DyPSI: Dynamic Physics Sensing via Joint Field and Sensor-Trajectory Generation

Yizhou Zhang[2] **Panqi Chen**[1] **Lei Cheng**[1] **Ting Zhang**[1] **Jianlong Li**[1]
Jianfeng Shi[3] **Shikai Fang**[1*]

[1]College of Information Science and Electronic Engineering, Zhejiang University
[2]School of Mathematical Sciences, Ocean University of China
[3]College of Energy Engineering, Zhejiang University

Abstract

Physics sensing in evolving environments requires reconstructing dense spatiotemporal fields from sparse observations while adapting where future observations should be collected. Existing reconstruction and sensor-placement methods have made substantial progress, but they typically rely on static sensing assumptions: they either use a single layout across all frames or optimize placements independently at each frame, which limits their ability to produce temporally coherent sensing trajectories. We introduce *DyPSI*, a dynamic physics sensing framework that formulates this problem as the joint generation of future fields and sensor trajectories conditioned on historical observations. DyPSI lifts discrete sensor coordinates into continuous Sensor Position Fields, encodes fields and sensing configurations with a shared Functional Tucker representation, and trains a joint diffusion model to generate future fields and sensing trajectories simultaneously in the resulting latent space. A Gaussian-process temporal kernel correlates the perturbation injected during diffusion training, biasing the denoiser toward temporally smooth recoveries, and we further provide an analysis linking the EDM training objective to a sequence-aggregated A-optimality criterion and a kernel-induced bound on discrete trajectory variation. Experiments on turbulent flow, GLORYS12 sea-surface temperature, and 3D car aerodynamics show that DyPSI consistently outperforms static and frame-wise placement strategies, with substantial error reduction under tight sensor budgets. The code is available at `https://github.com/mengqi132/DyPSI`.

1 Introduction

Physics sensing recovers dense spatiotemporal fields from sparse sensor observations and underpins applications in fluid dynamics [12, 11, 44], climate monitoring [35, 34], and industrial inspection [19, 41]. In many realistic deployments, sensing resources are both sparse and *mobile*: autonomous underwater vehicles [30, 42], drifting buoys [22, 8], and aerial platforms [3, 26] reposition over time to track evolving phenomena [39]. Unlike static deployments, mobile sensing must continually reallocate observations to follow drifting vortices [40], propagating fronts [32, 13], and translating surface-pressure peaks [43]. Effective sensing therefore requires jointly reconstructing fields and steering sensors, since reconstruction depends on past sensor locations while informed motion depends on predicted field evolution.

Recent learning-based methods couple reconstruction with sensor-placement optimization, exemplified by PhySense [27], which jointly trains a reconstructor and a learnable placement under a shared loss. However, such placement either optimizes a single global layout shared across frames

*Correspond to fsk@zju.edu.cn

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

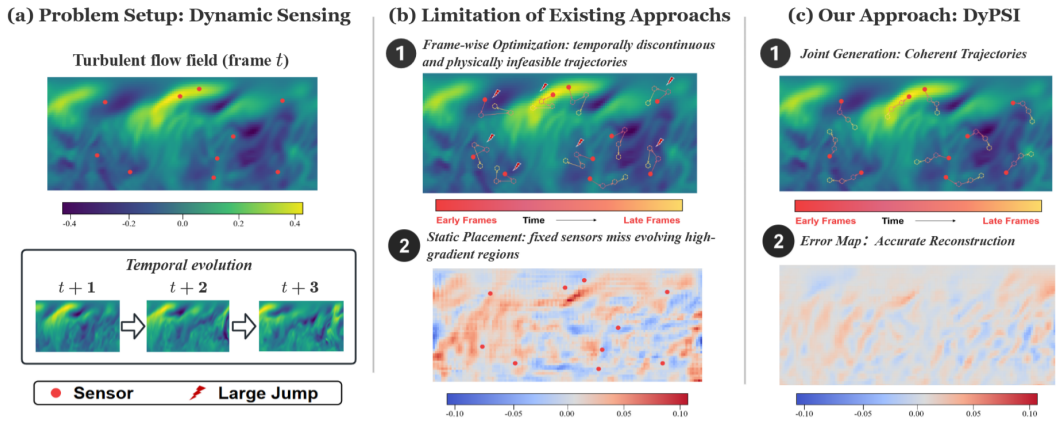

Figure 1: Dynamic active sensing. (a) Mobile sensors (red) traverse a time-evolving turbulent flow and must continuously reposition to capture informative regions. (b) Static placement is fixed across frames and misses migrating high-gradient structures; per-frame optimization is spatially adaptive but produces temporally discontinuous, kinematically infeasible jumps. (c) DyPSI generates temporally coherent sensor trajectories *jointly* with the reconstructed field, attaining the lowest residual error.

or optimizes each frame independently. The former cannot follow migrating structures, while the latter produces temporally discontinuous "teleporting" trajectories. Neither ensures that the sensor at time $t+1$ remains within reach of its position at time t, decoupling the sensing strategy from the sequential dynamics and platform kinematics.

Bridging these regimes poses three coupled challenges: a *representation mismatch* between continuous high-dimensional fields and discrete coordinate sequences; a *bidirectional coupling* in which future sensing depends on predicted dynamics while prediction depends on past sensing; and a *temporal-consistency gap* in which per-frame optima may be globally suboptimal or kinematically infeasible. Figure 1(a)–(b) illustrates how static layouts miss migrating structures while per-frame optimization produces jumps that real platforms cannot execute.

To address these challenges, we propose *DyPSI*, a framework for joint conditional generation of future fields and sensor trajectories. We lift discrete sensor coordinates into continuous *Sensor Position Fields*, encode fields and configurations in a shared Functional Tucker latent, and train a joint diffusion model to generate both simultaneously. Replacing the i.i.d. noise of standard EDM with Gaussian-process-correlated perturbations further encourages temporally smooth trajectories. We provide population-level analyses of sensing consistency and kinematic feasibility and validate DyPSI on three real-world physics-sensing benchmarks. As shown in Figure 1(c), DyPSI tracks migrating informative structures with temporally coherent, reachable trajectories while accurately reconstructing the field.

Contributions **(i) Formulation:** we recast dynamic active sensing as joint conditional generation over future fields and sensor trajectories, replacing static and per-frame placement with sequence-level joint modeling. **(ii) Method:** we introduce continuous Sensor Position Fields, a shared Functional Tucker latent, and a joint diffusion model with Gaussian-process-correlated perturbations for sequence-aware generation. **(iii) Theory and empirics:** we provide complementary sensing-consistency and kinematic-feasibility analyses and show that DyPSI consistently outperforms static and per-frame placement baselines on three real-world benchmarks.

2 Preliminary

This section recalls three ingredients reused throughout the paper: the Functional Tucker representation that maps fields and sensing configurations into a shared continuous-coordinate latent (§3.2), the Elucidated Diffusion Model (EDM) denoising objective that we instantiate on the joint latent (§3.3), and the A-optimality criterion that we extend to a sequence-aggregated form and connect to the EDM objective in §3.4.

2.1 Functional Tucker Representation

For continuous coordinates $\mathbf{z} = (z_1, \ldots, z_K) \in \Omega_1 \times \cdots \times \Omega_K$ and ranks $(R_1, \ldots, R_K)$, Functional Tucker representation [25, 9, 23] approximates a multivariate function $f(\cdot) : \mathbb{R}^K \to \mathbb{R}$ with Tucker decomposition [37] form:

$$f(\mathbf{z}) \approx \sum_{r_1,\ldots,r_K} \mathcal{W}_{r_1,\ldots,r_K} \prod_{k=1}^{K} \big[\boldsymbol{\phi}_k(z_k)\big]_{r_k} = \big\langle \mathrm{vec}(\mathcal{W}),\, \boldsymbol{\phi}_1(z_1) \otimes \cdots \otimes \boldsymbol{\phi}_K(z_K) \big\rangle, \tag{1}$$

where $\mathcal{W} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ is the core tensor and each $\boldsymbol{\phi}_k : \Omega_k \to \mathbb{R}^{R_k}$ is a vector-valued basis (a coordinate MLP, following SDIFT [5]). The Kronecker form on the right makes explicit that f is a linear combination of products of single-mode basis evaluations; per-mode ranks R_k are independent and can be chosen separately for each modality, enabling the shared spatial bases with modality-specific channel modes used in §3.2.

2.2 EDM Denoising in Latent Space

We adopt the Elucidated Diffusion Model (EDM) formulation of Karras et al. [17]. Given a clean latent $\mathbf{x}$, a noise level $\sigma \sim p(\sigma)$, and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, EDM trains a σ-conditioned denoiser D_θ via the squared-error objective

$$\mathcal{L}_{\mathrm{EDM}} = \mathbb{E}_{\mathbf{x},\sigma,\mathbf{n}} \Big[\lambda(\sigma) \, \big\| D_\theta(\mathbf{x} + \sigma\mathbf{n}; \sigma) - \mathbf{x} \big\|_2^2 \Big], \tag{2}$$

with noise-dependent weighting $\lambda(\sigma)$. Input/output preconditioning normalizes the denoiser's signal-to-noise ratio across σ, so a single network covers a wide range of noise scales — a property we exploit in §3.3 when the i.i.d. Gaussian draw $\mathbf{n}$ is replaced by a temporally correlated one without altering the loss form. At the population optimum, the Bayes-optimal denoiser is the conditional mean $D^*(\mathbf{x} + \sigma\mathbf{n}; \sigma) = \mathbb{E}[\mathbf{x} \mid \mathbf{x} + \sigma\mathbf{n}]$, and its risk equals the weighted trace of the posterior covariance of $\mathbf{x}$ given the noisy observation; this identity is the lever connecting EDM training to the sequence-aggregated uncertainty criterion analysed in §3.4.

2.3 A-Optimality for Sensing Uncertainty

A-optimality is the classical experimental-design criterion [33]: for a sensing configuration $\mathbf{s}_t$ inducing latent-field posterior covariance $\boldsymbol{\Sigma}_{\mathbf{s}_t}$, the single-step criterion is the trace of the posterior covariance,

$$\mathcal{L}_A^t := \mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{s}_t}). \tag{3}$$

We adopt the trace criterion (rather than D-optimality's $\log\det$ or E-optimality's $\lambda_{\max}$) because it is, up to a constant, the average per-coordinate posterior variance and is therefore directly interpretable as expected squared reconstruction error — the criterion most naturally aligned with the squared-error form of Eq. (2). Since dynamic sensing accumulates uncertainty over a *trajectory* rather than a single layout, we work with the sequence-aggregated form

$$\mathcal{L}_A^{\mathrm{seq}} := \sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{\Sigma}_{\mathbf{s}_t}), \tag{4}$$

which scores an entire trajectory by its cumulative posterior uncertainty, treating per-step posteriors independently and acting as an upper bound on any criterion that accounts for cross-time covariances. §3.4 and Appendix A show that the EDM training risk in Eq. (2) controls Eq. (4) up to constant factors under standard Gaussian-posterior assumptions.

3 Methodology

DyPSI generates future fields and sensor trajectories jointly from historical observations. We address the representation mismatch between physical fields and sensor trajectories with a functional Tucker latent (§3.2), build a sequence-aware joint diffusion model on top (§3.3), and ground both choices with two complementary population-level results (§3.4).

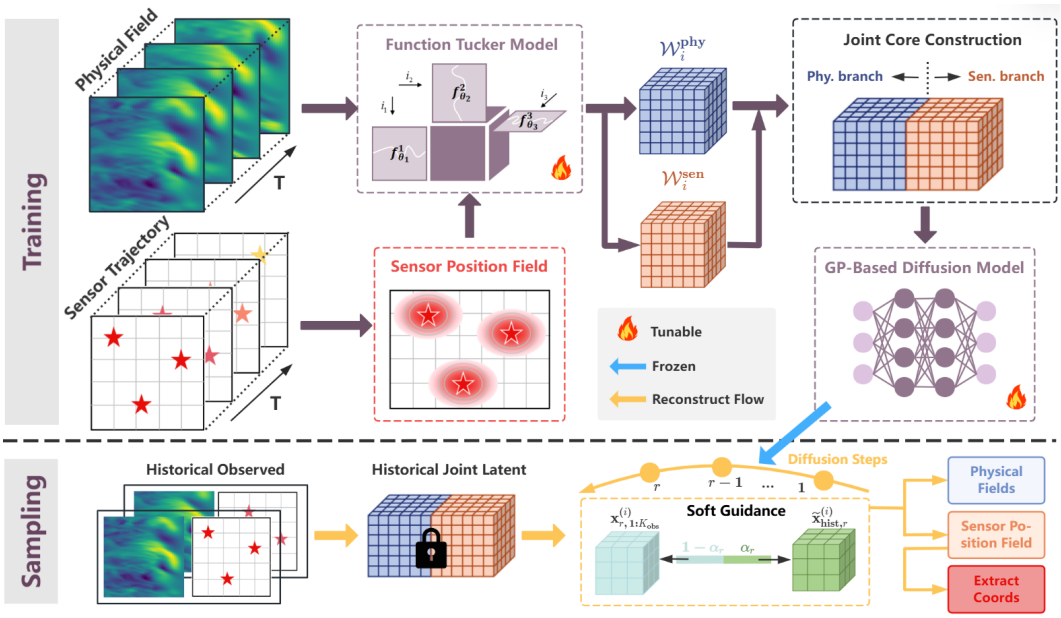

Figure 2: Overview of DyPSI. DyPSI jointly generates future physical fields and adaptive sensor trajectories via temporally correlated latent diffusion conditioned on historical observations.

3.1 Problem Formulation

A physical field $\mathcal{P}_t : \Omega \to \mathbb{R}^C$ evolves over $\Omega \subset \mathbb{R}^D$ for T timesteps. At each time t, k mobile sensors with coordinates $\mathbf{s}_t = \{(x_t^i, y_t^i)\}_{i=1}^k \in \Omega^k$ produce noisy point observations

$$\mathbf{y}_t = \left[\mathcal{P}_t(x_t^1, y_t^1), \ldots, \mathcal{P}_t(x_t^k, y_t^k)\right]^\top + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{obs}}^2 \mathbf{I}). \tag{5}$$

For notational simplicity, we use 2D sensor coordinates here, and the formulation directly extends to general D-dimensional domains by replacing (x_t^i, y_t^i) with $\mathbf{r}_t^i \in \Omega \subset \mathbb{R}^D$.

Given the history $\mathcal{H}_t = \{(\mathbf{s}_\tau, \mathbf{y}_\tau)\}_{\tau=1}^t$, DyPSI models the joint conditional $p(\mathcal{P}_{t+1:T}, \mathbf{s}_{t+1:T} \mid \mathcal{H}_t)$, in contrast to the two common placement abstractions:

$$\begin{aligned} \text{Static (e.g., PhySense-opt):} \quad & \mathbf{s}^\star = \arg\min_{\mathbf{s}\in\Omega^k} \mathbb{E}_t\big[L(\mathcal{P}_t, \mathbf{s})\big], \\ \text{Per-frame:} \quad & \mathbf{s}_t^\star = \arg\min_{\mathbf{s}\in\Omega^k} L(\mathcal{P}_t, \mathbf{s}), \quad \forall t, \\ \text{Joint (DyPSI):} \quad & p\big(\mathcal{P}_{t+1:T},\ \mathbf{s}_{t+1:T} \ \big|\ \mathcal{H}_t\big)\,, \end{aligned} \tag{6}$$

where L is a per-frame reconstruction loss. The static formulation cannot track evolving structures; the per-frame formulation is spatially adaptive but allows $\|\mathbf{s}_{t+1}^\star - \mathbf{s}_t^\star\|$ to grow without bound, violating mobile-platform kinematics. The joint formulation places field evolution and sensor motion under a single distribution, so any temporal correlation in the model regularizes both simultaneously. It also departs from action-level control: the output is a distribution over future *sequences* rather than a deterministic per-step policy.

3.2 Joint Representation of Fields and Sensor Trajectories

The two objects in Eq. (6) have incompatible native types: $\mathcal{P}_t$ is a continuous high-dimensional function on Ω, while $\mathbf{s}_t$ is a sparse, discrete set of coordinates. To operate on both with one generative model, we need to reconcile these types in a common space while preserving the spatial geometry they share. We do this in two steps. First, we lift the sensor configuration into a continuous surrogate on Ω via a parameter-free Gaussian density (*Sensor Position Field* below). Second, we encode this surrogate together with the physical field through a Functional Tucker decomposition that shares its spatial bases between the two modalities, so that a single denoiser can attend to either modality in a single latent without architectural duplication.

Sensor Position Fields For each $\mathbf{s}_t$, we define a Sensor Position Field $\mathcal{S}_t : \Omega \to \mathbb{R}_+$ by

$$\mathcal{S}_t(x, y) = \sum_{i=1}^{k} \exp\Big(-\tfrac{(x-x_t^i)^2}{2\sigma_x^2} - \tfrac{(y-y_t^i)^2}{2\sigma_y^2} \Big), \tag{7}$$

with the analogous Gaussian-product extension in 3D. We treat the SPF as a differentiable spatial surrogate, not as a new physical variable; it lets us place sensing configurations and physical fields on the same domain so that a single Functional Tucker basis can encode both. The map is bidirectional: Eq. (7) gives $\mathbf{s}_t \to \mathcal{S}_t$ in closed form, and at inference we recover $\mathbf{s}_t$ from $\mathcal{S}_t$ by a top-k $\arg\max$ readout. We invoke this readout only at sampling; during training, supervision flows through the SPF latent reconstruction itself, which is fully differentiable. We defer bandwidth selection and the mesh extension to Appendix C.2.

Joint Tucker latent We partition time into segments of length T_{seg}, and for each segment i apply Eq. (1) to the field and SPF sequences, producing

$$\mathcal{W}_i^{\text{phy}} \in \mathbb{R}^{T_{\text{seg}} \times R_1^{\text{phy}} \times R_2 \times \cdots \times R_{D+1}}, \quad \mathcal{W}_i^{\text{sen}} \in \mathbb{R}^{T_{\text{seg}} \times R_1^{\text{sen}} \times R_2 \times \cdots \times R_{D+1}}. \tag{8}$$

The encoder is *amortized*: $\boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{D+1}$ are trained once on the entire dataset, jointly with sample-specific cores, by reconstruction against arbitrary continuous-coordinate observations of $\mathcal{P}_t$ and $\mathcal{S}_t$. Once trained, these spatial bases are *shared across both modalities and across all samples and timesteps* — justified by the fact that $\mathcal{S}_t$ and $\mathcal{P}_t$ live on the same Ω — with only the channel-mode rank R_1 differing between $\mathcal{P}$ and $\mathcal{S}$. The core tensors of Eq. (8) therefore carry all sample- and time-specific information, and become the only objects on which the diffusion model operates. We concatenate them along the channel mode,

$$\mathbf{x}_{\text{joint}}^{(i)} = \text{Concat}_{R_1}\big[\mathcal{W}_i^{\text{phy}}, \mathcal{W}_i^{\text{sen}}\big] \in \mathbb{R}^{T_{\text{seg}} \times R_{\text{joint}} \times R_2 \times \cdots \times R_{D+1}}, \quad R_{\text{joint}} = R_1^{\text{phy}} + R_1^{\text{sen}}, \tag{9}$$

so that fields and sensing become a single sequence-valued latent on which one denoiser can attend to either modality at every layer.

3.3 Joint Latent Diffusion: Training and Sampling

We use a conditional denoiser $D_\theta(\,\cdot\,; \sigma, \mathbf{h}_{\text{hist}}^{(i)})$ that takes a σ-noised joint latent and the encoded history $\mathbf{h}_{\text{hist}}^{(i)}$ as input, and predicts the clean joint latent $\mathbf{x}_{\text{joint}}^{(i)}$. During training, D_θ is trained on joint trajectories constructed by a costly offline sequence-level optimizer. At test time, D_θ generates future fields and sensor trajectories conditioned only on historical observations. Details of training trajectory construction and network architecture are provided in Appendices C.2 and E.

From frame-independent to sequence-aware noise To produce temporally coherent trajectories, we follow the functional diffusion of sequence data Chen et al. [5], Biloš et al. [1], Kerrigan et al. [18]: instead of the i.i.d. Gaussian noise of standard EDM (Eq. (2)), we corrupt the latent with a single Gaussian-Process(GP) sample $\boldsymbol{\eta}^{(i)}$ correlated along the time mode of $\mathbf{x}_{\text{joint}}^{(i)}$. This re-frames the problem as functional temporal diffusion over sequence-valued latents rather than pointwise diffusion over per-frame noise. The motivation is direct: under i.i.d. noise, each frame in $\mathbf{x}_{\text{joint}}^{(i)}$ is corrupted independently and the denoiser recovers each frame in isolation, leaving no handle on trajectory smoothness at sampling time. A GP-correlated draw instead couples adjacent frames in the denoising task, making the resulting denoiser sequence-aware — without adding any regularizer to the loss.

Matérn-3/2 correlated perturbations We instantiate the correlated draw with a Matérn-$3/2$ kernel. Let $\mathbf{K}_{\text{time}} \in \mathbb{R}^{T_{\text{seg}} \times T_{\text{seg}}}$ denote its Gram matrix on the segment with Cholesky factor $\mathbf{L}\mathbf{L}^\top = \mathbf{K}_{\text{time}}$, and let $\boldsymbol{\xi}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be a standard Gaussian tensor of the same shape as $\mathbf{x}_{\text{joint}}^{(i)}$. We then define

$$\boldsymbol{\eta}^{(i)} = \big(\mathbf{L} \otimes \mathbf{I}_{R_{\text{joint}} R_2 \cdots R_{D+1}}\big)\ \text{vec}(\boldsymbol{\xi}^{(i)}), \tag{10}$$

reshaped back to the latent shape. By construction $\boldsymbol{\eta}^{(i)}$ is correlated only along the time mode and remains independent on all other modes; the kernel itself corresponds to a once-differentiable Gauss–Markov prior, and we discuss the rationale for this choice over RBF and white-noise endpoints

in Appendix C.3. By forcing D_θ to recover $\mathbf{x}^{(i)}_{\text{joint}}$ from a temporally smooth perturbation, we bias its outputs toward temporally smooth trajectories — both for fields and for sensor positions — which is the regularity we want for mobile sensing. We train with the EDM loss but with $\boldsymbol{\eta}^{(i)}$ in place of $\mathbf{n}$:

$$\mathcal{L}_{\text{joint}} = \mathbb{E}_{i,\sigma,\boldsymbol{\xi}}\left[\lambda(\sigma)\left\|D_\theta(\mathbf{x}^{(i)}_{\text{joint}} + \sigma\boldsymbol{\eta}^{(i)}; \sigma, \mathbf{h}^{(i)}_{\text{hist}}) - \mathbf{x}^{(i)}_{\text{joint}}\right\|_2^2\right]. \tag{11}$$

Conditional posterior sampling At inference, we observe the first K_{obs} frames of a segment and must generate the remainder. We split $\mathbf{x}^{(i)} = (\mathbf{x}^{(i)}_{1:K_{\text{obs}}}, \mathbf{x}^{(i)}_{K_{\text{obs}}+1:T_{\text{seg}}})$ and softly anchor the observed prefix during reverse denoising:

$$\mathbf{x}^{(i)}_{r,\,1:K_{\text{obs}}} \leftarrow (1-\alpha_r)\,\mathbf{x}^{(i)}_{r,\,1:K_{\text{obs}}} + \alpha_r\,\widetilde{\mathbf{x}}^{(i)}_{\text{hist},r}, \tag{12}$$

where $\widetilde{\mathbf{x}}^{(i)}_{\text{hist},r}$ is the historical latent perturbed to the current noise level and α_r anneals from 1 to 0 across reverse steps (schedule in Appendix C.4). The suffix follows the standard reverse dynamics. After sampling, we decode both modalities by inverse Functional Tucker and recover sensor coordinates from the SPF channel via the top-k $\arg\max$ readout of §3.2

3.4 Theoretical Properties

The analysis below provides two population-level consistency checks for the design of §3.3. First, under an idealized linear-Gaussian posterior model, a denoising/reconstruction risk is aligned with a sequence-aggregated A-optimality criterion, motivating the reconstruction-based trajectory supervision used by the offline optimizer. Second, the GP kernel used for temporal perturbations induces an algebraic bound on the discrete variation of trajectories favored by the corresponding negative log-prior. The first result is derived for the standard EDM loss with i.i.d. noise; the correlated perturbation in Eq. (10) is an additional temporal-coherence design and is not used in the closed-form Bayes-risk derivation.

Sensing consistency. Active sensing is fundamentally about uncertainty reduction: an informative trajectory should shrink the posterior covariance of the field. The next proposition shows that, under standard linear-Gaussian assumptions, an idealized EDM denoising risk is equivalent up to constants to a sequence-aggregated posterior-variance criterion.

Proposition 1 (EDM risk and sequence A-optimality). *Under Assumptions A.1–A.3 of Appendix A, there exist constants $C_{\text{low}}, C_{\text{high}} > 0$ such that*

$$C_{\text{low}} \sum_{t=1}^{T} \operatorname{tr}(\boldsymbol{\Sigma}_{\mathbf{s}_t}) \;\le\; \mathcal{L}^{\text{seq}}_{\text{EDM},*} \;\le\; C_{\text{high}} \sum_{t=1}^{T} \operatorname{tr}(\boldsymbol{\Sigma}_{\mathbf{s}_t}). \tag{13}$$

This result analyzes an idealized denoising posterior conditioned on sparse observations, and therefore supports the reconstruction-based trajectory-supervision objective rather than proving that the finite-sample joint diffusion loss directly optimizes A-optimality.

Kinematic feasibility. The Matérn-$3/2$ correlated noise of Eq. (10) is the only sequence-aware ingredient in DyPSI's diffusion loss; we do *not* add an explicit GP regularizer during diffusion training. Nevertheless, the same kernel defines a GP negative log-prior $\mathcal{L}_{\text{GP}}(\mathbf{s}) = \frac{1}{2}\mathbf{s}^\top\mathbf{K}^{-1}\mathbf{s}$ on trajectories. The next proposition characterizes the kinematic content of this kernel-level prior.

Proposition 2 (GP regularization controls trajectory variation). *Let $\mathbf{D}$ be the first-order time-difference matrix and $c_{\mathbf{K}} := \lambda_{\max}(\mathbf{D}\mathbf{K}\mathbf{D}^\top)$ for a positive-definite GP kernel matrix $\mathbf{K}$. For a zero-mean or time-constant-reference GP prior,*

$$\sum_{t=1}^{T-1} \left\|\mathbf{s}_{t+1} - \mathbf{s}_t\right\|_2^2 \;\le\; 2\,c_{\mathbf{K}}\,\mathcal{L}_{\text{GP}}(\mathbf{s}). \tag{14}$$

For a time-varying reference mean, an additional reference-variation term appears; see Appendix A. The result is a kernel-level algebraic characterization, not a control-theoretic guarantee.

Together, the two propositions motivate the two design axes of §3.3: reconstruction-based trajectory supervision aligns with posterior-variance reduction in an idealized setting, while the temporal GP kernel biases generated trajectories toward smooth motion. Full assumptions and proofs are in Appendix A.

4 Related Work

Sparse field reconstruction. Reconstructing dense spatiotemporal fields from sparse observations is a classical ill-posed inverse problem [16]. Deterministic deep models such as VoronoiCNN [10] and Senseiver [36] recover dense fields from scattered measurements through spatial interpolation structures or attention-based latent representations. Generative approaches including DiffusionPDE [14], S^3GM [24], and diffusion posterior sampling [6] instead model conditional field distributions under partial observation, and SDIFT [5] generates full-field evolution from irregular sparse observations in Functional Tucker space. In all of these methods the sensing locations are fixed, random, externally given, or optimized outside the generative model, whereas DyPSI generates future fields and future sensing trajectories jointly. A detailed itemized delineation between DyPSI and SDIFT is given in Appendix B.

Sensor placement and experimental design. Sensor placement is rooted in optimal experimental design [31, 33], where A-, D-, and E-optimality criteria quantify the information content of a sensing configuration. Classical data-driven methods exploit low-dimensional structure through POD and QR based sensor selection [28], compressed sensing [7], and Gaussian process placement [20]. PhySense [27] couples learning-based reconstruction with placement optimization and is closest to our work, but its layouts remain static or frame-wise independent, while DyPSI treats field evolution and sensor motion as one sequence-level generative problem.

Dynamic and active sensing. Informative path planning, model predictive control, and reinforcement learning address adaptive sensing by optimizing action sequences against explicit reward or information objectives under a known or learned dynamics model [20, 38, 2, 29]. These methods require a specified reward and per-step decision procedure and typically plan over actions alone, with reconstruction handled by a separate estimator. DyPSI addresses the complementary formulation of jointly generating future fields and sensing trajectories conditioned on history, amortizing sequence-level planning into a single conditional sampling procedure. The analyses in §3.4 serve as design-time consistency checks rather than control-theoretic guarantees.

5 Experiments

We evaluate DyPSI on three physics sensing tasks: two-dimensional turbulent channel flow, global sea surface temperature recovery with geographic masking, and surface pressure reconstruction over an irregular three-dimensional car mesh. These benchmarks cover regular Cartesian grids, constrained geospatial domains, and unstructured surface geometries in 2D and 3D settings.

Benchmarks: The benchmarks include: (1) Turbulent flow: two-dimensional channel-flow turbulence on a 128×48 regular grid [36]; (2) Sea temperature: daily global ocean surface temperature at 1° resolution (360×180) from the GLORYS12 reanalysis dataset spanning 27 years [21]; (3) Car aerodynamics: surface pressure fields simulated via OpenFOAM [15] for a passenger-car mesh with 95,428 vertices from ShapeNet [4].

Baselines and Experimental Settings: We compare against SSPOR [28], VoronoiCNN [10], Senseiver [36], S^3GM [24], DiffusionPDE [14], and PhySense [27] under both random layouts and static optimized placement. Identical training budgets are enforced across all methods, and reconstruction quality is measured by the relative L_2 metric. For turbulent flow, during training, sensor counts from 25 to 300 are randomly instantiated with coordinates sampled uniformly across the spatial extent, and the reconstructor is assessed under varying sensor layouts. Because DiffusionPDE and S^3GM incur prohibitive inference costs, their assessment is restricted to 500 test cases. For sea temperature, fields are down-sampled to $1^\circ \times 1^\circ$ resolution. During training, 10 to 100 sensors are randomly selected from valid ocean pixels, with continental land masses masked out. The training corpus comprises 9,843 daily snapshots from 1993 to 2019, and the test period covers 2020 to 2021. A binary land mask is supplied as auxiliary input to all methods. For car aerodynamics, aerodynamic pressure distributions are obtained via OpenFOAM simulations across inlet velocities of 20 to 40 m/s and yaw angles of -10° to 10°. Sensors are restricted to the vehicle exterior. Unless otherwise stated, DyPSI uses PhySense as the default teacher reconstructor for offline trajectory-supervision construction; all DyPSI test errors are computed from fields generated by the trained DyPSI model, not from test-time oracle optimization. All experiments are conducted with five independent random

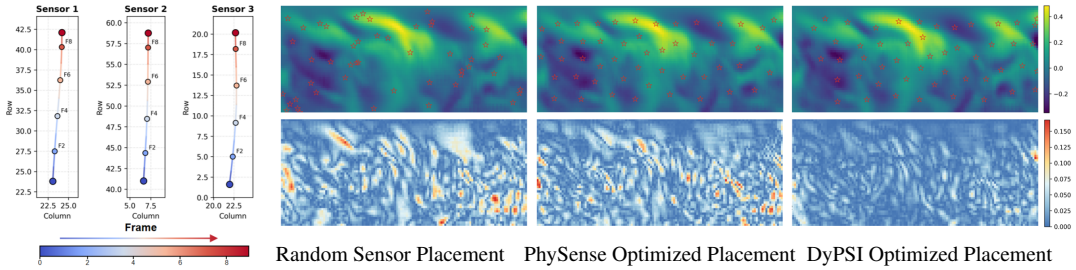

Figure 3: Qualitative and kinematic analysis on the turbulent flow benchmark. (a) Temporally coherent trajectories of 3 mobile sensors generated by DyPSI across frames 0-9. Color encodes temporal progression from early (blue) to late (red). (b) Reconstructed fields (top row) and point-wise error maps (bottom row) under random, static, and dynamic 50-sensor placements (denoted by ☆).

Table 1: Performance comparison on the turbulent flow and sea surface temperature benchmarks. Relative L_2 error is reported. DyPSI uses dynamically generated sensor trajectories; unless otherwise specified, its offline trajectory supervision is constructed using PhySense as the default teacher reconstructor. DyPSI errors are computed from fields generated by the trained joint diffusion model, without test-time oracle placement. # denotes the sensor count. Improvement quantifies the relative error reduction with respect to the strongest static baseline.

		Turbulent Flow				Sea Temperature			
Models	Placement	#200	#100	#50	#30	#100	#50	#25	#15
SSPOR	SSPOR	0.3018	0.4429	0.5921	0.6637	0.0756	0.0752	0.0789	0.0719
VoronoiCNN	Random	0.3588	0.4870	0.6919	0.7992	0.1496	0.1537	0.2433	0.2817
Senseiver	Random	0.1842	0.2316	0.3740	0.5746	0.0715	0.0732	0.0769	0.0784
S^3GM	Random	0.1856	0.2016	0.2681	0.6421	0.1115	0.1137	0.1581	0.1795
DiffusionPDE	Random	0.1993	0.3312	0.6604	0.9163	0.0664	0.0775	0.0859	0.1139
PhySense	Random	0.1233	0.1527	0.2586	0.5176	0.0439	0.0452	0.0477	0.0520
PhySense-opt	Static Opt.	0.1106	0.1257	0.1558	0.2157	0.0426	0.0430	0.0437	0.0439
DyPSI	Dynamic Gen.	**0.0986**	**0.1101**	**0.1352**	**0.1826**	**0.0387**	**0.0390**	**0.0394**	**0.0402**
Improvement		10.87%	12.41%	13.19%	15.35%	9.15%	9.30%	9.84%	8.43%

seeds. The main comparison tables report mean values for compactness, while Appendix D provides the complete results with standard deviations and paired t-test p-values against the strongest baseline in each setting. Further implementation details are relegated to Appendix C.

Tables 1 and 2 show that DyPSI achieves the lowest reconstruction error across all benchmarks and sensor budgets. Relative to PhySense-opt, it reduces error by 10.87% to 15.35% on turbulent flow, 8.43% to 9.84% on sea surface temperature, and 17.47% to 19.43% on car aerodynamics. The gains generally become more pronounced as sensor availability decreases, indicating that temporal adaptation is especially valuable when observations are sparse. On the car mesh, 15 dynamically coordinated sensors attain an error of 0.0311, outperforming PhySense-opt with 30 sensors and PhySense with 100 randomly placed sensors. Appendix D further shows that all turbulent flow and car aerodynamics improvements are significant at $p < 0.001$, while the sea surface temperature improvements are significant at $p \leq 0.01$. The consistency across regular grids, masked geospatial domains, and unstructured surface meshes indicates that the benefit is not restricted to a particular discretization. All DyPSI errors are produced by the learned generator without test-time oracle placement.

Table 2: Results on the car aerodynamics benchmark.

Models	#200	#100	#50	#30	#15
Senseiver	0.1009	0.1012	0.1018	0.1022	0.1053
DiffusionPDE	0.0967	0.0966	0.0987	0.1055	0.2095
PhySense	0.0375	0.0382	0.0395	0.0416	0.0465
PhySense-opt	0.0369	0.0370	0.0370	0.0372	0.0386
DyPSI	**0.0301**	**0.0301**	**0.0302**	**0.0307**	**0.0311**
Improvement	18.43%	18.65%	18.39%	17.47%	19.43%

Figure 3 visualizes three smooth trajectories that follow coherent flow structures over frames 0 to 9 without abrupt positional jumps. Under the 50-sensor setting, random placement leaves distributed

Table 3: Accuracy and kinematic feasibility under representative sparse sensor budgets. Mean values over five independent runs are reported. Viol. denotes the velocity violation rate, Smooth. denotes $\sum_t \|\mathbf{s}_{t+1} - 2\mathbf{s}_t + \mathbf{s}_{t-1}\|^2$, and runtime is measured in seconds per 64-frame sequence. The Per-frame Oracle and Sequence Teacher access future ground truth and perform test-time optimization, whereas DyPSI uses neither.

			Turbulent Flow, 50 sensors				**Sea Surface Temperature, 25 sensors**			
Strategy	Future GT	Test-time Opt.	Rel. $L_2 \downarrow$	Viol. $\downarrow$	Smooth. $\downarrow$	Runtime $\downarrow$	Rel. $L_2 \downarrow$	Viol. $\downarrow$	Smooth. $\downarrow$	Runtime $\downarrow$
Static PhySense-opt	×	×	0.1558	0	0	–	0.0437	0	0	–
Per-frame Oracle	✓	✓	0.1007	89%	142.5	887	0.0259	75%	165.3	628
Sequence Teacher	✓	✓	0.1206	5%	3.5	1683	0.0355	2%	4.5	1429
DyPSI	×	×	0.1352	8%	4.2	**1.9**	0.0394	4%	9.2	**2.1**

Table 4: Constructor/backbone generalization on the turbulent flow benchmark. Relative L_2 error is reported. For non-DyPSI rows, column groups denote the reconstruction backbone used for evaluation. For the DyPSI-opt row, column groups denote the teacher reconstructor used by the offline trajectory-supervision constructor to train DyPSI; the reported error is computed from fields generated by the trained DyPSI model.

	Senseiver				PhySense				DiffusionPDE			
Placement Strategy	#200	#100	#50	#30	#200	#100	#50	#30	#200	#100	#50	#30
Random Sample	0.1842	0.2316	0.3740	0.5746	0.1233	0.1527	0.2586	0.5176	0.1993	0.3312	0.6604	0.9163
Grid Sample	0.1615	0.2273	0.3845	0.5869	0.1235	0.1521	0.2436	0.4930	0.1932	0.3319	0.6703	0.9079
Min-Max Sample	0.1549	0.2091	0.3795	0.5747	0.1196	0.1533	0.3157	0.5022	0.1889	0.3307	0.6574	0.9010
PhySense-opt	0.1388	0.1750	0.3025	0.4570	0.1106	0.1257	0.1558	0.2157	0.1672	0.2841	0.5319	0.7904
DyPSI-opt	**0.1259**	**0.1670**	**0.2863**	**0.4527**	**0.0986**	**0.1101**	**0.1352**	**0.1826**	**0.1569**	**0.2640**	**0.5028**	**0.7116**

residuals, while static optimization retains errors near structures that have advected away from the fixed sensors. DyPSI reduces these localized errors by repositioning sensors around informative, high-gradient filaments. The sea temperature trajectories similarly respect the land mask and concentrate sensing within thermally informative ocean regions.

Accuracy and Kinematic Feasibility: Table 3 compares four sensing strategies on identical test sequences with a frozen reconstructor. The Per-frame Oracle independently applies PGD at each future frame without temporal or kinematic constraints, whereas the Sequence Teacher optimizes complete trajectories using future ground truth. DyPSI instead generates fields and trajectories from historical observations without future ground truth or test-time optimization.

The Per-frame Oracle obtains the lowest errors of 0.1007 on turbulent flow and 0.0259 on sea surface temperature, but violates the velocity limit in 89% and 75% of transitions. Its smoothness costs of 142.5 and 165.3 also reveal severe inter-frame discontinuities. DyPSI reduces the violation rates to 8% and 4% and the smoothness costs to 4.2 and 9.2, while remaining within 0.0146 and 0.0039 of the Sequence Teacher in reconstruction error. It requires only 1.9 and 2.1 seconds per sequence, compared with 628 to 1683 seconds for the optimization-based alternatives. These results quantify the accuracy and computational trade-offs of adaptive sensing under kinematic constraints. DyPSI thus provides a practical intermediate between non-adaptive static placement and costly ground-truth-informed trajectory optimization.

Constructor and Backbone Generalization: Table 4 evaluates conventional placements with three reconstruction backbones and uses each backbone independently as the teacher for a separately trained DyPSI model. No multi-teacher training or test-time ground-truth optimization is used, and the PhySense teacher setting matches the default configuration in the main comparison. Grid and Min-Max sampling do not consistently improve over random placement, indicating that spatial regularity alone is insufficient for tracking informative structures. DyPSI-opt achieves the lowest error for every teacher and sensor budget. With 30 sensors, it reduces the error from 0.4570 to 0.4527 for Senseiver, from 0.2157 to 0.1826 for PhySense, and from 0.7904 to 0.7116 for DiffusionPDE. Although the margin varies with backbone

Table 5: GP-kernel ablation on turbulent flow with 50 sensors. Mean $\pm$ standard deviation over five runs is reported.

Noise type	Rel. $L_2 \downarrow$	Smoothness $\downarrow$
i.i.d.	0.1515 ± 0.0039	18.6 ± 1.9
RBF	0.1489 ± 0.0025	6.2 ± 1.3
Matérn-3/2	$\mathbf{0.1352 \pm 0.0024}$	$\mathbf{4.2 \pm 1.0}$

capacity, the consistent gains across implicit, flow-based, and diffusion-based teachers show that the learned dynamic sensing strategy is not tied to a particular reconstruction architecture.

Sensitivity to the GP Kernel: We replace the Matérn-3/2 temporal perturbation with i.i.d. EDM noise or an RBF kernel on turbulent flow with 50 sensors. As shown in Table 5, i.i.d. noise increases both reconstruction error and trajectory irregularity. RBF improves smoothness relative to i.i.d. noise but retains a higher reconstruction error, suggesting that its smooth prior restricts necessary temporal variation. Matérn-3/2 achieves the best accuracy and smoothness, supporting its balance between temporal coherence and trajectory flexibility.

Table 6: Joint and separate generation on turbulent flow with 50 sensors. Mean $\pm$ standard deviation over five runs is reported.

Model	Rel. $L_2 \downarrow$	Smoothness $\downarrow$
Separate models	0.1421 ± 0.0045	9.1 ± 1.1
Joint model	$\mathbf{0.1352 \pm 0.0024}$	$\mathbf{4.2 \pm 1.0}$

We further compare joint generation with separate diffusion models that share only the history encoder. As shown in Table 6, joint generation reduces the relative L_2 error from 0.1421 to 0.1352 and the smoothness cost from 9.1 to 4.2. The simultaneous gains indicate that modeling both modalities within a shared denoiser better captures the dependence between field evolution and sensor motion.

6 Limitations

Several scope boundaries qualify our results. First, training trajectories are constructed by an offline optimizer with access to ground-truth fields, so DyPSI inherits a teacher dependence; the gain-retention analysis in §5 quantifies this amortization gap, and DyPSI does not fully match its oracle teacher. Second, like all diffusion-based reconstructors, DyPSI assumes deployment dynamics remain within the support of the training distribution; under abrupt regime shifts or strongly non-Gaussian sensing noise the generated fields and trajectories may be unreliable, and we make no out-of-distribution claims. Third, our evaluation is a history-conditioned open-loop rollout rather than fully closed-loop sensing, although the low inference cost permits receding-horizon re-planning. Fourth, our feasibility guarantees cover kinematic and domain constraints only, namely velocity and acceleration limits, land masking, and surface projection, and do not model full platform dynamics such as underactuation or ocean currents. Fifth, at sensor densities far above those evaluated here, SPF peaks may merge and the top-k readout requires the non-maximum-suppression fallback described in Appendix C. Finally, we assume point-wise observations, whereas real sensors such as radar often integrate over small spatial neighborhoods; incorporating regional observation models into both reconstruction and trajectory generation is a natural extension.

7 Conclusion

We present DyPSI, a dynamic physics sensing framework that jointly generates future physical fields and adaptive sensor trajectories by encoding both modalities into a shared Functional Tucker latent space via continuous Sensor Position Fields, where a Gaussian-process temporal kernel encourages temporal coherence during diffusion training. Complementary theoretical analyses provide design-time consistency checks, linking an idealized denoising risk to sequence-aggregated posterior variance and characterizing the trajectory-variation bias induced by the GP kernel. Extensive experiments on turbulent flow, sea-surface temperature, and car aerodynamics demonstrate that DyPSI consistently outperforms static and per-frame placement strategies, achieving particularly pronounced gains under sparse sensor budgets.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 6260070841 and in part by Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

[1] Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pages 2452–2470, 2023.

[2] Jonathan Binney and Gaurav S. Sukhatme. Branch and bound for informative path planning. In *IEEE International Conference on Robotics and Automation*, pages 2147–2154, 2012.

[3] Javier Burgués and Santiago Marco. Environmental chemical sensing using small drones: a review. *Science of the Total Environment*, 748:141172, 2020.

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[5] Panqi Chen, Yifan Sun, Lei Cheng, Yang Yang, Weichang Li, Yang Liu, Weiqing Liu, Jiang Bian, and Shikai Fang. Generating full-field evolution of physical dynamics from irregular sparse observations. *arXiv preprint arXiv:2505.09284*, 2025.

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023.

[7] David L Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4): 1289–1306, 2006.

[8] Shengren Fan, Biao Zhang, and Vladimir Kudryavtsev. Comprehensive assessment of ocean surface current retrievals using SAR Doppler shift and drifting buoy observations. *Remote Sensing*, 17(12):2007, 2025.

[9] Shikai Fang, Xin Yu, Zheng Wang, Shibo Li, Mike Kirby, and Shandian Zhe. Functional Bayesian Tucker decomposition for continuous-indexed tensor data. In *The Twelfth International Conference on Learning Representations*, 2024.

[10] Kai Fukami, Romit Maulik, Nesar Ramachandra, Koji Fukagata, and Kunihiko Taira. Global field reconstruction from sparse sensors with Voronoi tessellation-assisted deep learning. *Nature Machine Intelligence*, 3(11):945–954, 2021.

[11] Yaqing Gu and Dean S. Oliver. An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. *SPE Journal*, 12(04):438–446, 2007.

[12] Mohammad Yasin Hosseini and Yousef Shiri. Flow field reconstruction from sparse sensor measurements with physics-informed neural networks. *Physics of Fluids*, 36(7):073606, 2024.

[13] Haojie Hu, Senda Chen, Ying Shen, and Lin Zhang. To think or not to think: Pre-decisional reasoning budgets for referring audio-visual segmentation. *arXiv preprint arXiv:2610.21613*, 2026.

[14] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *NeurIPS*, 2024.

[15] Hrvoje Jasak. OpenFOAM: Open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1(2):89–94, 2009.

[16] Sergey I Kabanikhin. Inverse and ill-posed problems: theory and applications. *Inverse and Ill-posed Problems*, 2011.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[18] Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion generative models in infinite dimensions. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *PMLR*, pages 9538–9563, 2023.

[19] Christoph Kralovec and Martin Schagerl. Review of structural health monitoring methods regarding a multi-sensor approach for damage assessment of metal and composite structures. *Sensors*, 20(3):826, 2020.

[20] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[21] Jean-Michel Lellouche, Eric Greiner, Romain Bourdallé-Badie, Gilles Garric, Angélique Melet, et al. The Copernicus global 1/12 oceanic and sea ice GLORYS12 reanalysis. *Frontiers in Earth Science*, 9:698876, 2021.

[22] Naomi Ehrich Leonard, Derek A Paley, François Lekien, Rodolphe Sepulchre, David M Fratantoni, and Russ E Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[23] Yanyi Li, Xi Zhang, Yisi Luo, and Deyu Meng. Deep rank-one tensor functional factorization for multi-dimensional data recovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18539–18547, 2025.

[24] Zeyu Li, Wang Han, Yue Zhang, Qingfei Fu, Jingxuan Li, Lizi Qin, Ruoyu Dong, Hao Sun, Yue Deng, and Lijun Yang. Learning spatiotemporal dynamics with a pretrained generative model. *Nature Machine Intelligence*, 2024.

[25] Yisi Luo, Xile Zhao, Zhemin Li, Michael K. Ng, and Deyu Meng. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3351–3369, 2024.

[26] Xin Lyu, Xiaobing Li, Dongliang Dang, Huashun Dou, Kai Wang, and Anru Lou. Unmanned aerial vehicle (UAV) remote sensing in grassland ecosystem monitoring: A systematic review. *Remote Sensing*, 14(5):1096, 2022.

[27] Yuezhou Ma, Haixu Wu, Hang Zhou, Huikun Weng, Jianmin Wang, and Mingsheng Long. PhySense: Sensor placement optimization for accurate physics sensing. *arXiv preprint arXiv:2505.18190*, 2025.

[28] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[29] Philippe Morere, Roman Marchant, and Fabio Ramos. Sequential bayesian optimization as a pomdp for environment monitoring with uavs. In *IEEE International Conference on Robotics and Automation*, pages 6381–6388, 2017.

[30] Andrea Munafò, Enrico Simetti, Alessandro Turetta, and Andrea Caiti. Autonomous underwater vehicle teams for adaptive ocean sampling: a data-driven approach. *Ocean Dynamics*, 61(11): 1981–1994, 2011.

[31] Panos Y Papalambros and Douglass J Wilde. *Principles of optimal design: modeling and computation*. Cambridge University Press, 2000.

[32] Ananda Pascual, Simón Ruiz, Antonio Olita, Charles Troupin, María Claret, Benjamín Casas, Baptiste Mourre, Pierre-Marie Poulain, Antonio Tovar-Sánchez, Arthur Capet, Evan Mason, John T Allen, Amala Mahadevan, and Joaquín Tintoré. A multiplatform experiment to unravel meso- and submesoscale processes in an intense front (AlborEX). *Frontiers in Marine Science*, 4:39, 2017.

[33] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[34] Richard W. Reynolds, Nick A. Rayner, Thomas M. Smith, David C. Stokes, and Wanqiu Wang. An improved in situ and satellite SST analysis for climate. *Journal of Climate*, 15(13): 1609–1625, 2002.

[35] Richard W Reynolds, Thomas M Smith, Chunying Liu, Dudley B Chelton, Kenneth S Casey, and Michael G Schlax. Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20(22):5473–5496, 2007.

[36] Javier E Santos, Zachary R Fox, Arvind Mohan, Daniel O'Malley, Hari Viswanathan, and Nicholas Lubbers. Development of the Senseiver for efficient field reconstruction from sparse observations. *Nature Machine Intelligence*, 2023.

[37] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on signal processing*, 65(13):3551–3582, 2017.

[38] Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J. Kaiser. Efficient informative sensing using multiple robots. In *Journal of Artificial Intelligence Research*, volume 34, pages 707–755, 2009.

[39] Ryan N Smith, Yi Chao, Peggy P Li, David A Caron, Burton H Jones, and Gaurav S Sukhatme. Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. *International Journal of Robotics Research*, 29(10):1475–1497, 2010.

[40] Kunihiko Taira, Steven L Brunton, Scott TM Dawson, Clarence W Rowley, Tim Colonius, Beverley J McKeon, Oliver T Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55(12):4013–4041, 2017.

[41] Y. H. Tao, A. J. Fitzgerald, and V. P. Wallace. Non-contact, non-destructive testing in various industrial sectors with terahertz technology. *Sensors*, 20(3):712, 2020.

[42] Yanwu Zhang, James G. Bellingham, John P. Ryan, Brian Kieft, and Michael J. Stanway. Autonomous 4-D mapping and tracking of a coastal upwelling front by an autonomous underwater vehicle. *Journal of Field Robotics*, 33(1):67–81, 2016.

[43] Xiang Zhao, Zichen Deng, and Weiwei Zhang. Sparse reconstruction of surface pressure coefficient based on compressed sensing. *Experiments in Fluids*, 63(10):156, 2022.

[44] Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-conditional transformers towards universal neural PDE solvers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 79061–79088, 2025.

A Theoretical Details

This appendix provides the detailed assumptions and proof sketches for the theoretical motivation in the main paper. The results are established at the population level. They characterize idealized risks induced by simplified probabilistic models, rather than the approximation or optimization error of a finite-capacity neural network trained with finite data. Therefore, the theory should be interpreted as design-time consistency checks for the proposed objectives and kernels, rather than as a proof of global optimality for the practical DyPSI training procedure. Proposition A.6 analyzes the standard Gaussian denoising form of the EDM objective; the temporally correlated noise used in the full method is an additional design for temporal coherence, rather than part of the closed-form Bayes-risk derivation below.

Notation. Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the latent representation of the target physical field at timestep t, and let $\mathbf{s}_t$ denote the corresponding sensor configuration. Given $\mathbf{s}_t$ and the associated noisy sparse measurements $\mathbf{y}_t$, the posterior uncertainty of $\mathbf{x}_t$ is described by a covariance matrix $\mathbf{\Sigma}_{\mathbf{s}_t} \in \mathbb{R}^{d\times d}$.

Assumption A.1 (Conditional Gaussian Posterior). *For each timestep t, conditioned on the sensor configuration $\mathbf{s}_t$ and the corresponding noisy sparse measurements $\mathbf{y}_t$, the latent field posterior is Gaussian:*

$$\mathbf{x}_t \mid (\mathbf{s}_t, \mathbf{y}_t) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{s}_t}, \mathbf{\Sigma}_{\mathbf{s}_t}), \tag{15}$$

where $\mathbf{\Sigma}_{\mathbf{s}_t}$ is positive semidefinite.

Assumption A.2 (Uniform Spectral Bound). *There exists a finite constant $\Lambda_{\max} > 0$ such that*

$$\lambda_{\max}(\mathbf{\Sigma}_{\mathbf{s}_t}) \le \Lambda_{\max}, \qquad \forall t \in \{1, \dots, T\},\ \forall \mathbf{s}_t \in \mathcal{D}, \tag{16}$$

where $\mathcal{D}$ denotes the feasible set of sensor placements.

Assumption A.3 (Admissible EDM Weighting). *The EDM noise distribution $p(\sigma)$ and weighting function $\lambda(\sigma)$ satisfy*

$$\mathbb{E}_{\sigma\sim p(\sigma)}[\lambda(\sigma)] < \infty, \qquad \mathbb{E}_{\sigma\sim p(\sigma)}\left[\lambda(\sigma)\frac{\sigma^2}{\Lambda_{\max}+\sigma^2}\right] > 0. \tag{17}$$

Definition A.4 (Population-Optimal EDM Risk). *For a fixed timestep t, define the population-optimal EDM objective as*

$$\mathcal{L}^t_{\mathrm{EDM},*}(\mathbf{s}_t) := \mathbb{E}_{\sigma\sim p(\sigma)}\left[\lambda(\sigma)\inf_D \mathbb{E}_{\mathbf{x}_t,\mathbf{n}}\left[\|D(\mathbf{x}_t+\sigma\mathbf{n};\sigma)-\mathbf{x}_t\|_2^2\right]\right], \tag{18}$$

where $\mathbf{n}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, and the infimum is taken over all measurable denoisers D. The sequence-level risk is

$$\mathcal{L}^{\mathrm{seq}}_{\mathrm{EDM},*} := \sum_{t=1}^{T}\mathcal{L}^t_{\mathrm{EDM},*}. \tag{19}$$

Definition A.5 (Sequence-Aggregated A-Optimality). *For each timestep t, define*

$$\mathcal{L}^t_A := \mathrm{tr}(\mathbf{\Sigma}_{\mathbf{s}_t}), \tag{20}$$

and the corresponding sequence-aggregated criterion

$$\mathcal{L}^{\mathrm{seq}}_A := \sum_{t=1}^{T}\mathcal{L}^t_A = \sum_{t=1}^{T}\mathrm{tr}(\mathbf{\Sigma}_{\mathbf{s}_t}). \tag{21}$$

This is a temporal aggregation of classical A-optimality, rather than a full joint spatiotemporal covariance criterion.

Proposition A.6 (EDM Objective Linearly Controls Sequence-Aggregated A-Optimality). *Under Assumptions A.1–A.3, there exist constants $C_{\mathrm{low}}, C_{\mathrm{high}} > 0$ such that*

$$C_{\mathrm{low}}\,\mathcal{L}^{\mathrm{seq}}_A \le \mathcal{L}^{\mathrm{seq}}_{\mathrm{EDM},*} \le C_{\mathrm{high}}\,\mathcal{L}^{\mathrm{seq}}_A, \tag{22}$$

where

$$C_{\mathrm{high}} = \mathbb{E}_{\sigma\sim p(\sigma)}[\lambda(\sigma)], \qquad C_{\mathrm{low}} = \mathbb{E}_{\sigma\sim p(\sigma)}\left[\lambda(\sigma)\frac{\sigma^2}{\Lambda_{\max}+\sigma^2}\right]. \tag{23}$$

Proof sketch. Fix a timestep t and a noise level σ. Under Assumption A.1, the denoising problem

$$\mathbf{y} = \mathbf{x}_t + \sigma\mathbf{n}, \qquad \mathbf{n}\sim\mathcal{N}(\mathbf{0},\mathbf{I}), \tag{24}$$

is a linear Gaussian estimation problem. Hence, the Bayes-optimal denoiser is the conditional mean

$$D^*(\mathbf{y};\sigma) = \mathbb{E}[\mathbf{x}_t \mid \mathbf{y}], \tag{25}$$

and its minimum mean-square error equals the trace of the posterior covariance. Let $\{\lambda_{i,t}\}_{i=1}^d$ denote the eigenvalues of $\mathbf{\Sigma}_{\mathbf{s}_t}$. By spectral decomposition, the optimal single-step EDM risk is

$$\mathcal{L}^t_{\mathrm{EDM},*} = \mathbb{E}_{\sigma\sim p(\sigma)}\left[\lambda(\sigma)\sum_{i=1}^{d}\frac{\lambda_{i,t}\sigma^2}{\lambda_{i,t}+\sigma^2}\right]. \tag{26}$$

For the upper bound, since $\sigma^2/(\lambda_{i,t}+\sigma^2) \le 1$, we obtain

$$\mathcal{L}^t_{\mathrm{EDM},*} \le \mathbb{E}_\sigma[\lambda(\sigma)] \sum_{i=1}^{d} \lambda_{i,t} = C_{\mathrm{high}}\, \mathcal{L}^t_A. \tag{27}$$

For the lower bound, Assumption A.2 implies $\lambda_{i,t} \le \Lambda_{\max}$, hence

$$\frac{\sigma^2}{\lambda_{i,t}+\sigma^2} \ge \frac{\sigma^2}{\Lambda_{\max}+\sigma^2}, \tag{28}$$

which yields

$$\mathcal{L}^t_{\mathrm{EDM},*} \ge \mathbb{E}_\sigma \left[\lambda(\sigma)\frac{\sigma^2}{\Lambda_{\max}+\sigma^2}\right] \sum_{i=1}^{d} \lambda_{i,t} = C_{\mathrm{low}}\, \mathcal{L}^t_A. \tag{29}$$

Summing over $t = 1, \dots, T$ proves the result. □

Interpretation. Proposition A.6 should be read as an idealized posterior-denoising result. Under a conditional Gaussian posterior model, the optimal denoising risk is equivalent up to constant factors to a sequence-aggregated posterior-variance criterion. This motivates the reconstruction-based trajectory-supervision objective used by the offline optimizer.

To characterize the temporal regularity induced by the GP kernel, we consider the corresponding Gaussian-process negative log-prior on sensor trajectories.

Definition A.7 (GP Temporal Regularizer). *Let $\mathbf{s}^{(j,d)} \in \mathbb{R}^T$ denote coordinate d of the trajectory of sensor j over the discrete time grid. For a positive definite temporal kernel matrix $\mathbf{K}$ and mean vector $\mathbf{m}^{(j,d)}$, define*

$$\mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}) := \frac{1}{2}(\mathbf{s}^{(j,d)} - \mathbf{m}^{(j,d)})^\top \mathbf{K}^{-1}(\mathbf{s}^{(j,d)} - \mathbf{m}^{(j,d)}). \tag{30}$$

For a D-dimensional trajectory with k sensors, the total GP regularizer is the coordinate-wise sum

$$\mathcal{L}_{\mathrm{GP}}(\mathbf{s}) := \sum_{j=1}^{k}\sum_{d=1}^{D} \mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}). \tag{31}$$

Proposition A.8 (GP Regularization Controls Discrete Trajectory Variation). *Let $\mathbf{D} \in \mathbb{R}^{(T-1)\times T}$ be the first-order difference matrix defined by*

$$(\mathbf{D}\mathbf{u})_t = \mathbf{u}_{t+1} - \mathbf{u}_t. \tag{32}$$

Define

$$c_{\mathbf{K}} := \lambda_{\max}(\mathbf{D}\mathbf{K}\mathbf{D}^\top). \tag{33}$$

Then, for any trajectory coordinate $\mathbf{s}^{(j,d)}$ and reference mean $\mathbf{m}^{(j,d)}$,

$$\sum_{t=1}^{T-1} \left[\left(s^{(j,d)}_{t+1} - m^{(j,d)}_{t+1}\right) - \left(s^{(j,d)}_{t} - m^{(j,d)}_{t}\right)\right]^2 \le 2c_{\mathbf{K}}\, \mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}). \tag{34}$$

Thus the GP penalty controls the discrete variation of the trajectory relative to its reference mean. For the absolute variation,

$$\sum_{t=1}^{T-1} \left(s^{(j,d)}_{t+1} - s^{(j,d)}_{t}\right)^2 \le 4c_{\mathbf{K}}\, \mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}) + 2\sum_{t=1}^{T-1} \left(m^{(j,d)}_{t+1} - m^{(j,d)}_{t}\right)^2. \tag{35}$$

If the reference mean is zero or constant in time, the sharper bound

$$\sum_{t=1}^{T-1} \left(s^{(j,d)}_{t+1} - s^{(j,d)}_{t}\right)^2 \le 2c_{\mathbf{K}}\, \mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}) \tag{36}$$

holds. Consequently, for k sensors in D spatial dimensions,

$$\sum_{t=1}^{T-1}\sum_{j=1}^{k}\left\|\mathbf{s}_{t+1}^{(j)}-\mathbf{s}_{t}^{(j)}\right\|_2^2 \le 4c_{\mathbf{K}}\,\mathcal{L}_{\mathrm{GP}}(\mathbf{s}) + 2\sum_{t=1}^{T-1}\sum_{j=1}^{k}\left\|\mathbf{m}_{t+1}^{(j)}-\mathbf{m}_{t}^{(j)}\right\|_2^2. \tag{37}$$

For zero-mean or time-constant-reference GP priors, this reduces to

$$\sum_{t=1}^{T-1}\sum_{j=1}^{k}\left\|\mathbf{s}_{t+1}^{(j)}-\mathbf{s}_{t}^{(j)}\right\|_2^2 \le 2c_{\mathbf{K}}\,\mathcal{L}_{\mathrm{GP}}(\mathbf{s}). \tag{38}$$

The corresponding discrete kinetic-energy form follows by dividing both sides by $(\Delta t)^2$.

Proof sketch. Let $\mathbf{u}^{(j,d)} := \mathbf{s}^{(j,d)} - \mathbf{m}^{(j,d)}$. By definition,

$$2\mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}) = (\mathbf{u}^{(j,d)})^\top \mathbf{K}^{-1}\mathbf{u}^{(j,d)}. \tag{39}$$

The discrete variation relative to the reference mean is

$$\sum_{t=1}^{T-1}\left(u_{t+1}^{(j,d)}-u_{t}^{(j,d)}\right)^2 = (\mathbf{u}^{(j,d)})^\top \mathbf{D}^\top\mathbf{D}\mathbf{u}^{(j,d)}. \tag{40}$$

By a generalized Rayleigh quotient argument,

$$(\mathbf{u}^{(j,d)})^\top \mathbf{D}^\top\mathbf{D}\mathbf{u}^{(j,d)} \le \lambda_{\max}\left(\mathbf{K}^{1/2}\mathbf{D}^\top\mathbf{D}\mathbf{K}^{1/2}\right)(\mathbf{u}^{(j,d)})^\top \mathbf{K}^{-1}\mathbf{u}^{(j,d)}. \tag{41}$$

Since

$$\lambda_{\max}\left(\mathbf{K}^{1/2}\mathbf{D}^\top\mathbf{D}\mathbf{K}^{1/2}\right) = \lambda_{\max}(\mathbf{D}\mathbf{K}\mathbf{D}^\top) = c_{\mathbf{K}}, \tag{42}$$

we obtain

$$\sum_{t=1}^{T-1}\left(u_{t+1}^{(j,d)}-u_{t}^{(j,d)}\right)^2 \le 2c_{\mathbf{K}}\mathcal{L}_{\mathrm{GP}}(\mathbf{s}^{(j,d)}). \tag{43}$$

This proves the relative-variation bound. For a general time-varying mean,

$$s_{t+1}^{(j,d)} - s_{t}^{(j,d)} = \left(u_{t+1}^{(j,d)}-u_{t}^{(j,d)}\right) + \left(m_{t+1}^{(j,d)}-m_{t}^{(j,d)}\right), \tag{44}$$

and applying $(a+b)^2 \le 2a^2 + 2b^2$ gives the stated general bound. If the reference mean is zero or constant in time, the second term vanishes and the sharper bound follows directly without this relaxation. Summing over coordinates and sensors proves the vector-valued statement. □

Interpretation. Proposition A.8 shows that the GP negative log-prior controls the discrete variation of a trajectory relative to its prescribed mean. When the GP mean is zero or constant in time, this directly yields a bound on absolute frame-to-frame sensor motion; for a time-varying mean, the absolute motion is additionally controlled by the variation of the reference mean. This is a kernel-level characterization of temporal smoothness, not a control-theoretic guarantee for the deployed generative sampler.

B Delineation from SDIFT

DyPSI builds on two ingredients introduced by SDIFT [5]: the Functional Tucker representation used to encode fields into a continuous-coordinate latent, and the use of Gaussian-process-correlated perturbations within an EDM diffusion objective. These components are inherited and are not claimed as new in isolation. The task formulation, the generated object, and all sensing-specific machinery of DyPSI are new. Table 7 summarizes the relationship item by item, and we expand on the key distinctions below.

SDIFT addresses reconstruction: given irregular sparse observations, it generates the evolution of the physical field, with sensor locations supplied externally and held fixed as conditioning. DyPSI addresses dynamic active sensing, in which sensor placement is itself an output of the model. The diffusion model in DyPSI generates future fields and future sensor trajectories jointly from the

Table 7: Itemized delineation between SDIFT and DyPSI. Inherited components are explicitly credited; the remaining rows are new in this work.

Aspect	SDIFT [5]	DyPSI (Ours)
Task	Full-field generation from irregular sparse observations	Dynamic active sensing: coupled field reconstruction and adaptive path planning
Generated object	Physical fields $\mathcal{P}_{t+1:T}$ only	Joint fields $\mathcal{P}_{t+1:T}$ and sensor trajectories $\mathbf{s}_{t+1:T}$
Sensor role	Fixed, exogenous conditioning variable	A jointly generated modality
Sensor representation	None	Sensor Position Field: fixed-kernel encoding of discrete coordinates into a continuous surrogate
Latent structure	Functional Tucker over fields	Joint core with shared spatial bases, modality-specific channel ranks, and a concatenated unified latent
Training supervision	Observed fields	Offline-to-online amortization of a GT-informed sequence-level optimizer
Temporal treatment	GP-correlated noise	GP-correlated noise plus a kinematic analysis linking the kernel to bounded trajectory variation
Theory	—	EDM risk linked to sequence-aggregated A-optimality (Proposition 1)
Evaluation	Field reconstruction benchmarks	Dynamic sensing with kinematic-feasibility metrics and oracle trade-off analysis

observed history, so the sensing configuration is lifted from a conditioning variable to a first-class generated modality. This changes both the modeling target and the evaluation: generated fields are scored against held-out ground truth, and generated trajectories are additionally scored by kinematic-feasibility metrics that have no counterpart in SDIFT.

Because sensor coordinates are discrete and low-dimensional while physical fields are continuous functions on the spatial domain, the two cannot be encoded by a shared representation directly. DyPSI introduces the Sensor Position Field, a deterministic fixed-kernel encoding that lifts each sensor configuration into a continuous surrogate on the same domain as the field. The encoding is closed form and fully differentiable in both directions, with coordinates recovered by a top-k argmax readout that is exact whenever the kernel bandwidth is below the typical inter-sensor spacing, a condition that holds at all sensor densities evaluated in this work. SDIFT contains no such mechanism, since it never needs to generate sensor positions.

DyPSI retains the Functional Tucker backbone but restructures the latent: spatial bases are shared across the physical-field and SPF branches, channel-mode ranks are modality-specific, and the two cores are concatenated into a single sequence-valued joint latent on which one denoiser attends to both modalities at every layer. The ablation in §5 shows that this joint construction outperforms two independent diffusion models coupled only through the history encoder on both reconstruction accuracy and trajectory smoothness. Training supervision also departs from SDIFT: rather than fitting observed fields, DyPSI amortizes a costly offline sequence-level optimizer that has access to ground-truth fields, distilling its dynamic sensing strategy into a history-conditioned generator that requires neither future ground truth nor test-time optimization. The gain-retention analysis in §5 quantifies how much of the teacher's advantage survives this amortization.

In short, SDIFT generates fields *given* sensors, while DyPSI generates sensors *together with* fields. The Sensor Position Field, the joint latent, the offline-to-online amortization of trajectory supervision, the kinematic analysis, and the dynamic active-sensing evaluation are new contributions of this work. Figure 2 reuses the encode–decode schematic of SDIFT only for the shared representation stage, as noted in its caption.

C Implementation Details

This appendix collects the metrics, architecture, training, sampling, and per-benchmark hyperparameter choices used in §5, together with detailed rationales that were referenced from but omitted in the main text. Processed data and the full repository will be released upon acceptance; an anonymized code repository is available at `https://github.com/mengqi132/DyPSI`.

C.1 Metrics

To evaluate the accuracy of the reconstructed physical field, we employ the widely used *relative L2* metric, defined as

$$\text{Relative } L_2 = \frac{\|\mathbf{u} - \hat{\mathbf{u}}\|_2}{\|\mathbf{u}\|_2}, \tag{45}$$

where $\mathbf{u}$ denotes the reference (ground-truth) field and $\hat{\mathbf{u}}$ represents the model reconstruction.

C.2 Architecture

Functional Tucker basis networks. Each per-mode basis $\phi_k : \Omega_k \to \mathbb{R}^{R_k}$ is parameterized by a 4-layer coordinate MLP of width 128 with sinusoidal positional encoding. For 2D benchmarks the spatial bases share weights across the two coordinate axes; for the 3D mesh benchmark each axis uses an independent MLP, and the surface-mesh embedding is precomputed via the signed-distance projection of the original ShapeNet vertices.

Joint denoiser. The denoiser D_θ is a U-Net operating on $\mathbf{x}^{(i)}_{\text{joint}} \in \mathbb{R}^{T_{\text{seg}} \times R_{\text{joint}} \times R_2 \times \cdots \times R_{D+1}}$. It uses standard EDM preconditioning [17] with skip- and residual-scalings $c_{\text{skip}}, c_{\text{out}}, c_{\text{in}}, c_{\text{noise}}$ as defined therein, treats the modality channel R_{joint} as feature dimension, and incorporates temporal self-attention along the T_{seg} axis at each resolution.

History encoder. The segment-level history encoding $\mathbf{h}^{(i)}_{\text{hist}}$ is produced by a 4-layer Transformer over the observed prefix $(\mathbf{s}_\tau, \mathbf{y}_\tau)_{\tau=1}^{K_{\text{obs}}}$ embedded by a shared MLP. The encoded vector is broadcast and concatenated to the denoiser feature map at every block.

SPF bandwidth selection (extended). Equation (7) requires choosing (σ_x, σ_y) (and σ_z for $D = 3$). We pick $\sigma_x = \sigma_y \approx 1.5\Delta$, where Δ is the local grid spacing, so that the 3σ support of each Gaussian blob covers roughly 9×9 grid points — locally supported but, at the typical sensor densities of our benchmarks, with negligible overlap. Empirically, varying σ within a factor of two has no noticeable effect on reconstruction error. For the surface-mesh case we use a geodesic-distance-based kernel with the same effective support measured in mesh edge lengths; per-benchmark values are listed in Appendix C.6.

C.3 Training

We train all benchmarks with Adam (lr $= 2\times10^{-4}$, β_1=0.9, β_2=0.999), batch size 16 segments, and 200k iterations. The EDM noise schedule follows Karras et al. [17]: $\ln\sigma \sim \mathcal{N}(P_{\text{mean}}{=}-1.2, P_{\text{std}}{=}1.2)$ with weighting $\lambda(\sigma) = (\sigma^2{+}\sigma^2_{\text{data}})/(\sigma \cdot \sigma_{\text{data}})^2$ and σ_{data}=0.5. Segment length is T_{seg}=16. Mixed-precision (bfloat16) is used throughout.

Matérn-3/2 kernel choice (extended). The temporal kernel $\mathbf{K}_{\text{time}}$ in Eq. (10) uses signal variance σ^2_{GP}=1 and length-scale ℓ=4 frames, so that the effective correlation horizon spans roughly a quarter of one training segment. We chose Matérn-$3/2$ over the two natural endpoints — the squared-exponential (RBF) kernel and white noise — for the following reason. Matérn-$3/2$ corresponds to a once-differentiable Gauss–Markov prior whose 1D spectral density decays as $O(\omega^{-4})$, matching the "locally smooth, long-range independent" regime of typical mobile-sensing trajectories. By contrast, the C^∞ RBF kernel imposes infinitely many derivatives of regularity and tends to over-smooth legitimate inter-frame attention shifts, while white noise reduces to the standard EDM perturbation and forfeits sequence awareness altogether. Matérn-$3/2$ sits between these extremes and was chosen accordingly.

Table 8: Benchmark-specific hyperparameters for the experiments of §5.

	Turbulent Flow	Sea Temperature	Car Aerodynamics
Spatial dimension D	2	2	3
Grid / mesh size	128×48	360×180	95,428 vertices
Channels C	2 (velocity)	1 (SST)	1 (pressure)
Modality channel ranks $(R_1^{\text{phy}}, R_1^{\text{sen}})$	$(16, 8)$	$(16, 8)$	$(24, 8)$
Spatial ranks $(R_2, \ldots, R_{D+1})$	$(32, 16)$	$(48, 32)$	$(24, 24, 16)$
SPF bandwidth (in grid units)	$(1.5, 1.5)$	$(1.5, 1.5)$	$(2.0, 2.0, 2.0)$
Segment length T_{seg}	16	16	16
Observed prefix K_{obs}	4	4	4
Sensor budget k (test-time)	$\{30, 50, 100, 200\}$	$\{15, 25, 50, 100\}$	$\{15, 30, 50, 100, 200\}$

C.4 Sampling

At inference we use the second-order Heun sampler from Karras et al. [17] with 50 reverse steps. The soft-guidance coefficient α_r in Eq. (12) is annealed linearly from $\alpha_0{=}1$ at the highest noise level to $\alpha_R{=}0$ at the cleanest noise level, ensuring the prefix is exactly anchored at the start of denoising and freely refined toward the end. Sensor coordinates are extracted from each generated SPF $\widehat{\mathcal{S}}_t$ by a top-k $\arg\max$ readout: we take the k grid points (or mesh vertices, in 3D) with the largest SPF response.

Why naive top-k $\arg\max$ suffices. Because the bandwidth in Eq. (7) is set narrower than the typical inter-sensor spacing (§C.2), each generated sensor contributes a well-separated peak, and the magnitude-based readout returns one location per sensor without any post-processing. We verified this empirically by inspecting the histogram of pairwise distances between the k extracted coordinates on every benchmark: in all three datasets and every sensor budget we evaluate, no two extracted coordinates fell within a single bandwidth window of each other, which confirms that the top-k values do correspond to k distinct peaks rather than a cluster around the strongest one.

Robust fallback for higher-density regimes. The argument above assumes that bandwidth σ_x stays comfortably below the inter-sensor spacing. In settings where sensor density is forced higher than what the chosen bandwidth supports — for example, packing more than ~ 300 sensors onto our 128×48 turbulence grid — two peaks could merge and naive top-k would return multiple points around the strongest, missing some of the others. The standard remedy is a non-maximum suppression (NMS) pass, e.g. a 3×3 local-maximum filter applied to $\widehat{\mathcal{S}}_t$ before ranking, which guarantees one extracted point per local maximum. We did not require this fallback in any of our reported benchmarks, but flag it as a drop-in replacement for the readout in regimes that violate the bandwidth-vs-density assumption.

C.5 End-to-End Training and Inference Algorithm

For completeness we summarize the full DyPSI pipeline as a four-phase procedure (Algorithm 1). Phase 1 (data preparation) generates paired field/sensor sequences by applying the model-agnostic trajectory optimizer described in §E and Algorithm 2 to any pre-trained reconstructor (e.g., PhySense, DiffusionPDE, or Senseiver), yielding temporally coherent and kinematically feasible sensor trajectories alongside the ground-truth physical fields. The remaining three phases constitute our core method: Functional Tucker pre-training (Phase 2), joint diffusion training (Phase 3), and conditional sampling at inference (Phase 4). Symbols not redefined here follow the main text.

C.6 Per-Benchmark Hyperparameters

Table 8 summarizes the benchmark-specific architectural and dataset hyperparameters for the three physics-sensing tasks evaluated in §5. The modality channel ranks $(R_1^{\mathrm{phy}}, R_1^{\mathrm{sen}})$ control the expressiveness of the physical field and sensor position field branches within the shared Functional Tucker latent space; we allocate a higher channel rank to the physical field ($R_1^{\mathrm{phy}} > R_1^{\mathrm{sen}}$) because the physical dynamics are typically more complex than the sensor placement patterns. Spatial ranks $(R_2, \ldots, R_{D+1})$ are chosen proportional to the spatial resolution of each benchmark. The SPF bandwidth is set to approximately $1.5\times$ the local grid spacing so that each Gaussian blob covers roughly a 9×9 neighborhood while keeping inter-sensor overlap negligible at the evaluated densities. At inference, test trajectories longer than the training segment length are generated autoregressively in sliding windows: the final K_{obs} frames of each generated segment serve as the observed historical prefix for the next segment, repeating until the full test sequence is covered.

C.7 Compute

All experiments were run on a single NVIDIA A100-80GB. Training takes approximately 14 hours for the turbulent-flow benchmark, 22 hours for the sea-surface temperature benchmark, and 28 hours for the car-aerodynamics benchmark. Inference (per test trajectory of length $T{=}64$, sensor count $k{=}100$) takes 1.7–3.5 seconds depending on benchmark.

Algorithm 1 DyPSI: Functional Tucker pre-training, joint diffusion training, and conditional sampling.

Phase 1 (Trajectory-supervision construction.). Construct pairs of field snapshots $\{\mathcal{P}^{(n)}_{1:T_n}\}$ and supervision sensor coordinates $\{\mathbf{s}^{(n)}_{1:T_n}\}$ from costy offline optimizer, and split into train/val/test. See details at Algorithm 2.

Phase 2: Functional Tucker pre-training.

1: Initialize shared spatial basis MLPs $\boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_{D+1}$, the temporal basis $\boldsymbol{\phi}_1$, and per-sample core tensors $\mathcal{W}^{\text{phy}}_n, \mathcal{W}^{\text{sen}}_n$.
2: For every training sample n and timestep t, compute the SPF $\mathcal{S}^{(n)}_t$ from $\mathbf{s}^{(n)}_t$ via Eq. (7).
3: Jointly optimize $\{\boldsymbol{\phi}_k\}$ and $\{\mathcal{W}^{\text{phy}}_n, \mathcal{W}^{\text{sen}}_n\}$ to minimize the dual reconstruction loss $\sum_n \big(\|\text{Tucker}(\mathcal{W}^{\text{phy}}_n, \boldsymbol{\phi}) - \mathcal{P}^{(n)}\|^2 + \|\text{Tucker}(\mathcal{W}^{\text{sen}}_n, \boldsymbol{\phi}) - \mathcal{S}^{(n)}\|^2\big)$ against arbitrary continuous-coordinate samples until convergence.
4: **Freeze** $\{\boldsymbol{\phi}_k\}$. Cache the resulting per-sample joint cores $\mathbf{x}^{(n)}_{\text{joint}} = \text{Concat}_{R_1}\big[\mathcal{W}^{\text{phy}}_n, \mathcal{W}^{\text{sen}}_n\big]$ (Eq. (9)).

Phase 3: Joint diffusion training.

1: Precompute the Matérn-$3/2$ kernel matrix $\mathbf{K}_{\text{time}} \in \mathbb{R}^{T_{\text{seg}} \times T_{\text{seg}}}$ and its Cholesky factor $\mathbf{L}$.
2: **for** step $= 1, \ldots, N_{\text{iter}}$ **do**
3: Sample a sample n and a length-T_{seg} segment i of $\mathbf{x}^{(n)}_{\text{joint}}$ with its observed prefix $\mathcal{H}^{(i)}$ of length K_{obs}.
4: Encode the prefix to $\mathbf{h}^{(i)}_{\text{hist}}$ with the history Transformer.
5: Draw $\sigma \sim p(\sigma)$ and $\boldsymbol{\xi}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; form $\boldsymbol{\eta}^{(i)} = (\mathbf{L} \otimes \mathbf{I})\,\text{vec}(\boldsymbol{\xi}^{(i)})$ (Eq. (10)).
6: Update θ by stochastic gradient on $\mathcal{L}_{\text{joint}}$ in Eq. (11).
7: **end for**

Phase 4: Conditional sampling and sensor extraction.

Require: observed prefix $\{(\mathbf{s}_\tau, \mathbf{y}_\tau)\}_{\tau=1}^{K_{\text{obs}}}$, sensor budget k.

1: Encode prefix $\to \mathbf{h}^{(i)}_{\text{hist}}$. Initialize the suffix at the highest noise level: $\mathbf{x}^{(i)}_R \sim \mathcal{N}(\mathbf{0}, \sigma_R^2 \mathbf{I})$.
2: **for** $r = R, \ldots, 1$ **do**
3: Soft-anchor the prefix: $\mathbf{x}^{(i)}_{r,\,1:K_{\text{obs}}} \leftarrow (1-\alpha_r)\mathbf{x}^{(i)}_{r,\,1:K_{\text{obs}}} + \alpha_r \widetilde{\mathbf{x}}^{(i)}_{\text{hist},r}$ (Eq. (12)).
4: Reverse step: $\mathbf{x}^{(i)}_{r-1} \leftarrow$ second-order Heun update of $\mathbf{x}^{(i)}_r$ using $D_\theta(\,\cdot\,; \sigma_r, \mathbf{h}^{(i)}_{\text{hist}})$.
5: **end for**
6: Decode $\mathbf{x}^{(i)}_0$ via the frozen Functional Tucker basis to recover $\widehat{\mathcal{P}}_{K_{\text{obs}}+1:T_{\text{seg}}}$ and $\widehat{\mathcal{S}}_{K_{\text{obs}}+1:T_{\text{seg}}}$.
7: **return** predicted fields $\{\widehat{\mathcal{P}}_t\}$ and sensor coordinates $\{\widehat{\mathbf{s}}_t\}$, where $\widehat{\mathbf{s}}_t$ is the top-k $\arg\max$ of $\widehat{\mathcal{S}}_t$ on Ω (§C.4).

D Full Experimental Results

This appendix reports the complete experimental results corresponding to the main-text comparisons. All stochastic experiments are repeated with five independent random seeds, and entries denote mean $\pm$ standard deviation. For the reconstruction benchmarks, statistical significance is assessed using paired t-tests between DyPSI and the strongest baseline in each setting on identical test cases. The main-text tables report mean values for compactness, while the following tables provide the complete variability and significance results.

Table 9 confirms that DyPSI achieves the lowest mean error in every turbulent flow and sea surface temperature configuration. The standard deviations are small relative to the corresponding improvements over PhySense-opt. All turbulent flow comparisons are significant at $p < 0.001$, and the sea surface temperature improvements remain significant with p-values between 0.004 and 0.010.

On car aerodynamics, Table 13 shows consistent improvements of 17.47% to 19.43% over PhySense-opt. Every comparison is significant at $p < 0.001$. The DyPSI standard deviations range from 0.0011 to 0.0014, indicating stable performance across all sensor budgets.

Table 10 provides the complete variability results for the accuracy and kinematic feasibility study. The Per-frame Oracle achieves the lowest reconstruction error but violates the velocity constraint in 75% to 92% of transitions and produces substantially larger smoothness costs. DyPSI reduces the violation rate to 4% to 9% while improving over static PhySense-opt and remaining close to the ground-truth-informed Sequence Teacher. Its runtime of 1.9 to 2.8 seconds per sequence is approximately three orders of magnitude lower than that of the optimization-based alternatives.

Table 11 shows that DyPSI-opt attains the lowest mean error across all teacher architectures and sensor budgets. Eleven of the twelve comparisons against PhySense-opt are significant at $p \leq 0.007$. The Senseiver comparison with 30 sensors is not statistically significant, with $p = 0.400$, and the corresponding mean reduction is substantially smaller than in the other settings. Overall, the results support generalization across reconstruction architectures while also identifying the configuration where the observed improvement is not statistically resolved.

Table 9: Full performance comparison on the turbulent flow and sea surface temperature benchmarks. Relative L_2 error is reported as mean $\pm$ standard deviation over five independent runs. Improvement quantifies the relative error reduction with respect to PhySense-opt. The p-values are obtained from paired t-tests comparing DyPSI with PhySense-opt on identical test cases.

		Turbulent Flow				Sea Temperature			
Models	Placement	#200	#100	#50	#30	#100	#50	#25	#15
SSPOR	SSPOR	0.3018 ± 0.0057	0.4429 ± 0.0071	0.5921 ± 0.0105	0.6637 ± 0.0118	0.0756 ± 0.0026	0.0752 ± 0.0029	0.0789 ± 0.0026	0.0719 ± 0.0034
VoronoiCNN	Random	0.3588 ± 0.0041	0.4870 ± 0.0069	0.6919 ± 0.0108	0.7992 ± 0.0130	0.1496 ± 0.0041	0.1537 ± 0.0048	0.2433 ± 0.0063	0.2817 ± 0.0071
Senseiver	Random	0.1842 ± 0.0032	0.2316 ± 0.0049	0.3740 ± 0.0069	0.5746 ± 0.0086	0.0715 ± 0.0020	0.0732 ± 0.0021	0.0769 ± 0.0027	0.0784 ± 0.0028
S^3GM	Random	0.1856 ± 0.0031	0.2016 ± 0.0035	0.2681 ± 0.0042	0.6421 ± 0.0098	0.1115 ± 0.0034	0.1137 ± 0.0038	0.1581 ± 0.0045	0.1795 ± 0.0050
DiffusionPDE	Random	0.1993 ± 0.0035	0.3312 ± 0.0061	0.6604 ± 0.0103	0.9163 ± 0.0125	0.0664 ± 0.0025	0.0775 ± 0.0031	0.0859 ± 0.0029	0.1139 ± 0.0032
PhySense	Random	0.1233 ± 0.0022	0.1527 ± 0.0028	0.2586 ± 0.0047	0.5176 ± 0.0083	0.0439 ± 0.0016	0.0452 ± 0.0019	0.0477 ± 0.0023	0.0520 ± 0.0014
PhySense-opt	Static Opt.	0.1106 ± 0.0018	0.1257 ± 0.0020	0.1558 ± 0.0027	0.2157 ± 0.0045	0.0426 ± 0.0015	0.0430 ± 0.0016	0.0437 ± 0.0021	0.0439 ± 0.0018
DyPSI	Dynamic Gen.	$\mathbf{0.0986 \pm 0.0013}$	$\mathbf{0.1101 \pm 0.0017}$	$\mathbf{0.1352 \pm 0.0024}$	$\mathbf{0.1826 \pm 0.0041}$	$\mathbf{0.0387 \pm 0.0012}$	$\mathbf{0.0390 \pm 0.0014}$	$\mathbf{0.0394 \pm 0.0019}$	$\mathbf{0.0402 \pm 0.0020}$
Improvement		10.87%	12.41%	13.19%	15.35%	9.15%	9.30%	9.84%	8.43%
p-value		< 0.001	< 0.001	< 0.001	< 0.001	0.004	0.004	0.005	0.010

Table 10: Full accuracy and kinematic feasibility comparison under representative sparse sensor budgets. Entries denote mean $\pm$ standard deviation over five independent runs. Viol. denotes the velocity violation rate, Smooth. denotes $\sum_t \|\mathbf{s}_{t+1} - 2\mathbf{s}_t + \mathbf{s}_{t-1}\|^2$, and runtime is measured in seconds per 64-frame sequence. The Per-frame Oracle and Sequence Teacher access future ground truth and perform test-time optimization, whereas DyPSI uses neither.

			Turbulent Flow, 50 sensors				Sea Surface Temperature, 25 sensors				Car Aerodynamics, 30 sensors			
Strategy	Future GT	Test-time Opt.	Rel. $L_2 \downarrow$	Viol. $\downarrow$	Smooth. $\downarrow$	Runtime $\downarrow$	Rel. $L_2 \downarrow$	Viol. $\downarrow$	Smooth. $\downarrow$	Runtime $\downarrow$	Rel. $L_2 \downarrow$	Viol. $\downarrow$	Smooth. $\downarrow$	Runtime $\downarrow$
Static PhySense-opt	×	×	0.1558 ± 0.0027	0	0	–	0.0437 ± 0.0018	0	0	–	0.0372 ± 0.0015	0	0	–
Per-frame Oracle	✓	✓	0.1007 ± 0.0012	$89\% \pm 5\%$	142.5 ± 13.8	887 ± 51	0.0259 ± 0.0015	$75\% \pm 3\%$	165.3 ± 14.2	628 ± 39	0.0201 ± 0.0017	$92\% \pm 3\%$	88.6 ± 9.5	1026 ± 103
Sequence Teacher	✓	✓	0.1206 ± 0.0009	$5\% \pm 1\%$	3.5 ± 0.9	1683 ± 143	0.0355 ± 0.0011	$2\% \pm 1\%$	4.5 ± 1.1	1429 ± 107	0.0251 ± 0.0012	$6\% \pm 2\%$	2.6 ± 0.7	1952 ± 165
DyPSI	×	×	0.1352 ± 0.0024	$8\% \pm 2\%$	4.2 ± 1.0	$\mathbf{1.9 \pm 0.2}$	0.0394 ± 0.0019	$4\% \pm 1\%$	9.2 ± 1.9	$\mathbf{2.1 \pm 0.2}$	0.0307 ± 0.0014	$9\% \pm 2\%$	3.1 ± 0.8	$\mathbf{2.8 \pm 0.3}$

Table 11: Full constructor/backbone generalization results on the turbulent flow benchmark. Relative L_2 error is reported as mean $\pm$ standard deviation over five independent runs. For non-DyPSI rows, column groups denote the reconstruction backbone used for evaluation. For DyPSI-opt, they denote the teacher used to construct offline trajectory supervision for a separately trained DyPSI model. The p-values compare DyPSI-opt with PhySense-opt in each setting.

	Senseiver				PhySense				DiffusionPDE			
Placement Strategy	#200	#100	#50	#30	#200	#100	#50	#30	#200	#100	#50	#30
Random Sample	0.1842 ± 0.0037	0.2316 ± 0.0043	0.3740 ± 0.0053	0.5746 ± 0.0090	0.1233 ± 0.0021	0.1527 ± 0.0031	0.2586 ± 0.0040	0.5176 ± 0.0073	0.1993 ± 0.0046	0.3312 ± 0.0052	0.6604 ± 0.0105	0.9163 ± 0.0143
Grid Sample	0.1615 ± 0.0030	0.2273 ± 0.0041	0.3845 ± 0.0062	0.5869 ± 0.0092	0.1235 ± 0.0029	0.1521 ± 0.0036	0.2436 ± 0.0047	0.4930 ± 0.0068	0.1932 ± 0.0031	0.3319 ± 0.0071	0.6703 ± 0.0093	0.9079 ± 0.0115
Min-Max Sample	0.1549 ± 0.0028	0.2091 ± 0.0038	0.3795 ± 0.0067	0.5747 ± 0.0095	0.1196 ± 0.0024	0.1533 ± 0.0020	0.3157 ± 0.0052	0.5022 ± 0.0041	0.1889 ± 0.0037	0.3307 ± 0.0067	0.6574 ± 0.0117	0.9010 ± 0.0132
PhySense-opt	0.1388 ± 0.0025	0.1750 ± 0.0032	0.3025 ± 0.0054	0.4570 ± 0.0079	0.1106 ± 0.0014	0.1257 ± 0.0021	0.1558 ± 0.0031	0.2157 ± 0.0037	0.1672 ± 0.0025	0.2841 ± 0.0052	0.5319 ± 0.0085	0.7904 ± 0.0120
DyPSI-opt	$\mathbf{0.1259 \pm 0.0022}$	$\mathbf{0.1670 \pm 0.0030}$	$\mathbf{0.2863 \pm 0.0048}$	$\mathbf{0.4527 \pm 0.0072}$	$\mathbf{0.0986 \pm 0.0018}$	$\mathbf{0.1101 \pm 0.0020}$	$\mathbf{0.1352 \pm 0.0028}$	$\mathbf{0.1826 \pm 0.0041}$	$\mathbf{0.1569 \pm 0.0029}$	$\mathbf{0.2640 \pm 0.0048}$	$\mathbf{0.5028 \pm 0.0075}$	$\mathbf{0.7116 \pm 0.0110}$
p-value	< 0.001	0.007	0.003	0.400	< 0.001	< 0.001	< 0.001	< 0.001	0.003	0.001	0.002	< 0.001

Table 12: Full gain decomposition on all reconstruction benchmarks. Errors are reported as mean $\pm$ standard deviation over five independent runs. Gain Retention is computed using Eq. (46). TF and SST denote turbulent flow and sea surface temperature, respectively.

	Turbulent Flow				Sea Surface Temperature				Car Aerodynamics			
Strategy	#200	#100	#50	#30	#100	#50	#25	#15	#100	#50	#30	#15
Static PhySense-opt	0.1106 ± 0.0018	0.1257 ± 0.0020	0.1558 ± 0.0027	0.2157 ± 0.0045	0.0426 ± 0.0015	0.0430 ± 0.0016	0.0437 ± 0.0021	0.0439 ± 0.0018	0.0370 ± 0.0012	0.0370 ± 0.0016	0.0372 ± 0.0015	0.0386 ± 0.0021
Sequence Teacher	0.0915 ± 0.0008	0.1025 ± 0.0015	0.1206 ± 0.0009	0.1695 ± 0.0019	0.0318 ± 0.0018	0.0347 ± 0.0011	0.0355 ± 0.0011	0.0371 ± 0.0016	0.0230 ± 0.0008	0.0247 ± 0.0017	0.0251 ± 0.0012	0.0273 ± 0.0019
DyPSI	0.0986 ± 0.0013	0.1101 ± 0.0017	0.1352 ± 0.0024	0.1826 ± 0.0041	0.0387 ± 0.0012	0.0390 ± 0.0014	0.0394 ± 0.0019	0.0402 ± 0.0020	0.0301 ± 0.0013	0.0302 ± 0.0011	0.0307 ± 0.0014	0.0311 ± 0.0014
Gain Retention	**0.63**	**0.67**	**0.59**	**0.72**	**0.36**	**0.48**	**0.52**	**0.54**	**0.49**	**0.55**	**0.54**	**0.66**

Table 13: Full results on the car aerodynamics benchmark. Relative L_2 error is reported as mean $\pm$ standard deviation over five independent runs. The p-values are obtained from paired t-tests comparing DyPSI with PhySense-opt.

Models	#200	#100	#50	#30	#15
Senseiver	0.1009 ± 0.0030	0.1012 ± 0.0031	0.1018 ± 0.0037	0.1022 ± 0.0030	0.1053 ± 0.0032
DiffusionPDE	0.0967 ± 0.0028	0.0966 ± 0.0028	0.0987 ± 0.0039	0.1055 ± 0.0031	0.2095 ± 0.0060
PhySense	0.0375 ± 0.0013	0.0382 ± 0.0013	0.0395 ± 0.0017	0.0416 ± 0.0015	0.0465 ± 0.0023
PhySense-opt	0.0369 ± 0.0012	0.0370 ± 0.0012	0.0370 ± 0.0016	0.0372 ± 0.0015	0.0386 ± 0.0021
DyPSI	$\mathbf{0.0301 \pm 0.0013}$	$\mathbf{0.0301 \pm 0.0013}$	$\mathbf{0.0302 \pm 0.0011}$	$\mathbf{0.0307 \pm 0.0014}$	$\mathbf{0.0311 \pm 0.0014}$
Improvement	18.43%	18.65%	18.39%	17.47%	19.43%
p-value	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

D.1 Gain Decomposition Relative to the Offline Teacher

We further quantify how much of the offline teacher's improvement over static placement is retained by the learned DyPSI generator. Static PhySense-opt, the Sequence Teacher, and DyPSI are evaluated on identical test sequences with the same frozen reconstructor. The Sequence Teacher has access to future ground-truth fields and performs per-sequence optimization, whereas DyPSI uses only historical observations and performs no test-time optimization. Let E_{static}, E_{teacher}, and E_{DyPSI} denote their respective relative L_2 errors. We define

$$\mathrm{GainRetention} = \frac{E_{\mathrm{static}} - E_{\mathrm{DyPSI}}}{E_{\mathrm{static}} - E_{\mathrm{teacher}}}. \tag{46}$$

This metric measures the fraction of the teacher's total error reduction over static placement that remains after amortizing the offline optimizer into the history-conditioned generator. The difference between static placement and the teacher reflects the value of dynamic sequence-level sensing with oracle information, while the difference between the teacher and DyPSI reflects the amortization gap introduced by removing future ground-truth access and online optimization.

Table 12 shows that DyPSI retains 36% to 72% of the teacher's gain across the twelve configurations, with an average retention of approximately 56%. The teacher consistently achieves lower error, as expected from its access to future ground truth and its substantially larger optimization budget. Nevertheless, DyPSI recovers roughly half of the available dynamic improvement without either advantage. This decomposition demonstrates that the gains arise from both sequence-level adaptive sensing and the successful transfer of the teacher's strategy to the learned generator, while the remaining error difference quantifies the cost of amortization.

E Construct Trajectory Supervision Data

Motivation. Public physics-sensing datasets—including the GLORYS12 reanalysis, OpenFOAM car-aerodynamics simulations, and the turbulent channel-flow snapshots used in §5—provide dense spatiotemporal fields but do not contain paired sensor trajectories. To train DyPSI's joint generative model, we require paired sequences $\{(\mathcal{P}_{1:T}, \mathbf{s}_{1:T})\}$ in which the sensing configurations are not only informative but also temporally coherent and kinematically feasible for mobile platforms. We therefore construct high-quality dynamic trajectories via an offline sequence-level optimization that can be instantiated on top of *any* differentiable reconstructor.

Model-agnostic trajectory optimization. Given a pre-trained reconstructor $\mathcal{M}_\theta$ (which may be PhySense, DiffusionPDE, Senseiver, or any other end-to-end differentiable model), a ground-truth field sequence $\{\mathcal{P}_t^{\mathrm{gt}}\}_{t=1}^T$, and an initial static placement $\mathbf{s}_1^\star$ obtained from any frame-wise optimizer (e.g., the PGD stage of PhySense-opt), we seek a dynamic trajectory $\mathbf{S}_{1:T} = \{\mathbf{s}_t\}_{t=1}^T$ that minimizes the composite objective

$$\mathcal{L}_{\mathrm{total}}(\mathbf{S}_{1:T}) = \underbrace{\mathcal{L}_{\mathrm{recon}}}_{\text{reconstruction fidelity}} + \lambda \underbrace{\mathcal{L}_{\mathrm{GP}}}_{\text{temporal coherence}} + \gamma \underbrace{\mathcal{L}_{\mathrm{dyn}}}_{\text{kinematic feasibility}}. \tag{47}$$

The reconstruction term measures the cumulative sparse-to-dense error over the sequence:

$$\mathcal{L}_{\text{recon}} \;=\; \sum_{t=1}^{T} \big\| \mathcal{M}_\theta\big(\mathcal{P}_t^{\text{gt}}(\mathbf{s}_t);\, \mathbf{s}_t\big) \;-\; \mathcal{P}_t^{\text{gt}} \big\|_2^2, \tag{48}$$

where $\mathcal{P}_t^{\text{gt}}(\mathbf{s}_t)$ denotes the sparse point observations sampled at coordinates $\mathbf{s}_t$. Because $\mathcal{M}_\theta$ is differentiable, gradients flow end-to-end from the reconstruction loss back to the sensor coordinates; when the reconstructor operates on the Sensor Position Field surrogate $\mathcal{S}_t$ (§3.2), the map $\mathbf{s}_t \to \mathcal{S}_t$ in Eq. (7) supplies a fully differentiable bridge that avoids discrete argmax operations during optimization.

The temporal term is the GP log-prior adopted in DyPSI's diffusion training (Eq. (10)):

$$\mathcal{L}_{\text{GP}} \;=\; \frac{1}{2} \sum_{i=1}^{k} \big(\mathbf{s}_{1:T}^{(i)} - \mathbf{m}\big)^\top \mathbf{K}^{-1} \big(\mathbf{s}_{1:T}^{(i)} - \mathbf{m}\big), \tag{49}$$

where $\mathbf{s}_{1:T}^{(i)} \in \mathbb{R}^{T\times 2}$ is the trajectory of the i-th sensor, $\mathbf{m}$ is a reference path (we use the initial static placement as mean), and $\mathbf{K} \in \mathbb{R}^{T\times T}$ is the Matérn-$3/2$ Gram matrix with length scale $\ell = 4$ frames and signal variance $\sigma_{\text{GP}}^2 = 1$, identical to the kernel used in §3.3. Crucially, this is the *same* kernel family analysed in Proposition 2 (§3.4): its quadratic form furnishes a principled upper bound on the discrete frame-to-frame variation of the trajectory, thereby guaranteeing kinematic feasibility without any ad hoc smoothing penalty.

The dynamics term enforces hard platform constraints via hinge losses on velocity and acceleration:

$$\mathcal{L}_{\text{dyn}} \;=\; \sum_{t=1}^{T-1} \text{ReLU}\Big(\|\mathbf{s}_{t+1} - \mathbf{s}_t\| - v_{\max}\Delta t \Big)^2 \;+\; \sum_{t=2}^{T-1} \text{ReLU}\Big(\|\mathbf{s}_{t+1} - 2\mathbf{s}_t + \mathbf{s}_{t-1}\| - a_{\max}\Delta t^2 \Big)^2, \tag{50}$$

where $v_{\max}$ and $a_{\max}$ are platform-specific speed and acceleration limits. The full procedure is summarized in Algorithm 2.

Algorithm 2 Sequence-Level Sensor-Trajectory Optimization (Offline Data Generator)

Require: Pre-trained reconstructor $\mathcal{M}_\theta$; ground-truth fields $\{\mathcal{P}_t^{\text{gt}}\}_{t=1}^T$; initial static placement $\mathbf{s}_1^\star$ from PhySense-opt (or any frame-wise optimizer); sensor budget k; Matérn-$3/2$ kernel matrix $\mathbf{K}$ with Cholesky factor $\mathbf{L}$; hyperparameters $\lambda, \gamma, v_{\max}, a_{\max}$; step size η; iterations N_{iter}.

Ensure: Optimized dynamic trajectory $\mathbf{S}_{1:T} = \{\mathbf{s}_t\}_{t=1}^T$.

Initialize $\mathbf{s}_1 \leftarrow \mathbf{s}_1^\star$; for $t = 2, \dots, T$ initialize $\mathbf{s}_t$ by optical-flow advection from $\mathbf{s}_{t-1}$ or by sampling from the GP prior.
for iteration $= 1, \dots, N_{\text{iter}}$ **do**
  $\mathcal{L}_{\text{recon}} \leftarrow \sum_{t=1}^{T} \big\| \mathcal{M}_\theta\big(\mathcal{P}_t^{\text{gt}}(\mathbf{s}_t); \mathbf{s}_t\big) - \mathcal{P}_t^{\text{gt}} \big\|_2^2$
  $\mathcal{L}_{\text{GP}} \leftarrow \frac{1}{2} \sum_{i=1}^{k} \big\| \mathbf{L}^{-1}\big(\mathbf{s}_{1:T}^{(i)} - \mathbf{m}\big) \big\|_2^2$
  $\mathcal{L}_{\text{dyn}} \leftarrow$ velocity&acceleration hinge losses via Eq. (50)
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{recon}} + \lambda\, \mathcal{L}_{\text{GP}} + \gamma\, \mathcal{L}_{\text{dyn}}$
  $\mathbf{S}_{1:T} \leftarrow \mathbf{S}_{1:T} - \eta\, \nabla_{\mathbf{S}} \mathcal{L}_{\text{total}}$
  Project each $\mathbf{s}_t$ onto the feasible domain Ω (e.g., land masking for sea temperature, mesh surface for car aerodynamics).
end for
return $\mathbf{S}_{1:T}$

Remarks on generality. The pipeline above is deliberately model-agnostic: it only requires that $\mathcal{M}_\theta$ admit gradients with respect to the sensing coordinates (or their SPF surrogate). Consequently, the same offline generator can be paired with PhySense, DiffusionPDE, Senseiver, VoronoiCNN, or any future reconstructor without architectural modification. The initial static placement $\mathbf{s}_1^\star$ may likewise be produced by any frame-wise optimizer—PhySense's PGD stage, Diff-SPORT's SHAP-guided MAP-GA, or even a simple greedy A-optimal design—because the sequence-level GP prior and dynamics constraints are applied *after* initialization and are independent of the static optimizer's internals.

Validation. Table 14 compares the reconstruction accuracy of a fixed pre-trained PhySense reconstructor under two sensing protocols on the training corpus: (i) static placements optimized by PhySense-opt's PGD stage across all frames, and (ii) dynamically optimized trajectories produced by Algorithm 2. The dynamic trajectories consistently outperform their static counterparts, with relative error reductions of approximately 20% on the two grid benchmarks and 28.92% on the unstructured car mesh, confirming that temporal coherence and kinematic feasibility are not merely physical desiderata but also measurably improve the information content of the sparse observations used for training.

Table 14: Training-set reconstruction error (↓) of a fixed PhySense reconstructor under static placements versus dynamically optimized trajectories (Algorithm 2). The relative improvement quantifies the gain attained by sequence-level trajectory optimization in the data-preparation stage.

Benchmark	#Sensors	Static (PhySense-opt)	Dynamic (Ours)	Relative Improvement
Turbulent Flow	50	0.1558	0.1206	22.59%
Sea Temperature	50	0.0430	0.0345	19.77%
Car Aerodynamics	50	0.0370	0.0263	28.92%

Figure 4 visualizes representative trajectories generated by Algorithm 2 on the turbulent-flow benchmark. Each panel tracks one mobile sensor across frames 0–9; the color gradient from blue to red encodes temporal progression. The paths exhibit smooth, physically feasible vertical migration that tracks coherent flow structures, in contrast to the discontinuous "teleporting" behavior produced by independent per-frame optimization.

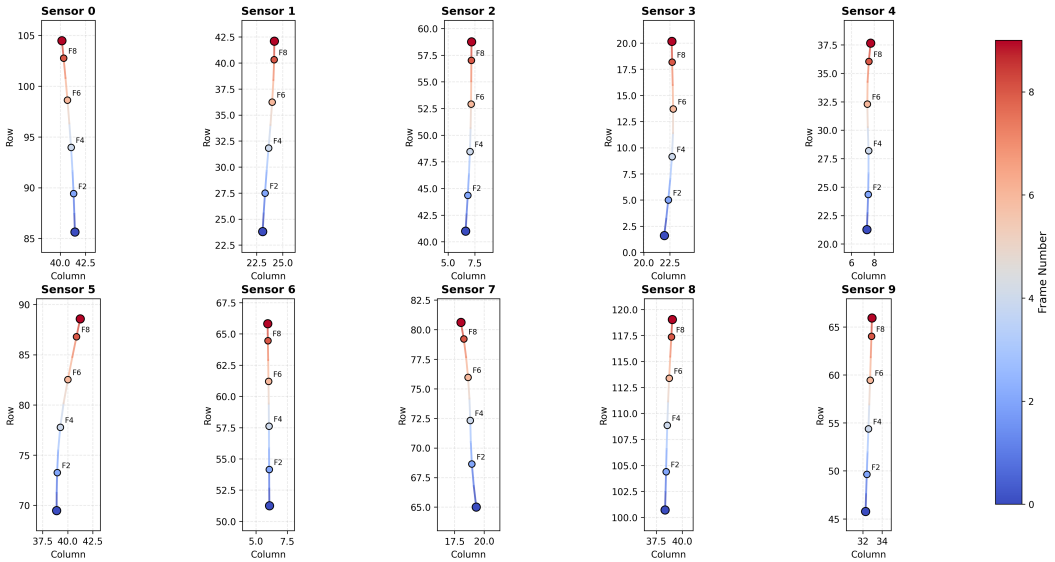

Figure 4: Optimized sensor trajectories on the turbulent-flow benchmark (training-set visualization). Each panel tracks one mobile sensor across frames 0–9; color encodes temporal progression from early (blue) to late (red). The smooth vertical migration demonstrates the effect of the Matérn-$3/2$ GP prior and the kinematic constraints in Eq. (50).

F Broader Impacts

DyPSI advances adaptive physical sensing by enabling temporally coherent, kinematically feasible sensor trajectories that track evolving phenomena, with direct benefits for climate monitoring, oceanographic surveys, and industrial diagnostics. By maximizing information extraction from sparse mobile sensor budgets, our framework can reduce the operational costs and environmental footprint of autonomous sensing campaigns conducted by underwater vehicles, drifting buoys, or aerial platforms. We encourage the responsible deployment of such adaptive sensing systems to

enhance scientific understanding and public safety, while remaining mindful of potential dual-use concerns in unauthorized tracking or surveillance scenarios.

NeurIPS Paper Checklist

Acknowledgments and Disclosure of Funding

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the joint field–trajectory formulation, the DyPSI method, and the theoretical and empirical contributions, with evaluation on three benchmarks. The experimental claims concern improvements over random and static optimized placements and the accuracy–efficiency trade-off relative to ground-truth-informed per-frame and sequence-level oracle optimization, consistent with §3.4 and §5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses teacher dependence and the amortization gap, the training-support assumption and out-of-distribution deployment, history-conditioned open-loop evaluation, the restriction to selected kinematic and domain constraints rather than full platform dynamics, SPF peak collisions at high sensor densities, and the point-wise observation assumption.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are numbered and cross-referenced. Propositions 1–2 in the main text (§3.4) are proved formally in Appendix A as Propositions A.6 and A.8. Assumptions A.1–A.3 (conditional Gaussian posterior, uniform spectral bound, admissible EDM weighting) are stated explicitly in Appendix A, and detailed proof sketches using spectral decomposition and generalized Rayleigh quotients are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.

- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: §5 describes the benchmarks, baselines, and evaluation protocol. Appendix C provides the metric (C.1), architectures (C.2), optimization and diffusion settings (C.3), sampling procedure (C.4), end-to-end algorithm (C.5), per-benchmark hyperparameters in Table 8 (C.6), and compute resources (C.7); the public repository provides configuration files and reproduction scripts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The public code repository is available at `https://github.com/mengqi132/DyPSI`, including implementation code, dependency and configuration files, data-preparation instructions, and scripts for the main experimental pipeline. The raw datasets remain available from their original public sources, and the repository documents data access, preprocessing, training, and inference procedures.

   Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details are specified in §5 and Appendix C, including data splits, sensor budgets, optimizer and sampler settings, segment lengths, model ranks, and per-benchmark hyperparameters in Table 8; executable settings are provided in the public repository's configuration files.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For each benchmark and sensor budget, we independently train five models with different random seeds and evaluate each trained model on the same held-out test set. Appendix D reports the mean $\pm$ standard deviation across the five independent training runs and paired t-tests on matched test-case results against the strongest baseline; the reported variability therefore captures training-run variability rather than only stochastic sampling at inference.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.7 explicitly states that all experiments were run on a single NVIDIA A100-80GB GPU, lists per-benchmark training times, and reports inference latency.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses public physics-sensing benchmarks and simulations (turbulent channel flow, GLORYS12 reanalysis, and OpenFOAM car aerodynamics on ShapeNet) and involves no human subjects, crowdsourcing, or sensitive personal data. Appendix F discusses potential societal impacts and dual-use considerations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix F discusses potential benefits for climate monitoring, oceanographic surveys, and industrial diagnostics, including reduced sensing costs and environmental footprint. It also acknowledges potential dual-use concerns involving unauthorized tracking or surveillance and encourages responsible deployment.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper introduces a physics-sensing reconstruction and trajectory optimization framework. It does not release pre-trained language models, image generators, or large-scale scraped datasets that carry high risk of misuse or dual-use, so safeguards are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the original papers producing the datasets and software, and the public repository documents their sources. However, explicit version and license or terms-of-use information is not yet provided for every existing asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented in the public repository at `https://github.com/mengqi132/DyPSI`, which includes a README, dependency specifications, configuration files, usage instructions, and an MIT license. No new dataset or model checkpoint is released as part of this work.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects; all experiments are conducted on standard physics-sensing benchmarks and simulations.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subjects are involved in this work; therefore, IRB approval or equivalent ethical review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as an important, original, or non-standard component of the core methodology, scientific rigor, or originality of this research. Any writing or formatting assistance did not impact the core technical contributions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.