta_i \|f_i - t_i\|_2^2 \mid X] = a_i \|f_i - \mu_i\|_2^2 + \mathbb{E}[\beta_i \|t_i - \mu_i\|_2^2 \mid X]. \tag{19}$$

One has $\|\mu_i\|_2 \leq 1$. Thus the unconstrained unit-ball optimum is μ_i. If candidates have unit norm and $\mu_i \neq 0$, the optimal direction is $v_i = \mu_i/\|\mu_i\|_2$, and their excess conditional risk is $a_i\|\mu_i\|_2\|f_i - v_i\|_2^2$. If $\mu_i = 0$, all unit directions have the same risk. Where $a_i = 0$, the conditional loss is zero and gives no constraint.

Proof. Expand $f_i - t_i = (f_i - \mu_i) + (\mu_i - t_i)$. The cross term vanishes because $\mathbb{E}[\beta_i(t_i - \mu_i) \mid X] = 0$. The weighted triangle inequality gives $\|\mu_i\|_2 \leq \mathbb{E}[\beta_i \|t_i\|_2 \mid X]/a_i \leq 1$. On the unit sphere, the only candidate-dependent term in the expanded risk is $-2a_i\langle f_i, \mu_i\rangle$, minimized by alignment with μ_i. Subtracting the minimum yields $2a_i\|\mu_i\|_2(1 - \langle f_i, v_i\rangle) = a_i\|\mu_i\|_2\|f_i - v_i\|_2^2$. Nonnegative β_i has conditional mean zero only when it is zero almost surely, establishing the last case. □

The population final-level objective is $\mathbb{E}\sum_i \beta_i \|f_i(X) - t_i\|_2^2$ by uniform selection of the batch member. Thus the proposition preserves the implemented batch reduction, including dependence

of its denominator on other examples and on the target. The statement about the unit ball is a pointwise optimum over unrestricted measurable normalized outputs, not a claim that a particular neural network attains it. The implemented normalization can output any vector in that ball through raw vectors below its epsilon floor, whereas ordinary nonzero activations above the floor lie on the sphere. No optimization trajectory towards the ball optimum is asserted. In particular, even with target-independent weights, the unit-direction target involves $\mathbb{E}[\nu_\epsilon(z_t) \mid X]$, not generally the normalization of $\mathbb{E}[z_t \mid X]$.

Physical ambiguity and excess prediction risk. Let Y be the square-integrable physical target for the chosen task, and let $Y^\star(X) = \mathbb{E}[Y \mid X]$. For any square-integrable field predictor $\widehat{Y}(X)$, conditional expectation in the finite-grid Hilbert space gives

$$\mathbb{E}\big\|\widehat{Y}(X) - Y\big\|_h^2 = \underbrace{\mathbb{E}\big\|Y - Y^\star(X)\big\|_h^2}_{V_Y} + \mathbb{E}\big\|\widehat{Y}(X) - Y^\star(X)\big\|_h^2. \tag{20}$$

To verify the identity, expand $\widehat{Y} - Y = (\widehat{Y} - Y^\star) + (Y^\star - Y)$. The cross term vanishes because $\widehat{Y} - Y^\star$ is measurable with respect to X and $\mathbb{E}[Y^\star - Y \mid X] = 0$.

The term V_Y is the irreducible squared prediction risk associated with the available observations and the physical target. The second term is excess prediction risk. It can reflect observation encoding, model capacity, finite training data, and optimization; the identity does not isolate a representation-specific contribution. It is a reference decomposition under squared field error, not a claim that the implemented training procedure attains the conditional mean.

Physical ambiguity must also be distinguished from teacher-dependent latent uncertainty. A collapsed teacher can have zero conditional latent variance while $V_Y > 0$. Moreover, the normalized, target-weighted latent objective has the conditional target characterized in Proposition A.1, rather than the unweighted conditional mean of the raw teacher latent. The field-risk decomposition therefore neither replaces that proposition nor guarantees accurate physical recovery from latent alignment alone.

A.4 Proof of Theorem 3.2

For random grid fields write $\mathcal{E}(V) = (\mathbb{E}\|V\|_h^2)^{1/2}$. Let $H_0(X)$ be the pretrained student encoder representation and $H(X)$ its fine-tuned representation, supplied to the final task readout D_ω. Choose any fixed linear channel map A from the prediction-latent space to the readout representation space, applied at each grid position, and let $K_A = \|\mathsf{A}\|_2$. This analysis map defines a comparison in the readout space; the deployed prediction is $D_\omega(H, X)$. With $r_{s,i} = r_\epsilon(\hat{z}_i)$ and $r_{t,i} = r_\epsilon(z_{t,i})$, define

$$\delta_{\mathrm{scale}} = \left(\mathbb{E}\frac{1}{N}\sum_i (r_{s,i} - r_{t,i})^2\right)^{1/2}, \qquad \delta_{\mathrm{FT}} = \mathcal{E}(H - H_0), \tag{21}$$

$$\delta_{\mathrm{int}} = \mathcal{E}(H_0 - \mathsf{A}\hat{z}), \qquad \delta_t = \mathcal{E}(D_\omega(\mathsf{A}z_t, X) - Y). \tag{22}$$

The teacher residual uses the same final readout as the deployed prediction. The result holds for every fixed A and does not assume that it reconstructs H_0 exactly. If the readout coordinates already coincide with the predicted latent and $H_0 = \hat{z}$, the identity map gives $K_A = 1$ and $\delta_{\mathrm{int}} = 0$. Assume $r_{s,i} \le R$ almost surely, finite residuals, and

$$\|D_\omega(V, X) - D_\omega(V', X)\|_h \le L_D\|V - V'\|_h \tag{23}$$

uniformly over the relevant inputs. Each grid norm sums over its own channels and averages over the same N positions.

Proof of Theorem 3.2. Equation (11) gives, positionwise,

$$\hat{z}_i - z_{t,i} = r_{s,i}(f_i - t_i) + (r_{s,i} - r_{t,i})t_i. \tag{24}$$

Since $\|t_i\|_2 \le 1$, Minkowski's inequality and Theorem 3.1 imply

$$\mathcal{E}(\hat{z} - z_t) \le R\big(\mathbb{E}\|f - t\|_h^2\big)^{1/2} + \delta_{\mathrm{scale}} \le R\sqrt{\mathcal{L}_{\mathrm{final}}/\kappa} + \delta_{\mathrm{scale}}. \tag{25}$$

The exact decomposition $H - \mathsf{A}z_t = (H - H_0) + (H_0 - \mathsf{A}\hat{z}) + \mathsf{A}(\hat{z} - z_t)$ then gives

$$\begin{aligned}\mathcal{E}(D_\omega(H, X) - Y) &\le \delta_t + L_D \mathcal{E}(H - \mathsf{A}z_t) \\ &\le \delta_t + L_D \left[\delta_{\mathrm{FT}} + \delta_{\mathrm{int}} + K_A \left\{\delta_{\mathrm{scale}} + R\sqrt{\mathcal{L}_{\mathrm{final}}/\kappa}\right\}\right]. \end{aligned} \tag{26}$$

This proves the result for the actual fine-tuned encoder–readout prediction. A small pretraining loss controls the alignment term; the two representation residuals explicitly retain the effects of representation transfer and fine-tuning. □

The encoder, projection, and predictor may be composed into a single student network for implementation and notation. The residual definitions depend on the functions and features, not on how modules are packaged. In particular, A can be any chosen linear comparison map, with its residual evaluated on the specified observation law; no trained inverse or exact factorization is assumed. A vacuous map can give a loose bound. A useful numerical certificate would require evaluating all residuals, which is separate from the reported accuracy experiments.

A.5 Finite constants for the supplied FNO and readout

The constants in Theorem 3.2 can be bounded from a finite checkpoint without an approximation theorem for FNO. Work in evaluation mode, with finite real weights. The supplied FNO encoder ends with one-group GroupNorm on $q = d_e N$ entries. Before its learned affine parameters γ, β, any normalized entry obeys

$$\frac{|v_j - \bar{v}|}{\sqrt{q^{-1}\sum_l (v_l - \bar{v})^2 + \epsilon_g}} \le \sqrt{q}. \tag{27}$$

Consequently its output has entrywise absolute value at most $a_0 = \|\gamma\|_\infty \sqrt{q} + \|\beta\|_\infty$. Regard each subsequent convolution as an affine map $v \mapsto A_j v + b_j$ on the fixed grid, with its actual padding. Starting from a_0, propagate $a_j = \|A_j\|_\infty a_{j-1} + \|b_j\|_\infty$, where the induced infinity norm is the maximum absolute row sum. GELU, $g(x) = x\Phi(x)$, satisfies $|g(x)| \le |x|$, so it does not increase this entrywise bound. If the final predictor has d output channels and final bound a_J, then

$$R = \max\{\sqrt{d}\, a_J, \epsilon\} \tag{28}$$

is a valid uniform bound for the final predicted latent. This argument uses the actual final normalization and heads; it makes no smoothness assumption about a Bayes map. The same reasoning bounds the full-field teacher path when needed. It is specific to the supplied architecture and is not asserted for arbitrary networks lacking this normalization.

For the standard convolutional field head with linear convolution maps W_1, W_2, W_3 and two exact GELU activations, a valid constant is

$$L_D = (5/4)^2 \|W_3\|_2 \|W_2\|_2 \|W_1\|_2. \tag{29}$$

Indeed GELU satisfies $|g'(x)| = |\Phi(x) + x\phi(x)| \le 1 + 1/\sqrt{2\pi e} < 5/4$. Biases cancel in differences. The grid norm rescales both Euclidean norms by $N^{-1/2}$, preserving these constants.

The two-layer FNO readout has residual spectral blocks, a final GroupNorm, and a two-convolution GELU projection. GroupNorm with variance floor $\epsilon_g > 0$ and affine scale γ has Lipschitz constant at most $L_G = \|\gamma\|_\infty / \sqrt{\epsilon_g}$: centering is an orthogonal projection, and the Jacobian of $v/\sqrt{\|v\|_2^2/q + \epsilon_g}$ has spectral norm at most $1/\sqrt{\epsilon_g}$. For block j with spectral map S_j and pointwise map W_j, the residual form $v + \mathrm{GELU}(\mathrm{GN}((S_j + W_j)v + b_j))$ therefore has constant at most $1 + (5/4)L_{G,j}(\|S_j\|_2 + \|W_j\|_2)$. If the final projection maps are V_1, V_2, then

$$L_D \le \|V_2\|_2 (5/4) \|V_1\|_2 L_{G,0} \prod_{j=1}^{2} \left[1 + (5/4) L_{G,j}(\|S_j\|_2 + \|W_j\|_2)\right] \tag{30}$$

is a finite weight-based bound in evaluation mode. Output denormalization contributes its linear-map norm. These constants cover the main readouts and can be loose. We have not evaluated the constants or representation residuals for the reported checkpoints; Eq. (26) is a conditional analysis, not a numerical explanation of their error magnitudes.

A.6 Observation protocols and reported field metrics

Mixed-density pretraining. Suppose the batch protocol is sampled from a finite set with probabilities $\pi_p > 0$, and $\mathcal{L}_p$ is the final-level population loss under protocol p, with the same complete-field distribution. Nonnegativity gives $\mathcal{L}_{\mathrm{mix}} = \sum_p \pi_p \mathcal{L}_p \geq \pi_p \mathcal{L}_p$. Therefore the normalized latent error on protocol p is bounded by $\mathcal{L}_{\mathrm{mix}}/(\pi_p \kappa_p)$, where κ_p is its coverage constant; Theorem 3.2 uses the square root of that quantity and residuals evaluated under p. This permits task/rate sharing within a minibatch without treating positions as independent. It does not certify a density or mask distribution absent from the mixture. The latent loss can instead be evaluated under the fixed 3% downstream observation law; the exact pretraining rate list is needed to infer π_p from a mixed loss. The reported 2%-train/5%-test controls are empirical shift tests, not consequences of this mixture inequality.

Relative error. Write $\mathcal{B}_{\mathrm{field}} = \delta_t + L_D[\delta_{\mathrm{FT}} + \delta_{\mathrm{int}} + K_A\{\delta_{\mathrm{scale}} + R\sqrt{\mathcal{L}_{\mathrm{final}}/\kappa}\}]$. If the physical target has $\|Y\|_h \geq m > 0$ almost surely, then Cauchy–Schwarz gives $\mathbb{E}[\|\hat{Y} - Y\|_h/\|Y\|_h] \leq \mathcal{B}_{\mathrm{field}}/m$ and $\mathbb{E}[\|\hat{Y} - Y\|_h^2/\|Y\|_h^2] \leq \mathcal{B}_{\mathrm{field}}^2/m^2$. Multiplication by 100 expresses percentages. These implications require the stated lower bound and the same physical units; they are not numerical bounds for the published Relative L_2 results. A denominator floor instead yields a bound using that floor, which may be uninformative.

Binary coefficient recovery. For scalar coefficients taking two values separated by $\Delta > 0$, threshold a real-valued prediction at their midpoint. A misclassified position must have absolute error at least $\Delta/2$, hence its indicator is at most $4|\hat{Y}_i - Y_i|^2/\Delta^2$. Averaging positions and examples yields $\mathbb{E}\,\mathrm{BER} \leq \min\{1, 4\mathcal{B}_{\mathrm{field}}^2/\Delta^2\}$. The threshold and coefficient scale must match this premise; the result does not convert arbitrary latent distance directly into the benchmark's BER.

A.7 A finite-sample statement for final checkpoints

Proposition A.2 (Independent evaluation of a fixed checkpoint family)**.** *Let $K \geq 1$ student–teacher checkpoint pairs be fixed before drawing $m \geq 1$ independent evaluation batches with the law of Appendix A.1. The pairs may depend arbitrarily on separate training data, including through EMA. For each pair j, recompute its final-level loss on these batches and denote its average by $\widehat{\mathcal{L}}_j$. For $0 < \delta < 1$, with probability at least $1 - \delta$, simultaneously for all j,*

$$\mathcal{L}_j \leq \min\left\{U, \widehat{\mathcal{L}}_j + U\sqrt{\frac{\log(K/\delta)}{2m}}\right\}, \qquad U = 4(\lambda_m + \lambda_c). \tag{31}$$

On this event, the right side may replace $\mathcal{L}_{\mathrm{final}}$ in Theorems 3.1 and 3.2 for any selected pair from the family. The field bound still requires the other residuals and the readout constant; this proposition does not estimate them.

Proof. Condition on the training data and the entire checkpoint family. Each pair's batch losses are independent, identically distributed in $[0, U]$. For $Z \in [0, U]$, set $F(s) = \log \mathbb{E}e^{s(\mathbb{E}Z - Z)}$. Its second derivative is the variance of Z under exponential tilting, at most $U^2/4$: under any law on $[0, U]$, $\mathrm{Var}(Z) \leq \mathbb{E}(Z - U/2)^2 \leq U^2/4$. Since $F(0) = F'(0) = 0$, integration gives $F(s) \leq s^2U^2/8$ for $s \geq 0$. Multiplying these exponential-moment bounds for the m independent losses and applying Markov's inequality gives $\Pr(\mathcal{L}_j - \widehat{\mathcal{L}}_j > t) \leq \inf_{s>0} \exp(-smt + ms^2U^2/8) = \exp(-2mt^2/U^2)$. Set $t = U\sqrt{\log(K/\delta)/(2m)}$ and apply a union bound over K pairs. The deterministic bound $\mathcal{L}_j \leq U$ gives the minimum. Removing the conditioning preserves the probability statement. □

The independent sampling unit is a fresh batch of PDE instances together with its observation masks and auxiliary target-block draws. Spatial positions, repeated training epochs, and repeated masks of the same fixed collection of fields do not supply independent population batches. The proposition concerns losses recomputed with fixed network parameters, rather than an online loss average accumulated while the student and EMA teacher change. For an independent evaluation of a single terminal checkpoint pair, the specialization $K = 1$ applies; the general K statement provides simultaneous coverage for a prespecified family. The reported main models use terminal checkpoints, without validation-based checkpoint selection. This proposition does not estimate the other residuals in Theorem 3.2 and is not used to claim empirical confidence intervals for the reported field errors.

A.8 COUNTEREXAMPLES AND THE PRECISE INTERPRETATION

Coverage is necessary. If context weight is zero and the observed-position variant is used, an always-observed position is never in M. Set student and teacher equal elsewhere and different at that position. The loss is zero but full-grid normalized error is positive. Thus a positive uniform κ cannot be inferred from missing-position training alone. For complement context, omitting a coefficient similarly removes the pointwise guarantee on the associated region. This is a coverage statement, not a prediction that every context-weight ablation must rank monotonically.

The scalar examples below take $\mathsf{A} = I$ and $H_0 = \hat{z}$ unless stated otherwise.

Normalization cannot control raw scale. In a scalar, single-position example, take $\hat{z} = 1$, $z_t = c > 1$, $H = \hat{z}$, $D(v) = v$, and $Y = c$. Both normalized vectors equal one, so the latent loss is zero; teacher decoding is exact and there is no downstream drift. Nonetheless field error is $c - 1$, exactly the scale residual. Therefore a raw-readout bound that omits δ_{scale} is false, even with full coverage and nonzero features.

Latent agreement does not preclude collapse. Take both latents to be the same constant unit vector, with X uninformative and scalar $Y \in \{-1, 1\}$ equally likely. Let D return zero. The latent loss and both representation residuals are zero, but field RMSE is one. Here $\delta_t = 1$ and $\mathcal{V}_Y = 1$. This invalidates a bound that removes teacher task quality, and also shows why a latent conditional variance cannot stand in for physical ambiguity. EMA and stop-gradient alone are not proved to prevent this case.

Fine-tuning changes the object being decoded. Take $\hat{z} = z_t = 1$, $Y = 1$, $D(v) = v$, and final features $H = 1 + c$. The pretraining loss, scale residual, and teacher decoding error vanish, but downstream error is $|c| = \delta_{\mathrm{FT}}$. The downstream-change residual is therefore necessary even for exact initial alignment.

Representation transfer requires its own residual. Let $\hat{z} = z_t = 1$, $\mathsf{A}v = (v, 0)$, $H_0 = H = (1, c)$, $D(v_1, v_2) = v_2$, and $Y = 0$. Alignment, scale, fine-tuning change, and teacher decoding residuals vanish, but the field error is $|c| = \delta_{\mathrm{int}}$. Thus combining network modules does not remove this residual from a general field-error statement.

Distribution and causality. A student can agree with its teacher on every observation pattern present in training and disagree on a pattern outside that support. Neither mask coverage within the training law nor the checkpoint concentration result addresses that shift. Likewise, the theorems hold for any fixed network pair with the stated loss and interfaces; they do not prove optimization convergence or an advantage over direct supervision, uniform latent alignment, or another backbone. Their contribution is an implementation-aligned chain from achieved masked normalized loss to full-grid alignment and, with the explicit residuals, to physical-field error. Matched controls and readout experiments remain empirical evidence at their reported scope. Appendix C.7 clarifies the scope of the component evidence.

B EXPERIMENTAL PROTOCOL AND PDE PROBLEM DETAILS

This appendix specifies the data, task definitions, observation protocols, metrics, and training procedures used in Sec. 4. We distinguish the main benchmark from the representation and sensitivity studies, which use different training budgets and, where stated, different encoder initializations or observation samplers. Appendix C provides the additional backbone, sparse-training, density, noise, and readout experiments.

B.1 COMMON EXPERIMENTAL SETTING

B.1.1 JENO OPERATORS

For a complete discretized instance $a_h = (g_h, f_h)$ on grid Ξ_h, the student receives $x_s = (u_h, m_h, S, \Xi_h)$ and the teacher receives $x_t = (a_h, \mathbf{1}, \Xi_h)$. Here $u_h = m_h \odot S(a_h)$ contains

the sparse measurements, m_h identifies observed locations, and S identifies the observed field. The student projection and predictor define

$$z_s = B_\eta(E_\theta(x_s)), \qquad \widehat{z} = P_\phi(z_s) = J_\Theta(x_s), \qquad J_\Theta = P_\phi \circ B_\eta \circ E_\theta,$$

where $\Theta = (\theta, \eta, \phi)$. The complete-field target is $z_t = \mathrm{sg}[B_{\bar{\eta}}(E_{\bar{\theta}}(x_t))]$, with sg denoting stop-gradient. The teacher encoder is updated by exponential moving average (EMA). In the main models, encoder features have 64 channels and student and teacher latents have 32. Each teacher projector is a 1×1 convolution ($64 \to 32$), GELU, and 1×1 convolution ($32 \to 32$). It uses PyTorch's default Conv2d initialization (seed 42); its detached targets receive no loss gradient and its parameters remain at initialization, with no copy from the student or EMA update. The teacher encoder alone is initialized from the student and updated with momentum 0.996. The student projection/predictor is implemented jointly as 1×1 ($64 \to 32$), GELU, 3×3 ($32 \to 32$, padding 1), GELU, and 1×1 ($32 \to 32$), and is optimized with the student encoder. The target projection is the identity only in variants whose encoder and latent widths coincide. The analysis in Sec. 3.2 fixes the resulting complete target map at the evaluated checkpoint.

Main JENO pretraining uses latent supervision only. For the main downstream tasks, the pretrained student encoder is copied and jointly fine-tuned with a task-specific readout using dense targets. The deployed model consists of this adapted encoder and readout; the complete-field teacher, student projection, and pretraining predictor are not used at inference.

B.1.2 Task Notation and Forward/Inverse Maps

A complete instance contains a problem-defining field g_h, such as a coefficient, source, or initial state, and a response field f_h, such as a solution or terminal state. The selectors S and T specify the observed and predicted fields. For $m_h \in \{0,1\}^{128\times128}$, the general prediction task is

$$(u_h, m_h, S, \Xi_h) \mapsto r_h, \qquad u_h = m_h \odot S(a_h), \qquad r_h = T(a_h). \tag{32}$$

Unobserved entries are zero-filled. The mask is supplied separately, so a measured zero remains distinguishable from a missing value.

The forward and inverse directions are defined by

$$S(a_h) = g_h, \qquad T(a_h) = f_h \qquad \text{(forward)}, \tag{33}$$

$$S(a_h) = f_h, \qquad T(a_h) = g_h \qquad \text{(inverse)}. \tag{34}$$

Thus, forward prediction recovers the response from sparse observations of the problem-defining field, while inverse recovery reverses the direction. For time-dependent PDEs, these fields can be the initial and terminal states.

The full-recovery diagnostic observes both physical fields and predicts the complete pair. Writing $r_{\mathrm{in}} = S_{\mathrm{in}}(a_h)$ and $r_{\mathrm{out}} = S_{\mathrm{out}}(a_h)$, its input and target are

$$\left(m_h^{\mathrm{in}} \odot r_{\mathrm{in}}, m_h^{\mathrm{out}} \odot r_{\mathrm{out}}, m_h^{\mathrm{in}}, m_h^{\mathrm{out}}, S_{\mathrm{in}}, S_{\mathrm{out}}, \Xi_h\right) \mapsto (r_{\mathrm{in}}, r_{\mathrm{out}}). \tag{35}$$

Both masks have the same nominal observation rate. The implemented paired full-recovery view uses a shared spatial mask for the two fields. Its reported error averages the two field-wise relative errors and is distinct from the single-target forward and inverse metrics.

B.1.3 Observation Protocol

The main benchmark uses a nominal 3% observation rate. The DiffusionPDE/FunDPS reference protocol uses 500 sensors for the first four PDE settings (Huang et al., 2024; Yao et al., 2025):

$$K_{\mathrm{ref}} = 500, \qquad \frac{K_{\mathrm{ref}}}{128^2} \simeq 3.05\%. \tag{36}$$

The point sampler used in the main JENO retest instead draws each grid entry independently from $\mathrm{Bernoulli}(\rho)$. Consequently, $K \sim \mathrm{Binomial}(128^2, \rho)$ and $\mathbb{E}[K] = 491.52$ at $\rho = 0.03$. This is point sampling with observation patch size 1; the realized count varies across examples. Fixed-count sampling in the noise and readout studies is specified separately in Appendix C.

The NS (BCs) reference entries use boundary observations together with 1% random interior observations, whereas the JENO entries are reported at nominal 3%. We retain these results without ranking

Table 4: **Partial-OBS datasets.** Each sample has stored shape $2 \times 128 \times 128$. Forward and inverse tasks swap the observed and predicted channels. The symbols in the channel columns follow the PDE-specific notation below.

Dataset key	PDE family	Input-side channel	Output-side channel	Train	Test	Main evaluation
`darcy`	Darcy flow	Permeability c	Solution u	50,000	10,000	1,000
`poisson`	Poisson	Source f	Solution u	50,000	1,024	1,000
`helmholtz`	Helmholtz	Source f	Solution u	50,000	10,000	1,000
`ns-nonbounded`	Navier–Stokes	Initial vorticity ω_0	Terminal vorticity ω_T	50,000	1,000	1,000
`ns-bounded`	Navier–Stokes with BCs	Earlier scalar q_4	Later scalar q_8	14,000	1,000	1,000

the NS (BCs) columns. More generally, equal nominal density does not imply identical sensor counts or masks across literature-reported methods.

Forward and inverse tasks use the same nominal rate, applied to the field selected by S. The shared-teacher representation diagnostic evaluates $\rho \in \{1\%, 2\%, 5\%, 10\%\}$, and the context-weight and latent-dimension study evaluates $\rho \in \{2\%, 5\%, 10\%\}$. The additional fixed-model density sweep extends to full observation.

B.1.4 Data Release, Download Instructions, and Splits

We use the normalized Partial-OBS datasets released with FunDPS and processed from Diffusion-PDE (Huang et al., 2024; Yao et al., 2025). The released fields follow the FunDPS normalization convention, with zero mean and standard deviation 0.5. The dataset keys are `darcy`, `poisson`, `helmholtz`, `ns-nonbounded`, and `ns-bounded`. The release can be obtained using the repository's download script:

```
git clone https://github.com/neuraloperator/FunDPS
cd FunDPS
python utils/download_dataset.py all
python utils/download_dataset.py all --test
```

The normalized Hugging Face dataset is `jcy20/DiffusionPDE-normalized`. Table 4 gives the released training and test sizes. Each processed sample stores two task-relevant channels on a 128×128 grid. The main quantitative evaluation uses the first $1{,}000$ instances in each released test file, in file order, without subsampling by difficulty. The additional training-slice normalization and validation partition used by the JENO loader are described in Appendix B.3.1.

B.1.5 PDE Problem Definitions

The equations below specify the physical fields in the released datasets. Their local symbols follow the dataset convention: for example, f denotes a source and u a solution in the Poisson and Helmholtz equations. Their roles as the generic input-side and output-side fields are determined by the selectors in Appendix B.1.2.

Darcy flow. On the unit square, the pressure solution satisfies

$$-\nabla \cdot (c(x)\nabla u(x)) = 1, \quad x \in (0,1)^2, \qquad u(x) = 0, \quad x \in \partial(0,1)^2.$$

The permeability is generated by thresholding a Gaussian random field, $c \sim h_{\#}\mathcal{N}(0, (-\Delta + 9I)^{-2})$, where $h(r) = 12$ for $r > 0$ and $h(r) = 3$ otherwise. Forward prediction recovers u from sparse observations of c; inverse recovery predicts the binary permeability labels from sparse observations of u. Darcy inverse uses binary error rate.

Poisson equation. The source and solution satisfy

$$\Delta u(x) = f(x), \quad x \in (0,1)^2, \qquad u(x) = 0, \quad x \in \partial(0,1)^2,$$

with $f \sim \mathcal{N}(0, (-\Delta + 9I)^{-2})$. The forward task predicts the dense solution from sparse source measurements, and the inverse task predicts the dense source from sparse solution measurements.

Helmholtz equation. The benchmark uses

$$\Delta u(x) + k^2 u(x) = f(x), \quad x \in (0,1)^2, \qquad u(x) = 0, \quad x \in \partial(0,1)^2,$$

with $k = 1$ and Gaussian-random-field sources following the released benchmark convention. As in Poisson, forward prediction maps sparse source observations to the dense solution, while inverse recovery reverses this direction.

Unbounded Navier–Stokes. The name follows the benchmark terminology; the simulation uses periodic boundary conditions on the unit square. Let v denote velocity and $\omega = \nabla \times v$ vorticity. The dynamics satisfy

$$\partial_t \omega + v \cdot \nabla \omega = \nu \Delta \omega + f, \qquad \nabla \cdot v = 0, \qquad x \in (0,1)^2, \quad t \in (0,T],$$

where $\nu = 10^{-3}$, $T = 1$, and

$$f(x) = \frac{1}{10}\left[\sin\big(2\pi(x_1+x_2)\big) + \cos\big(2\pi(x_1+x_2)\big)\right], \qquad \omega_0 \sim \mathcal{N}\Big(0, 7^{3/2}(-\Delta + 49I)^{-5/2}\Big).$$

Forward prediction recovers terminal vorticity ω_T from sparse observations of ω_0, and inverse recovery predicts ω_0 from sparse observations of ω_T.

Navier–Stokes with boundary conditions. This benchmark contains flows through domains with randomly placed cylindrical obstacles. Velocity v and pressure p satisfy

$$\partial_t v + v \cdot \nabla v = -\nabla p + \nu \nabla^2 v, \qquad \nabla \cdot v = 0, \qquad x \in \Omega, \quad t \in (0,T].$$

The viscosity is $\nu = 0.001$, the fluid density is 1.0, and boundary conditions and obstacle geometries follow the released simulator. The preprocessing pairs stored snapshot indices 4 and 8 (Yao et al., 2025). The stored channels are the scalar arrays `v.npy[...,4]` and `v.npy[...,8]`, denoted q_4 and q_8; they do not constitute two-component velocity vectors. The released loader alone does not identify the scalar as a particular velocity component or speed, so we use the stored-field notation here. Snapshot indices do not specify a four-second interval. Forward prediction recovers q_8 from sparse measurements of q_4, and inverse recovery reverses this direction. The main JENO input contains no separate obstacle-geometry channel. Its Bernoulli sampler can select any grid location, including boundary and obstacle pixels, and its reported errors include the entire 128×128 grid.

B.1.6 Baselines and Reported Values

Published deterministic results. The full-observation-trained FNO, PINO, DeepONet, and PINN rows in Table 1 are the benchmark results reported by DiffusionPDE and reproduced in the FunDPS comparison (Huang et al., 2024; Yao et al., 2025). They are distinct from our mask-aware controls. The full-observation-trained FNO and DeepONet in the subsequent diagnostics also follow the corresponding DiffusionPDE architectures and training protocols. The underlying methods are FNO (Li et al., 2021a), PINO (Li et al., 2021b), DeepONet (Lu et al., 2021), and PINN (Raissi et al., 2019).

Mask-aware deterministic controls. The main-table FNO uses width 40, 14 Fourier modes, four layers, projection width 128, and padding 9. The main-table DeepONet has a CNN branch with channels $[32, 64, 128, 192]$, branch hidden width 360, basis size 256, and a coordinate trunk with hidden width 128 and depth 2. CNO (Raonić et al., 2023) uses three levels, two residual blocks, four neck blocks, channel multiplier 52, and BatchNorm. These controls use sparse, mask-aware inputs and the released test files; their executed training and evaluation settings are given in Table 5 and below. The MLP sensor-branch DeepONet in Table 3 is a separate non-convolutional implementation.

Executed mask-aware training protocol. Each FNO, CNO, and convolutional DeepONet starts from random initialization and predicts both dense channels from masked values, separate masks, and coordinates. Values are multiplied by their masks before entering the model. The field-only objective averages the two per-instance, unsquared physical Relative L_2 errors, with no teacher, pretraining, or auxiliary objective. Inputs use the released-data normalization. Physical errors invert that affine transform; source-audited zero training targets use the training-split RMS as their finite scale. The test split is never used in training or checkpoint selection.

Each PDE uses 50,000 training records (14,000 for NS (BCs)), 100 epochs, batch size 125, and the terminal epoch-100 checkpoint. This gives 40,000 updates per model (11,200 for NS (BCs)). AdamW uses learning rate 1.25×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.05 on rank-at-least-two tensors, with zero decay on bias, normalization, and positional parameters. The learning rate warms up linearly from 10^{-6} for five epochs, then follows cosine decay to 10^{-6}. The

Table 5: **Executed main mask-aware controls.** The common loss, masks, optimizer, normalization, seed, and terminal-checkpoint rule are specified above. Counts include the two-output model.

Model	Parameters	Epochs / batch	Updates (other / NS (BCs))
FNO	5,029,946	100 / 125	40,000 / 11,200
DeepONet	4,977,354	100 / 125	40,000 / 11,200
CNO	4,774,776	100 / 125	40,000 / 11,200

gradient-norm clipping threshold is 1.0. All runs use seed 20260913, BF16 autocast with FP32 master weights, deterministic cuDNN, and TF32 disabled.

Training mixes five exact-budget observation families: uniform points, regular grids, clusters, random lines, and blocks. Each five-batch cycle contains one batch per family; forward/inverse/joint tasks occur with proportions $40/40/20\%$. Uniform/grid/cluster masks cycle through the budget slots $(500, 500, 1024, 2048, 4096, 8192, 12288, 16384)$; lines/blocks use 4915 or 9830 points. Thus, these main-table controls are *mixed-density mask-aware* models, evaluated with 500 uniform points; they are distinct from the matched-downstream-3% comparison in Table 3 and the 2% controls in Appendix C.2.

Test evaluation. The main-table control values are held-out test results, not the field errors logged on training minibatches. Evaluation uses uniform 500-point masks, mask seed 20261013, and BF16 inference. It includes all $10{,}000$ Darcy and Helmholtz test records, $1{,}024$ Poisson records, and $1{,}000$ periodic-NS and NS (BCs) inverse records. NS (BCs) forward excludes three source-audited zero targets and uses 997 records. Darcy inverse uses BER; the remaining control errors are physical-space Relative L_2. JENO's archived retest instead uses the first $1{,}000$ records, Bernoulli masks, and released-data-space Relative L_2 (Appendix B.3.1). Reported errors therefore retain these evaluation differences; they are not estimates from a common-mask, common-normalization rerun.

Generative and sampling-based results. We include DiffusionPDE (Huang et al., 2024), FunDPS and FunDPS+ (Yao et al., 2025), PRISMA-50 (Sawhney et al., 2025), Equi-FunDPS (Tolooshams et al., 2025), PISD (Gallon et al., 2026), and DDIS (Lin et al., 2026). The DiffusionPDE values are the reproduction reported by FunDPS, not the original paper's headline results. FunDPS and FunDPS+ use the reported 200- and 500-step results, respectively, and PRISMA-50 uses 50 steps. PISD entries are the means in its original mean$\pm$standard-deviation table. DDIS entries are inverse-only values from its best standard-supervision time-budget row.

Published values retain their original protocols. In particular, the copied DiffusionPDE/FunDPS NS (BCs) entries use boundary observations and 1% interior sensors, so those columns are not ranked against the nominal-3% JENO entries. A dash denotes either an unreported endpoint or an excluded task/protocol. In particular, PRISMA-50's cited v1 reports sparse NS results but uses next-step prediction with $\Delta t = 0.1$, rather than our initial-to-terminal task; we therefore exclude those NS values. Its Darcy entries are from v1, Table 7. PISD's trajectory-generation NS task is likewise excluded. Table 6 records the source versions, evaluation sizes, observation conventions, and sampler budgets. Bold and underline in the first four main-table settings identify the lowest and second-lowest reported errors, not statistical significance. Repeated-mask JENO estimates are given in Table 13.

B.1.7 Evaluation Metrics

For a non-binary target r and prediction $\widehat{r}$, the field error is

$$\text{Rel-}L_2(\widehat{r}, r) = \frac{\|\widehat{r} - r\|_2}{\max\{\|r\|_2, 10^{-12}\}}. \tag{37}$$

Forward evaluation measures the predicted response field. Inverse evaluation measures the predicted coefficient, source, or initial-state field. The exception is Darcy inverse, for which we use binary error rate:

$$\text{BER}(\widehat{c}, c) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\widehat{c}_i^{\text{bin}} \neq c_i^{\text{bin}}\right], \tag{38}$$

where the main JENO evaluator thresholds both the continuous prediction and target at $\tau_c = (\min_i c_i + \max_i c_i)/2$ for each instance, using labels $\mathbf{1}[c_i \geq \tau_c]$ and $\mathbf{1}[\widehat{c}_i \geq \tau_c]$. The threshold is in released-

Table 6: **Provenance of literature-derived errors.** Versions are arXiv versions; table numbers refer to the source. D/P/H/N/B denote Darcy/Poisson/Helmholtz/NS/NS (BCs), and F/I denote forward/inverse. Metrics are percentages: Relative L_2, except Darcy inverse (BER). NR means the source does not specify the item. These rows retain their source evaluation protocols.

Method / source	Version; table	Endpoints	Test count; observations	Sampling budget
FNO, PINO, DeepONet, PINN, via FunDPS (Yao et al., 2025)	2505.17004v2; 1	D/P/H/N/B, F/I	1,000; 500 points for D/P/H/N; boundary + 1% interior for B	Not a diffusion sampler
DiffusionPDE reproduction, FunDPS, FunDPS+ (Yao et al., 2025)	2505.17004v2; 1	D/P/H/N/B, F/I	Same source protocol above	2,000 / 200 / 500 steps
PRISMA-50 (Sawhney et al., 2025)	2512.01370v1; 7	D, F/I	1,000; 3% observed	50 steps
Equi-FunDPS (Tolooshams et al., 2025)	2505.22973v1; 4	H/N, F/I	100 randomly selected; FunDPS observations	500 steps
PISD (Gallon et al., 2026)	2602.09708v1; 1–2	P/H, F/I	100 independent runs; 500 points	NR
DDIS (Lin et al., 2026)	2601.23280v1; 2, last standard-budget row	P/H/N, I	Count NR; 500 random points	200 annealing stages, 5 diffusion and 100 Langevin steps per stage

data coordinates after undoing training-slice normalization. Darcy inverse is trained by continuous regression, not a classification loss. Thus, Darcy forward uses Relative L_2, whereas Darcy inverse uses BER.

For full recovery from sparse observations of both fields, we average their relative errors:

$$\mathrm{Err}_{\mathrm{full}} = \frac{1}{2}\left(\frac{\|\widehat{r}_{\mathrm{in}} - r_{\mathrm{in}}\|_2}{\|r_{\mathrm{in}}\|_2} + \frac{\|\widehat{r}_{\mathrm{out}} - r_{\mathrm{out}}\|_2}{\|r_{\mathrm{out}}\|_2}\right). \tag{39}$$

Field-prediction errors in the tables are multiplied by 100 and reported as percentages; lower is better. Latent mismatch and CKA are separate representation statistics and are not percentages.

B.2 Representation Diagnostics

These experiments examine the relation between sparse-view features and the representation of the corresponding complete PDE instance. We use Poisson and unbounded Navier–Stokes, reporting latent mismatch, linear CKA, and qualitative field predictions.

B.2.1 Diagnostic Protocol and Implementation Details

The full-observation-trained FNO and convolutional DeepONet used in these diagnostics, including Figure 2, follow the architectures and full-observation training procedures of the corresponding DiffusionPDE baselines (Huang et al., 2024). They are evaluated with masked inputs. JENO instead uses its pretrained sparse-input latent predictor and complete-field teacher reference. Ridge and CKA diagnostics use the test dataset. Table 7 summarizes the protocol. Because the direct controls undergo full-to-partial input shift, these results complement rather than replace the sparse-trained comparisons in the main text and Appendix C.

B.2.2 Full-Field Latent Mismatch and Ridge Risk

Let $X_\rho^{(j)}$ be the sparse view of test instance j at density ρ, and let $z_t^{(j)}$ be its complete-field latent under the fixed teacher. For model m, write $H_m(X_\rho^{(j)})$ for the diagnostic representation. JENO uses its predicted latent $\widehat{z}_\rho^{(j)} = J_\Theta(X_\rho^{(j)})$; FNO and DeepONet use the penultimate features before output projection.

Table 7: **Representation-diagnostic protocol for Figure 2 and Table 8.** All diagnostic models are evaluated under the stated partial-observation settings.

Item	Setting
PDEs	Poisson and unbounded Navier–Stokes.
Observation ratios	1%, 2%, 5%, and 10%.
JENO training data and budget	10,000 examples per PDE; 30 epochs.
JENO optimizer	AdamW; learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 8.
FNO control	Corresponding DiffusionPDE architecture and full-observation training protocol; sparse-input testing (Huang et al., 2024).
DeepONet control	Corresponding DiffusionPDE architecture and full-observation training protocol; sparse-input testing (Huang et al., 2024).
JENO representation	Predicted latent from sparse observations, compared with the complete-field teacher target.
JENO objective	Latent prediction with $\lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c$; no dense-field reconstruction loss.
Diagnostics	Full-field latent mismatch/ridge risk and linear CKA.

For the direct controls, a linear probe maps these features to the shared teacher space. It is fitted by ridge regression on the same test dataset $\mathcal{D}_{\text{test}}$ used to compute the diagnostic:

$$P^{\star}_{m,\rho} = \arg\min_{P} \sum_{j \in \mathcal{D}_{\text{test}}} \left\| P\Big(H_m(X^{(j)}_{\rho})\Big) - z^{(j)}_t \right\|_2^2 + \lambda_{\text{ridge}} \|P\|_F^2.$$

Here P is a linear map from the model's diagnostic features into the shared teacher latent space, and $\lambda_{\text{ridge}} \geq 0$ controls the strength of its squared-Frobenius-norm penalty. The resulting in-sample probe mismatch is

$$\mathcal{M}_{\text{latent}}(m, \rho) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{j \in \mathcal{D}_{\text{test}}} \frac{\left\| P^{\star}_{m,\rho}\Big(H_m(X^{(j)}_{\rho})\Big) - z^{(j)}_t \right\|_2^2}{\|z^{(j)}_t\|_2^2 + \epsilon}.$$

JENO already predicts in the teacher latent space and therefore uses the identity probe:

$$\mathcal{M}_{\text{latent}}(\text{JENO}, \rho) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{j \in \mathcal{D}_{\text{test}}} \frac{\|\widehat{z}^{(j)}_{\rho} - z^{(j)}_t\|_2^2}{\|z^{(j)}_t\|_2^2 + \epsilon}.$$

Here $\epsilon > 0$ stabilizes the denominator. Lower mismatch indicates better recovery of this particular teacher reference. Its value depends on latent scale, the reference representation, and probe fitting, so downstream field error remains the primary cross-model performance measure.

B.2.3 Linear CKA Alignment

Linear centered kernel alignment (CKA) measures the similarity between the geometry of two representations (Kornblith et al., 2019). For a fixed model and observation density, flatten each test example's features into one row of $H \in \mathbb{R}^{n \times D_H}$, and form $Z \in \mathbb{R}^{n \times D_Z}$ from the corresponding complete-field teacher latents. The rows must refer to the same examples in the same order. To compute centered linear CKA, center each feature column across examples:

$$J_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}, \qquad H_c = J_n H, \qquad Z_c = J_n Z.$$

The metric is

$$\text{CKA}(H, Z) = \frac{\|H_c^{\top} Z_c\|_F^2}{\|H_c^{\top} H_c\|_F \, \|Z_c^{\top} Z_c\|_F}. \tag{40}$$

Centering is across examples, not across the channels within each example, and is distinct from the channel normalization in the pretraining loss. Higher CKA indicates stronger alignment of the two feature geometries. A vanishing denominator makes CKA undefined; the archived N/A endpoint is discussed separately in Appendix C.6. CKA complements field error and latent mismatch, but does not by itself establish physical consistency or non-collapse.

Table 8: **Full-field representation diagnostics.** Entries correspond to observation ratios $1\%/2\%/5\%/10\%$. Lower latent mismatch/ridge risk and higher CKA are better. These statistics use a shared teacher reference rather than a common physical-field error.

PDE	Model	Latent mismatch / ridge risk ↓	CKA ↑
Poisson	JENO	0.163 / 0.134 / 0.114 / 0.105	0.961 / 0.966 / 0.970 / 0.973
	FNO	0.908 / 0.904 / 0.892 / 0.868	0.733 / 0.755 / 0.777 / 0.802
	DeepONet	0.816 / 0.816 / 0.815 / 0.823	0.577 / 0.629 / 0.675 / 0.712
Navier–Stokes	JENO	0.187 / 0.147 / 0.127 / 0.118	0.970 / 0.980 / 0.983 / 0.983
	FNO	0.809 / 0.806 / 0.797 / 0.779	0.755 / 0.819 / 0.896 / 0.933
	DeepONet	1.163 / 1.168 / 1.184 / 1.219	0.328 / 0.385 / 0.469 / 0.514

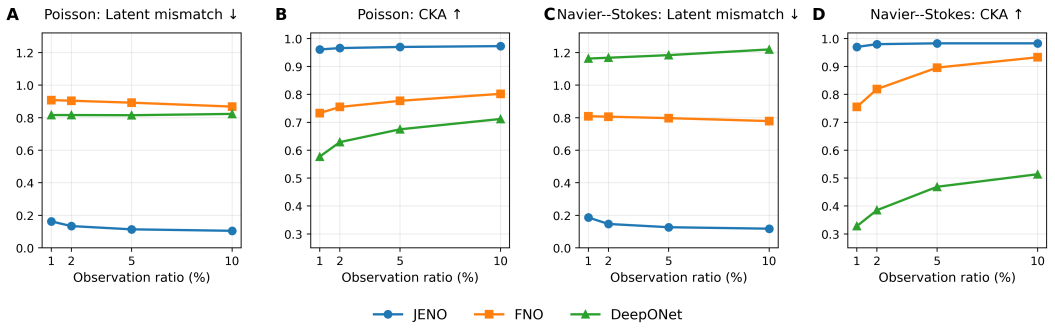


Figure 2: **Representation diagnostics under partial observations.** Latent mismatch and CKA are evaluated against a complete-field teacher reference. FNO and convolutional DeepONet are trained on complete inputs and tested after masking, so their results include input-distribution shift.

B.2.4 Numerical Results and Qualitative Comparisons

Table 8 gives the numerical results plotted in Figure 2. Entries follow the order $1\%/2\%/5\%/10\%$. JENO has lower mismatch and higher CKA in this shared-teacher diagnostic.

Figures 3 and 4 show physical-field predictions and absolute-error maps from the main JENO terminal checkpoints (Table 11), using the readouts in Table 10 jointly fine-tuned with the encoder under dense-field supervision (Appendix B.3.1). These main models are distinct from the latent-only diagnostic models in Table 7. FNO and DeepONet use the full-observation-trained controls described in Appendix B.1.6.

B.3 Main Partial-OBS Benchmark

The main benchmark evaluates forward and inverse recovery on five PDE settings spanning four PDE families. The main readouts produce the dense pair; each endpoint evaluates only the requested target field. The training and evaluation conventions below apply to the main JENO models; the additional studies identify their distinct initializations and observation samplers separately.

B.3.1 Common JENO Training and Inference Convention

Two-stage training. The first stage pretrains a sparse-input student against complete-field teacher latents. The second stage copies the pretrained student encoder and jointly fine-tunes it with a task-specific readout. The saved pretraining model is retained unchanged, while its copy is adapted separately for forward and inverse prediction. Dense-field reconstruction is disabled during pretraining.

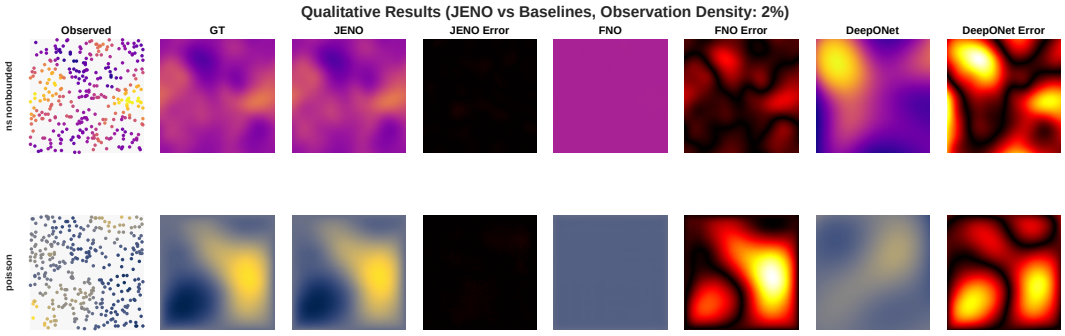


Figure 3: **Qualitative predictions at 2% observation density.** Columns show the observed input, ground truth, predictions, and absolute-error maps. JENO uses the main task model; FNO and DeepONet are the full-observation-trained controls.

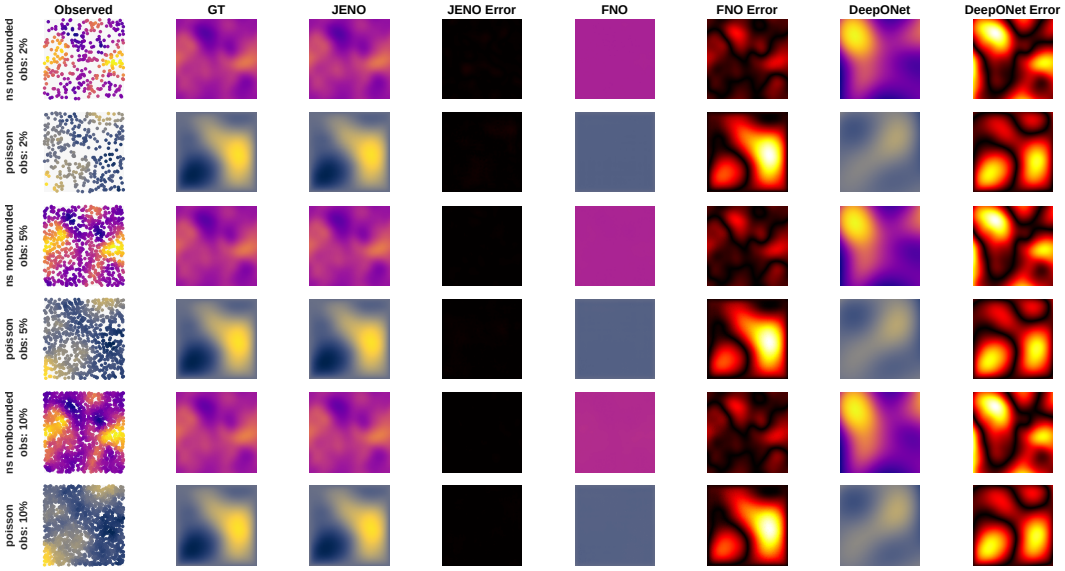


Figure 4: **Qualitative comparison across observation densities.** Rows correspond to Poisson and unbounded Navier–Stokes at 2%, 5%, and 10% observations. Columns display sparse inputs, ground truth, and predictions and absolute errors for the main JENO models and the full-observation-trained FNO and DeepONet controls.

Observation mixing. Mixed pretraining samples a density uniformly from $\{1, 2, 3, 5, 10, 15, 20\}\%$ and independently selects forward, inverse, or paired full-recovery observations with probability $1/3$ each, per minibatch. This stage lasts 100 epochs for Darcy, Poisson, Helmholtz, and unbounded NS, followed by 50 epochs at 3% with forward/inverse modes sampled equally. NS (BCs) uses 200 mixed-density epochs followed by 100 such 3% epochs. The continuation's observation-resampling seed group changes every five epochs. Separate downstream tasks then use fixed nominal 3% observations. Table 9 summarizes the common parameters.

Optimizer. Main pretraining and fine-tuning use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-4}, and constant learning rates 3×10^{-4} and 2×10^{-4}, respectively. Batch sizes are 8 for Darcy, Poisson, and Helmholtz and 16 for both NS settings. These main-run trainers use one optimizer step per minibatch, without a learning-rate scheduler, gradient accumulation, gradient clipping, or automatic mixed precision. The separate paired-task diagnostic recipe is specified in Appendix C.4.

Table 9: **Common JENO pretraining and adaptation parameters.** The teacher supplies complete-field latent targets during pretraining only.

Item	Setting
Student input	Sparse measurements, masks, task information, and coordinates
Teacher input	Complete paired fields, all-one masks, and coordinates
Pretraining objective	Sampled-target and complement latent prediction
Dense-field pretraining loss weight	0
Sampled-target weight λ_m	1.0
Complement weight λ_c	0.25
Teacher encoder update	EMA with momentum 0.996
Pretraining learning rate	3×10^{-4}
Fine-tuning learning rate	2×10^{-4}
Weight decay	1×10^{-4}
Mixed pretraining tasks	Forward / inverse / full recovery, probability $1/3$ each
Mixed pretraining densities	$\{1, 2, 3, 5, 10, 15, 20\}\%$, probability $1/7$ each
Downstream observation density	3%

Spatial masks and latent indexing. The FNO encoder and latent heads preserve the $H \times W$ grid. Latent tensors have shape $B \times d_z \times H \times W$, and loss masks have shape $B \times 1 \times H \times W$, broadcast over channels at the corresponding spatial indices. In a forward or inverse input, only the selected physical channel and its observation mask are exposed; the other channel and its mask are zero. Coordinates are appended by the encoder. The full-recovery view uses a shared mask for both fields. Spatial indexing determines where supervision is applied, not which physical pixels exclusively determine a feature.

For each example, four 16×16 blocks are sampled independently and uniformly over valid top-left positions, allowing overlap. Their union is intersected with the unobserved region to obtain M. If this intersection is empty, M falls back to the entire unobserved region. The complement weights are

$$C = (1 - M)\gamma, \qquad \gamma = 1 + 2\bar{G}\left(0.5 + 0.5\,\mathrm{dilate}_5(M)\right), \tag{41}$$

Here, for the complete normalized output-side channel f, $G_{ij} = \mathbf{1}_{i>0}|f_{ij} - f_{i-1,j}| + \mathbf{1}_{j>0}|f_{ij} - f_{i,j-1}|$ and $\bar{G} = G/(\max_{i,j} G_{ij} + 10^{-6})$, separately per example. Differences are not divided by grid spacing, and missing boundary differences are zero, without periodic wrapping. Both task directions use this same output-side channel (the solution or later state), even when it is the observed channel. The operator dilate_5 is 5×5 max pooling with stride 1 and padding 2; padding contributes no positive mask value. Thus, $1 \le \gamma \le 3$. The complement includes both observed positions and unobserved positions outside M. Complete-field gradients allocate training supervision; neither these weights nor the auxiliary target blocks are network inputs or inference requirements.

Loss reduction. Each supervised latent vector is normalized over channels with $\nu_\epsilon(v) = v/\max\{\|v\|_2, \epsilon\}$ and $\epsilon = 10^{-12}$. The target and complement losses sum weighted squared channel distances over the entire batch and spatial grid, then divide by their respective total spatial weights, floored at $\epsilon_\ell = 10^{-6}$. There is no additional division by latent dimension, and an empty region contributes zero. The final latent width is 32. Levels 1, 3, and 5 are zero-based FNO-block indices: each supplies a $B \times 64 \times 128 \times 128$ tensor before the encoder's final GroupNorm. Each has its own student predictor and teacher projector with the architectures in Appendix B.1.1, yielding $B \times 32 \times 128 \times 128$ latents. The final normalized encoder output has separate heads. Heads are not shared across levels, and the same spatial masks and weights are reused at every level. Losses at these three levels and the final prediction level are summed rather than averaged:

$$\mathcal{L}_{\mathrm{pre}} = \sum_{j \in \mathcal{J}} \left(\lambda_m \mathcal{L}_m^{(j)} + \lambda_c \mathcal{L}_c^{(j)}\right), \qquad \lambda_m = 1, \quad \lambda_c = 0.25.$$

The shorthand $\lambda_m\mathcal{L}_m + \lambda_c\mathcal{L}_c$ therefore uses $\mathcal{L}_m = \sum_j \mathcal{L}_m^{(j)}$ and $\mathcal{L}_c = \sum_j \mathcal{L}_c^{(j)}$. The batch reductions agree with Eq. (4).

Training splits and random seeds. The pretraining loader reserves $1{,}000$ examples from each training file for validation and fits its normalization on the remaining training slice. The two-layer-FNO-head runs for Darcy, Poisson, and NS (BCs) use all training-file examples during fine-tuning

Table 10: **Main-model architectures.** Each downstream encoder is initialized from the pretrained student encoder and jointly fine-tuned with the listed readout.

PDE	Encoder	Width	Depth	Fourier modes	Readout
Darcy	FNO	64	6	16×16	Two-layer FNO head
Poisson	FNO	64	6	16×16	Two-layer FNO head
Helmholtz	FNO	64	6	16×16	Standard FNO decoder
Navier–Stokes	FNO	64	6	16×16	Standard FNO decoder
Navier–Stokes (BCs)	FNO	64	6	16×16	Two-layer FNO head

Table 11: **Main-model training schedules.** Pretraining combines the initial mixed-observation stage with its continuation. Fine-tuning is performed separately for forward and inverse recovery.

PDE	Pretraining epochs	Pretraining observations	Fine-tuning readout	Fine-tuning epochs
Darcy	150	Mixed observations + continuation	Two-layer FNO head	50
Poisson	150	Mixed observations + continuation	Two-layer FNO head	50
Helmholtz	150	Mixed observations + continuation	Standard FNO decoder	30
Navier–Stokes	150	Mixed observations + continuation	Standard FNO decoder	30
Navier–Stokes (BCs)	300	Mixed observations + random 3% continuation	Two-layer FNO head	100

with that normalization; the standard-decoder runs retain the loader's training slice. Test and OOD evaluations use separate files, with $1{,}000$ examples per evaluated split. The JENO pretraining, main fine-tuning, and readout-swap runs use seed 42 for Python, NumPy, and PyTorch. The main retest uses mask seed base $12{,}345$ and five mask repeats. Noise perturbation seeds are specified in Appendix C.4.

Normalization and downstream loss. Starting from the normalized released fields, the loader computes a mean and sample standard deviation per channel over training instances and grid positions, flooring the standard deviation at 10^{-6}. It applies this transform before masking; missing values are zero in model-normalized coordinates. Each main readout predicts both normalized channels and minimizes $\mathcal{L}_{\mathrm{FT}} = (2BHW)^{-1} \sum_{b,c,i,j} |\widehat{a}_{bcij} - a_{bcij}|$, with no auxiliary latent, gradient, or PDE loss. Evaluation reverses this training-slice transform for predictions and targets, selects the requested channel, computes its per-instance metric, and averages over examples (and over masks as specified below). It does not undo the release's original normalization: the reported main errors are in released-data coordinates, not unnormalized physical units.

Readout layers. The standard FNO decoder takes the 64-channel encoder output through 1×1 ($64 \to 64$), GELU, 3×3 ($64 \to 64$, padding 1), GELU, and 1×1 ($64 \to 2$) convolutions. The two-layer FNO head uses two width-64 FNO blocks with 16×16 modes, GroupNorm with one group, 1×1 ($64 \to 64$), GELU, and 1×1 ($64 \to 2$). Both preserve the grid and take encoder features only; neither has a separate direct-observation skip input. Selecting the requested output channel gives D_ω in Eq. (6).

Readouts and terminal checkpoints. The readouts used for the main models are listed in Table 10. The copied student encoder and readout are jointly fine-tuned. Evaluation uses the **final training step of each prescribed schedule**, without validation-based checkpoint selection. The validation partition described above does not change this rule. Auxiliary test/OOD errors logged during training are not used to select the reported checkpoints. Readout sensitivity is evaluated separately in Appendix C.8.

Training schedules. Table 11 lists the terminal pretraining and fine-tuning epochs. The main models inherit the student encoder from pretraining epoch 150 for Darcy, Poisson, Helmholtz, and unbounded NS, and epoch 300 for NS (BCs). Darcy and Poisson readouts are fine-tuned for 50 epochs; Helmholtz and unbounded NS use 30 epochs. NS (BCs) fine-tuning continues from 50 to 100 epochs. In each case, evaluation uses the final encoder–readout state.

The noise study uses task-specific epoch-200 checkpoints from a separate paired-task run. The readout-swap experiment starts from those same task encoders and adds 20 epochs of joint encoder–readout fine-tuning. These are distinct from the main schedules, which explains why clean errors in Tables 1, 18, and 21 need not coincide.

Table 12: **Downstream task and reporting conventions.** All listed tasks use nominal 3% sparse observations and are fine-tuned separately.

Task	Observed field	Predicted field	Metric
Forward	Problem-defining field g	Response f	Relative L_2 of f
Inverse, except Darcy	Response f	Problem-defining field g	Relative L_2 of g
Darcy inverse	Response f	Continuous coefficient, then thresholded	BER

Table 13: **Repeated-mask retest of the main models.** Values are percentages. CI95 is 1.96 times the instance-level standard error; the final column is mean$\pm$standard error.

PDE	Task	Target	Metric	Mean	Standard error	CI95 half	Mean $\pm$ error bar
Darcy	Forward	u	Rel. L_2	4.313313	0.041951	0.082224	4.31 ± 0.04
	Inverse	a	BER	2.354734	0.036331	0.071208	2.35 ± 0.04
Poisson	Forward	u	Rel. L_2	1.797948	0.025124	0.049242	1.80 ± 0.03
	Inverse	a	Rel. L_2	9.715707	0.078229	0.153329	9.72 ± 0.08
Helmholtz	Forward	u	Rel. L_2	2.164425	0.026316	0.051580	2.16 ± 0.03
	Inverse	a	Rel. L_2	10.095286	0.077834	0.152555	10.10 ± 0.08
NS	Forward	u	Rel. L_2	2.781806	0.018750	0.036750	2.78 ± 0.02
	Inverse	a	Rel. L_2	7.362184	0.042442	0.083185	7.36 ± 0.04
NS (BCs)	Forward	u	Rel. L_2	4.077674	0.240998	0.472356	4.08 ± 0.24
	Inverse	a	Rel. L_2	3.583556	0.161175	0.315904	3.58 ± 0.16

Inference and timing. Inference applies $x_s \mapsto H = E_{\theta'}(x_s) \mapsto \widehat{r}_h = D_\omega(H, X)$ in one deterministic pass. All timings in Table 2 use an NVIDIA H200 and one input sample producing one dense prediction. Each entry is the arithmetic mean of 200 single-sample inference measurements on the same test set. We use our own JENO implementation and the public implementations and pretrained models provided through the official GitHub repositories of DiffusionPDE and FunDPS (Huang et al., 2024; Yao et al., 2025). Speed ratios compare these single-sample inference times relative to JENO; they do not measure training cost.

B.3.2 Task-Specific Reporting

Only the requested target channel contributes to each main endpoint. Forward error concerns the response f_h, and inverse error concerns the problem-defining field g_h. Darcy inverse uses binary permeability labels rather than a continuous-field relative error. Table 12 summarizes these conventions. Full-pair recovery is reported only in the dedicated diagnostic.

B.3.3 Supplementary Main Results

Table 13 reports the fixed-checkpoint retest on $1{,}000$ held-out examples with five Bernoulli(0.03) mask draws per example. The table retains the original reporting symbols: target u denotes the output-side field and target a denotes the input-side field, corresponding to generic f and g, respectively. This table's a does not denote the complete pair.

For instance i and mask draw r, let e_{ir} be the percentage error. We first average the five draws within each instance, $\bar{e}_i = 5^{-1}\sum_{r=1}^{5} e_{ir}$, then report $\bar{e} = 1000^{-1}\sum_i \bar{e}_i$, $\widehat{\mathrm{SE}} = s(\bar{e}_1, \ldots, \bar{e}_{1000})/\sqrt{1000}$ with sample standard deviation (divisor 999), and normal-approximation 95% intervals $\bar{e} \pm 1.96\widehat{\mathrm{SE}}$. The independent unit is the PDE instance; mask repeats are not treated as additional independent instances. These recomputed intervals describe across-instance uncertainty conditional on the fixed checkpoints and five-mask procedure, not independent training-seed variation. The compact final column is mean$\pm$standard error. NS (BCs) forward rounds to 4.08% in this retest, while its separately reported original main-table estimate remains 4.10%.

Table 14: **Sensitivity-study protocol.** The same data split, optimizer, backbone family, and observation construction are used across variants within each sweep.

Item	Setting
PDEs	Poisson and unbounded Navier–Stokes.
Training data and budget	10,000 examples per PDE; 30 epochs.
Observation ratios	2%, 5%, and 10%.
Context-weight sweep	$\lambda_c \in \{0, 0.1, 0.25, 1.0\}$.
Latent-width sweep	$d_z \in \{16, 32, 64\}$.
Backbone	FNO-style encoder with width 64, depth 6, and 16 Fourier modes.
Pretraining objective	$\lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c$; no dense-field reconstruction loss.
Downstream initialization	Copy the EMA teacher encoder and attach a new decoder.
Adaptation	Jointly fine-tune the copied encoder and decoder.
Tasks and metrics	Forward, inverse, and full recovery; field-wise Relative L_2, averaged over both fields for full recovery.

Table 15: **Sensitivity to context weight and latent dimension.** Each entry reports test Relative L_2 error (%) at observation ratios $2\%/5\%/10\%$. The two sweeps are independent; bold identifies the lowest value at each density within each sweep and task. This diagnostic initializes the downstream model from the EMA teacher encoder, unlike the student-initialized main benchmark. Full recovery observes and predicts both fields.

Group	Variant	Poisson			Navier–Stokes		
		Forward	Inverse	Full recovery	Forward	Inverse	Full recovery
λ_c sweep	0.0	5.36 / 3.18 / 3.59	20.31 / 16.87 / 15.07	10.15 / 7.25 / 5.75	6.22 / **2.98** / **2.13**	13.01 / 10.23 / 8.91	8.55 / 5.33 / **3.91**
	0.1	5.32 / 3.22 / **3.15**	20.36 / 16.88 / 15.05	10.07 / 7.26 / 5.63	6.20 / 3.08 / 2.25	13.01 / 10.26 / 9.00	8.60 / 5.37 / 3.93
	0.25	5.49 / 3.54 / 3.68	20.07 / 16.62 / 15.02	10.07 / **7.11** / 5.65	6.29 / 3.02 / 2.21	**12.98** / 10.22 / 8.95	8.50 / 5.34 / 3.92
	$\lambda_c = 1.0$	**5.10** / **3.14** / 3.18	**19.90** / **16.52** / **14.69**	**9.89** / 7.13 / **5.49**	**6.11** / 3.20 / 2.28	12.99 / **10.08** / **8.86**	**8.47** / **5.30** / 3.92
Latent dim.	16	**5.10** / 3.96 / 3.63	19.67 / 16.42 / 14.43	9.92 / **6.99** / **5.45**	**6.12** / 3.18 / **2.14**	13.16 / 10.43 / 9.29	8.76 / 5.49 / 4.06
	32	5.17 / **3.76** / **2.96**	20.02 / 16.63 / 14.87	10.05 / 7.15 / 5.82	6.29 / **3.02** / 2.21	**12.98** / **10.22** / **8.95**	**8.50** / **5.34** / **3.92**
	64	5.25 / 3.91 / 3.35	**19.21** / **15.92** / **14.22**	**9.78** / 7.06 / 5.67	6.23 / 3.25 / 2.28	**12.98** / 10.33 / 9.04	8.63 / **5.34** / 4.03

B.4 Context-Weight and Latent-Dimension Sensitivity

This study uses 10,000 training examples per PDE and a 30-epoch diagnostic budget on Poisson and unbounded Navier–Stokes. We evaluate observation ratios 2%, 5%, and 10%. Two independent sweeps vary the complement weight $\lambda_c \in \{0, 0.1, 0.25, 1.0\}$ and latent width $d_z \in \{16, 32, 64\}$. The downstream models in this study are initialized from the copied EMA teacher encoder, rather than the student encoder used in the main benchmark. The copied encoder and a new decoder are jointly fine-tuned; no teacher branch is used at inference.

For Poisson, S_{in} selects the source and S_{out} the solution. For NS, they select ω_0 and ω_T. The forward and inverse diagnostic tasks are

$$(m_h \odot S_{\text{in}}(a_h), m_h, S_{\text{in}}, \Xi_h) \mapsto S_{\text{out}}(a_h), \tag{42}$$

$$(m_h \odot S_{\text{out}}(a_h), m_h, S_{\text{out}}, \Xi_h) \mapsto S_{\text{in}}(a_h). \tag{43}$$

Full recovery uses both sparse channels:

$$\left(m_h^{\text{in}} \odot r_{\text{in}}, m_h^{\text{out}} \odot r_{\text{out}}, m_h^{\text{in}}, m_h^{\text{out}}, S_{\text{in}}, S_{\text{out}}, \Xi_h\right) \mapsto (r_{\text{in}}, r_{\text{out}}), \tag{44}$$

where $r_{\text{in}} = S_{\text{in}}(a_h)$ and $r_{\text{out}} = S_{\text{out}}(a_h)$. Its metric is

$$\text{Err}_{\text{full}} = \frac{1}{2}\left(\frac{\|\widehat{r}_{\text{in}} - S_{\text{in}}(a_h)\|_2}{\|S_{\text{in}}(a_h)\|_2} + \frac{\|\widehat{r}_{\text{out}} - S_{\text{out}}(a_h)\|_2}{\|S_{\text{out}}(a_h)\|_2}\right). \tag{45}$$

All three diagnostic task errors are reported as percentages. Table 14 summarizes the protocol, and Table 15 reports the two sweeps.

The effects depend on the task and observation density. At 5% observations, increasing λ_c from 0 to 1 changes Poisson inverse error from 16.87% to 16.52%, but NS forward error from 2.98% to 3.20%.

No tested setting is uniformly best. The two sweeps are independent runs and should be interpreted within their respective table groups. In particular, $\lambda_c = 0$ retains the teacher and sampled-target latent prediction; it is not a model trained without predictive pretraining. Appendix C.7 explains the distinction between changing this weight and replacing the spatially separated objective.

C Additional Sparse-Observation Experiments and Clarifications

This appendix examines backbone dependence, sparse-input training, observation density, measurement noise, characteristic spatial scales, representation alignment, and downstream readout choices. Each study specifies its own checkpoint and observation protocol. Complete fields provide training supervision, while field predictions use only the stated sparse inputs. Test-example, mask, and perturbation variability are distinguished from variability across independently trained models.

C.1 Matched Three-Percent Controls and Backbone Dependence

Table 3 compares JENO-FNO, JENO-CNO, JENO-U-Net, and a non-convolutional DeepONet on Poisson forward and unbounded NS inverse. All four use the same Partial-OBS dataset, split, and metric, with matched nominal 3% downstream training and test densities. Main JENO-FNO pretraining uses mixed densities (Appendix B.3.1); matching here does not imply identical realized masks, model sizes, or total optimization budgets. The CNO and U-Net variants are trained from scratch within the JENO framework. The DeepONet branch is an MLP that encodes sparse sensor values and locations, and its MLP trunk encodes output query coordinates. This control does not construct a zero-filled grid or use a convolutional branch.

The CNO and U-Net variants use width 64, seed 42, 100 pretraining epochs and 20 joint encoder–readout fine-tuning epochs. Both stages use AdamW with constant learning rate 2×10^{-4}, weight decay 10^{-4}, and batch size 8. Their masks select exactly 492 uniformly sampled points in both stages and at test time. Their pretraining objective is mean squared prediction error over the final teacher feature grid, with teacher-encoder EMA 0.996; it does not use the multilevel spatial weighting of the main FNO run. Fine-tuning uses mean target-channel L_1 loss in model-normalized coordinates, and evaluation uses the terminal models on the first $1{,}000$ test records. Thus, these results test alternative backbone variants of the two-stage procedure rather than an otherwise identical substitution into the main configuration.

JENO-FNO has the lowest Poisson forward error among the three JENO backbones at 1.80%, followed by CNO at 2.04% and U-Net at 2.45%. On NS inverse, CNO and U-Net obtain 6.29% and 6.33%, compared with FNO's 7.36%. The sensor-coordinate DeepONet obtains 6.29% and 28.92% on the two tasks. These results support the use of the JENO procedure with all three tested backbones, without identifying one backbone as uniformly best. Because the architectures and training procedures differ, this is a comparison of complete models rather than an isolated pretraining-objective ablation.

C.2 Sparse-Trained FNO and Convolutional DeepONet Controls

The controls in Table 16 use the same training samples as JENO and receive sparse field values, observation masks, and coordinates. FNO and convolutional DeepONet are trained at 2% observations and evaluated at both 2% and 5% without changing their weights. JENO uses the 3%-fine-tuned model from the density sweep in Appendix C.3. The 5% evaluation is therefore a cross-density test, and the methods do not have matched training densities.

JENO has lower error in all four reported comparisons. FNO's increase in error at 5% occurs under a change in the input value-and-mask distribution relative to training; it does not establish that additional observations are intrinsically harmful. The convolutional DeepONet here is distinct from the MLP sensor-branch model in Table 3. These experiments show the performance of sparse-trained models under the stated protocols rather than isolating the effect of one loss component.

C.3 Observation-Density Sensitivity at Fixed Spatial Resolution

We keep the spatial grid at 128×128 and vary the observed fraction from 1% to 100%. We use the main JENO terminal checkpoints from Table 11, fine-tuned at 3% observations. Only the test

Table 16: **Sparse-trained deterministic controls.** Mean Relative L_2 error (%). FNO and convolutional DeepONet train at 2% observations; JENO fine-tunes at 3%. Bold marks the lowest error in each row.

Task	Test density	JENO	FNO	Conv. DeepONet
Poisson forward	2%	**2.70**	5.41	29.93
Poisson forward	5%	**1.31**	7.04	26.68
NS inverse	2%	**6.93**	12.04	62.23
NS inverse	5%	**4.79**	12.77	61.08

Table 17: **Cross-density evaluation of fixed JENO models.** Test Relative L_2 error (%), reported as mean±standard deviation. Spatial resolution is 128×128, and downstream training density is 3%.

Test density	Poisson forward	Unbounded NS inverse
1%	5.16 ± 2.96	11.12 ± 2.34
2%	2.70 ± 1.46	6.93 ± 1.35
5%	1.31 ± 0.74	4.79 ± 0.89
10%	0.82 ± 0.43	4.17 ± 0.77
20%	0.50 ± 0.21	3.81 ± 0.68
50%	0.57 ± 0.17	4.28 ± 0.78
100%	0.59 ± 0.25	2.77 ± 0.76

observation masks change; model weights remain fixed without additional training. Table 17 reports test-error means and standard deviations; the deviations do not represent independent training-seed variation.

Both tasks improve substantially between 1% and 20% observations. Poisson forward error decreases from 5.16% to 0.50%, and NS inverse error decreases from 11.12% to 3.81%. The trend is not monotonic: both errors increase between 20% and 50%, and the full-observation NS error subsequently decreases to 2.77%. The experiment measures transfer across observation densities at one spatial resolution. Its 100% endpoint is not a comparison against an independently trained full-observation operator, and it does not test higher-resolution, irregular-domain, or three-dimensional inference.

C.4 Observation-Noise Sensitivity

Separate checkpoint recipe. The archived paired-task run uses seed 42, 200 pretraining epochs at 3% Bernoulli observations, and forward, inverse, and joint-recovery tasks cycling in ten-epoch blocks. Its objective is $\mathcal{L}_m + 0.25\mathcal{L}_c + \mathcal{L}_{\text{field}}$, where $\mathcal{L}_{\text{field}}$ is mean paired-field L_1 loss in model-normalized coordinates. This auxiliary field term belongs to this diagnostic run, not the latent-only main pretraining. The student encoder from the pretraining checkpoint with lowest validation pair Relative L_2 initializes each task model, which then receives 200 epochs of target-channel L_1 fine-tuning. AdamW uses weight decay 10^{-4}, initial learning rates 3×10^{-4}/2×10^{-4} for pretraining/fine-tuning, and separate 200-epoch cosine schedules to zero, with automatic mixed precision. Batch size is 8 except 16 for NS (BCs). The diagnostics below use the terminal fine-tuning checkpoints; the main benchmark retains the separate terminal-checkpoint procedure in Appendix B.3.1.

We evaluate task-specific epoch-200 JENO checkpoints from the separate paired-task run on 1,000 held-out instances without retraining. For each instance, the observation mask selects exactly 492 of the 128^2 grid locations uniformly without replacement. Gaussian noise is added to the model-normalized observed values, with standard deviation equal to 0%, 1%, or 5% of $\text{RMS}_{\text{obs}} = \max\{\sqrt{K^{-1}\sum_{i:m_i=1} x_i^2}, 10^{-12}\}$, where x is the observed channel after training-slice normalization and $K = 492$. Only sampled positions enter this RMS, with no additional mean subtraction. The same mask is retained across noise levels for each instance and perturbation seed. Seeds 0–4 control observation perturbations, not independent model training.

Table 18: **Observation-noise sensitivity.** Errors are percentages averaged over five perturbation seeds. E_0, E_1, and E_5 denote errors at 0%, 1%, and 5% noise; A_1 and A_5 are the corresponding amplification ratios. Darcy inverse uses BER; all other endpoints use Relative L_2.

PDE	Direction	E_0	E_1	E_5	A_1	A_5
Darcy	Forward	4.073	4.073	4.080	1.000	1.002
	Inverse	2.098	2.288	7.933	1.091	3.781
Poisson	Forward	1.590	1.606	1.924	1.010	1.210
	Inverse	6.715	21.393	44.196	3.186	6.582
Helmholtz	Forward	1.629	1.644	1.938	1.009	1.190
	Inverse	6.549	23.501	49.200	3.589	7.513
Unbounded NS	Forward	2.509	2.542	3.201	1.013	1.276
	Inverse	5.448	12.528	24.290	2.299	4.458
NS (BCs)	Forward	4.004	4.014	4.208	1.002	1.051
	Inverse	3.046	3.054	3.252	1.003	1.068

Let E_η denote the mean prediction error at noise level η, and define amplification relative to the matched clean input as

$$A_\eta = \frac{E_\eta}{E_0}. \tag{46}$$

The ratios are computed from unrounded means. Table 18 uses Relative L_2 for all tasks except Darcy inverse, which uses BER. The separate checkpoint and fixed-count mask protocol distinguish this study from the main repeated-mask retest. Its clean errors therefore do not replace the main-table entries.

The strongest amplification occurs in inverse Poisson, Helmholtz, and unbounded NS. At 1% noise their errors increase by $3.186\times$, $3.589\times$, and $2.299\times$, whereas the corresponding forward ratios are $1.010\times$, $1.009\times$, and $1.013\times$. At 5% noise the inverse ratios reach $6.582\times$, $7.513\times$, and $4.458\times$. NS (BCs) changes less in both directions. These results characterize task-dependent sensitivity of the trained predictors, rather than an architecture-independent PDE condition number or evidence of dynamical chaos.

C.5 Points per Characteristic Wavelength

The points-per-characteristic-wavelength diagnostic examines whether errors are associated with how well the observations resolve a field's spatial scales. We use the reported per-instance statistic PPW_{90} at the matched 3% observation setting. For each of the $1{,}000$ test instances, its statistic is paired with the prediction error of each evaluated model. The correlation ρ_{PPW} summarizes the association between these two quantities within a task.

For each complete observed-side field (Poisson source or terminal NS vorticity), we subtract its spatial mean, apply a separable Hann window for nonperiodic Poisson and no window for periodic NS, and compute the two-dimensional FFT in float64. Let f_{90} be the smallest radial frequency, in cycles per grid cell, whose cumulative non-DC power reaches 90% of the total non-DC power. Then $\lambda_{90} = 1/f_{90}$, the effective sensor spacing is $\Delta_s = \sqrt{HW/K}$ for the realized sensor count K, and $\mathrm{PPW}_{90} = \lambda_{90}/\Delta_s$. Non-finite or at most 10^{-20} total power, or $K = 0$, gives an undefined statistic and is excluded. We use Spearman rank correlation and the two-sided `scipy.stats.spearmanr` test, with one error/statistic pair per instance, not repeated masks treated as independent observations. The archived JENO runs use one fixed-count 492-point mask per test instance. The FNO correlations use the separate paired-task epoch-200 checkpoints from Appendix C.4; they are not recomputed from the main-model errors in Table 1.

A negative correlation means that instances with fewer effective observations per characteristic wavelength tend to have larger prediction errors. The statistic is used to interpret variation in recovery difficulty across instances, rather than to rank models by prediction accuracy. Table 19 retains the reported correlations for the model families in Table 3; all reported p-values are below 3×10^{-50}.

Table 19: **Characteristic-scale diagnostic.** Spearman correlation ρ_{PPW} between PPW_{90} and prediction error on $1{,}000$ test instances under matched 3% observations.

Method	Poisson forward	Unbounded NS inverse
JENO-FNO	-0.632	-0.578
JENO-CNO	-0.447	-0.582
JENO-U-Net	-0.457	-0.635
DeepONet (MLP sensor branch)	-0.573	-0.539

Table 20: **Within-model representation endpoints.** Each cell reports normalized mismatch / CKA against the model's own full-view representation. Mismatch is scaled by the reference latent norm and the model's 1% value. The tasks are Poisson forward and unbounded NS inverse.

Task	Model	1%	2%	5%	10%	20%	50%	100%
Poisson	JENO	1.00/.97	1.01/.96	.98/.96	.91/.94	.79/.91	.34/.98	.00/1.00
	FNO	1.00/.53	.91/.55	.93/.56	.87/.64	.52/.78	.15/.95	.00/1.00
	Conv. DeepONet	1.00/.91	.71/.94	.43/.96	.30/.97	.22/.98	.10/.99	.00/1.00
Unbounded NS	JENO	1.00/.99	.96/1.00	.98/1.00	.88/1.00	.88/.97	.57/1.00	.00/1.00
	FNO	1.00/.53	1.00/.54	1.00/.56	.98/.62	.93/.79	.56/.96	.00/N/A
	Conv. DeepONet	1.00/.82	.72/.87	.46/.92	.32/.94	.23/.96	.10/.98	.00/1.00

The correlations are negative for every evaluated model in both tasks. This is consistent with spatial under-resolution contributing to sparse-recovery difficulty, including on the elliptic Poisson problem. The association does not imply that JENO recovers information absent from the observations, nor does it identify the causal contribution of predictive pretraining.

C.6 Within-Model Representation Endpoint Check

This study compares each model's representation of a partially observed input with its own representation of the corresponding complete input. FNO and convolutional DeepONet are trained on sparse values, masks, and coordinates, unlike the complete-input-trained controls in Appendix B.2. We evaluate densities from 1% to 100% with fixed model weights.

Mismatch is normalized by the full-view reference latent norm and then by that model's own 1% mismatch. The resulting values describe relative changes within a model, not a shared cross-architecture risk. The CKA comparison pairs the partial-view and full-view feature matrices; its centered definition is given in Appendix B.2.3. Each entry in Table 20 reports normalized mismatch followed by CKA.

At full observation the two inputs coincide, so their mismatch vanishes by construction. CKA equals one when the centered features are non-degenerate. For FNO on unbounded NS at 100%, the archived result is N/A; this alone does not establish that the mathematical denominator is exactly zero. Since the full-view reference is fixed across densities, a truly zero reference norm would make every density undefined. We therefore do not interpret this endpoint as evidence of feature collapse. These endpoint properties do not imply zero field-prediction error. Likewise, a decreasing within-model mismatch need not imply competitive downstream accuracy. Intermediate-density changes are empirical observations, not a monotonicity guarantee.

C.7 Mask-Aware Supervision and Component Interpretation

What the masks specify. The target mask M selects sampled unobserved positions. The weighted complement $C = (1 - M)\gamma$ covers all remaining positions, including both observed and unobserved locations. Both losses compare predicted latent vectors with complete-field teacher targets. The masks therefore specify the spatial allocation of supervision, not whether a latent feature contains exclusively observed or exclusively unobserved physical information. A feature at any grid position can aggregate information from other locations through the encoder.

Table 21: **Downstream readout sensitivity.** Mean test Relative L_2 error (%) after equal 20-epoch joint encoder–readout fine-tuning budgets. Bold identifies the lowest value in each task.

Readout	Poisson forward	Unbounded NS inverse
Conv-1 head	1.92	6.24
Standard FNO decoder	**1.74**	5.77
MLP-2 head	1.83	6.10
FNO-1 head	1.76	5.50
FNO-3 head	1.87	**5.24**

Why the two terms are separate. The sampled-target and complement terms have different supports and are normalized by their own total weights. Their coefficients control the relative contributions of these regions despite differences in region size. The complement loss is latent prediction, not a constraint that the field output exactly reproduce measured values. The tensor shapes, mask construction, and batch reductions are specified in Appendix B.3.1.

What the context-weight sweep tests. Changing λ_c preserves the sparse-input student, complete-field teacher, predictor, and sampled-target objective. Setting $\lambda_c = 0$ removes only the complement contribution. In contrast, uniform full-grid latent alignment would replace the separately normalized spatial objective, while training without predictive pretraining would remove an entire learning stage. These are different comparisons. The results in Table 15 establish sensitivity to the weight within the present construction; they do not quantify an advantage over uniform alignment or isolate the effect of pretraining itself.

C.8 Downstream Readout Swap

For each task, all five variants inherit the same task-specific epoch-200 encoder from the recipe in Appendix C.4 and attach a new readout. The copied encoder and readout are jointly fine-tuned for 20 epochs with seed 42, batch size 8, and AdamW with learning rate 2×10^{-4} and weight decay 10^{-4}. The training objective is target-channel L_1 loss, averaged over batch and grid in model-normalized coordinates. All readouts take width-64 encoder features and produce one target channel. Conv-1 is one 1×1 convolution ($64 \to 1$); MLP-2 is 1×1 ($64 \to 64$), GELU, 1×1 ($64 \to 1$). The standard decoder is the convolutional head in Appendix B.3.1, with one output channel. FNO-1/FNO-3 apply one/three width-64, 16×16-mode FNO blocks, followed by one-group GroupNorm and the MLP-2 projection. Training and test masks select exactly 492 points uniformly without replacement. We evaluate the terminal models on 1,000 test pairs with one mask per pair, without validation-based checkpoint selection.

The preferred readout differs between tasks. The standard FNO decoder gives the lowest Poisson forward error at 1.74%, while the FNO-3 head gives the lowest NS inverse error at 5.24%. Across all readouts, the respective ranges are 1.74–1.92% and 5.24–6.24%. Shared initialization and equal fine-tuning budgets support comparisons within this study. The main benchmark uses the separate schedules in Table 11; these results neither replace the main entries nor isolate the pretraining objective.

C.9 Scope of the Representation-Learning Comparison

Training information and deployment. JENO uses the paired complete PDE fields supplied by the Partial-OBS training data. During pretraining, the student receives sparse measurements and observation information, while the teacher encodes the complete pair. Dense targets subsequently supervise the copied student encoder and task readout. At deployment, only the adapted encoder and readout are used. Compared methods also have access to complete training fields, using them as supervised targets or to learn field distributions. The distinction is how these resources are used, not additional information available to JENO at test time. Equal data access does not imply equal architectures or optimization budgets.

Observation representations. Grid-based JENO receives zero-filled measurements, masks, and coordinates. The mask preserves the distinction between missing values and measured zeros. Zero filling does not reconstruct a complete physical input, nor does it by itself establish an unavoidable

prediction-error penalty. The non-convolutional DeepONet control encodes sensor values and locations directly and evaluates outputs through a coordinate trunk. It is distinct from the convolutional grid-input DeepONet used in the other controls, and its results should be interpreted at the level of the evaluated implementation.

Relation to partial-observation operators. LANO learns from partially observed training trajectories using mask-to-predict supervision and latent autoregressive propagation (Hou et al., 2026). Our setting supplies complete training pairs and recovers one dense field from sparse observations of the other, including initial-to-terminal and terminal-to-initial state prediction. CORAL uses coordinate-based neural fields for operator learning on general geometries (Serrano et al., 2023). These approaches are relevant alternatives, but their published results are not matched numerical comparisons for the present tasks and protocols.

Physical uncertainty and theoretical scope. Equation (20) separates irreducible physical uncertainty from excess squared prediction risk. The latter can include representation, model-capacity, data, and optimization effects. JENO investigates complete-field latent prediction as a way to improve the learned predictor. The fixed-checkpoint analysis in Sec. 3.2 relates the normalized objective to latent alignment and then to field error through explicit teacher-decoding, scale, representation-transfer, and fine-tuning residuals. It does not imply that latent agreement alone resolves non-identifiability or guarantees accurate field recovery.

Guidance approximation in conditional sampling. For a finite-grid illustration, let a_0 be a complete field pair, a_t its diffusion-corrupted version, and $\ell(a_0) = p(u \mid a_0, I)$ a positive, regular measurement likelihood. If u and a_t are conditionally independent given (a_0, I), then

$$p(u \mid a_t, I) = \mathbb{E}[\ell(a_0) \mid a_t, I].$$

A DPS-style approximation evaluates the likelihood at a denoised estimate $\bar{a}_0 = \mathbb{E}[a_0 \mid a_t, I]$ instead of averaging over the conditional distribution (Chung et al., 2023). The guidance score is consequently approximated by

$$\nabla_{a_t} \log \mathbb{E}[\ell(a_0) \mid a_t, I] \;\approx\; \nabla_{a_t} \log \ell(\bar{a}_0).$$

This likelihood approximation can introduce method-dependent error, but does not establish a universal field-error ordering between deterministic and generative methods. The illustration assumes a regular likelihood, not an ordinary density for exact noiseless constraints. JENO avoids iterative sampling while returning a point prediction rather than a posterior distribution.

What the empirical controls establish. The main comparison evaluates the complete pretraining and adaptation procedure. Backbone controls examine its use with FNO, CNO, and U-Net; sparse-trained controls examine observation-aware deterministic prediction; and readout swaps examine downstream architectural sensitivity. The context-weight sweep studies one coefficient within the spatially separated latent objective. These experiments answer complementary questions but do not independently identify the effects of every loss term, model component, or optimization budget.

Broader operator-learning context. Related work improves PDE data generation through Krylov subspace recycling (Wang et al., 2024), accelerates linear-system solving with learned Krylov iterations (Luo et al., 2024), and studies adaptive operator transformations for large-scale PDE pretraining (Lv et al., 2026). These directions address numerical efficiency or operator pretraining more broadly, whereas JENO focuses on predictive representation learning from sparse inputs.

Practical scope. A motivating application is repeated sparse-snapshot reconstruction after offline training on complete simulations. JENO can serve as a fast reconstruction module in a larger data-assimilation system, but is not itself a sequential forecast–analysis algorithm (Asch et al., 2016). The present evaluation concerns structured 128×128 fields and the stated observation protocols; it does not establish correction of simulator bias, adaptation to environmental changes, or calibrated posterior uncertainty.