SELF-SUPERVISED LEARNING OF TASK-AGNOSTIC REPRESENTATIONS FOR PHYSICAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Foundation models for spatio-temporal physical systems seek representations that generalize across systems, parameters, and operating conditions. We explore a self-supervised route to this goal, decoupling representation learning from task-specific prediction to capture reusable properties of physical dynamics. We study masked reconstruction in physical space and JEPA-style prediction in latent space, and introduce regime fingerprinting to capture trajectory-level properties of the dynamics: an early partial observation must identify a broader view of the same trajectory, thereby encouraging the preservation of information that persists across time. The learned representations provide explicit, compact conditioning signals for downstream models and support efficient in-context adaptation to previously unseen physical systems. Experiments on heterogeneous 2D and 3D spatio-temporal dynamics across multiple downstream tasks show strong performance in few-shot settings, together with computationally efficient cross-physics generalization.

1 INTRODUCTION

Neural surrogates for time-dependent partial differential equations (PDEs) have become an efficient alternative to numerical solvers, with inference several orders of magnitude faster once the model is trained (Sanchez-Gonzalez et al., 2018; Li et al., 2021; Brandstetter et al., 2023). A recent direction is the development of foundation models for physics, in which a single large model is trained across many physical systems to predict future states from a window of past observations (Herde et al., 2024; Ohana et al., 2024; Morel et al., 2025). The goal is to obtain models that transfer to new physical systems, parameter regimes, or operating conditions with little or no additional training (Subramanian et al., 2023; Cao et al., 2026).

While next-state prediction is a natural training objective, it is not necessarily aligned with the broader goal of learning reusable representations of physical dynamics. Forecasting implicitly combines two distinct problems: identifying the underlying dynamical regime from observations and using this information to answer the specific forward query of what happens next. A forecasting loss directly supervises only the latter and does not require the inferred properties of the system to be explicitly represented or accessible. As a result, information about the underlying dynamics may remain entangled with the forecasting mechanism and cannot be readily reused for other downstream tasks. This design can also increase inference cost, since adaptation to a new system typically requires repeatedly conditioning on a history of observed states rather than on a compact representation of the inferred regime.

In this work, inspired by advances in self-supervised learning (Caron et al., 2021; Oquab et al., 2024), we explore an alternative paradigm for building physical foundation models by separating the encoding of system characteristics from downstream prediction tasks. An encoder is first pretrained with a self-supervised objective across a diverse set of physical systems to extract a representation of the underlying dynamics from observations. This representation is then used to condition surrogate models trained separately for downstream tasks (Zhou & Farimani, 2024).

This paradigm offers several potential advantages. First, because the representation is task-agnostic, the same encoder can be reused across a variety of tasks, including parameter estimation (Qu et al., 2026), spatial and temporal super-resolution (Zhao et al., 2025), forecasting, and in-context adaptation (Yang et al., 2023; Koupaï et al., 2025). Second, unlike many current foundation models that infer the dynamical regime from a history of past states during autoregressive prediction, our approach

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

disentangles system identification from state evolution. The physical regime is inferred once from a short observation window, after which forecasting proceeds from a single state conditioned on the resulting representation. This substantially reduces the cost of conditioning, which no longer grows with the rollout horizon. Finally, the same mechanism provides a simple way to adapt to unseen dynamics: the encoder extracts a context from a few observations of the new system, which can then be used by the downstream surrogate without retraining.

Our pretraining strategy draws inspiration from recent advances in self-supervised learning for image and video modeling. More specifically, we adapt Joint Embedding Predictive Architectures (JEPA) (Assran et al., 2023) to the setting of physical dynamics. This raises two main challenges. First, the data used to build physical foundation models are highly heterogeneous: different systems involve different variables and fields (e.g., velocity, pressure, density), coordinate systems (e.g., Cartesian or polar), spatial dimensionalities (2D or 3D), and resolutions. This contrasts with image and video models, which typically operate on a fixed and substantially more homogeneous input structure. Second, self-supervised learning for images and videos primarily relies on masking strategies designed to capture spatial structure or short-term temporal dynamics. Our objective is different: we seek to identify latent factors that remain stable throughout a trajectory, characterize persistent properties of the physical regime, and govern its evolution. Learning such a representation of the underlying dynamical regime constitutes the main challenge of our approach. In classical numerical modeling, this information is specified explicitly through the governing equations, physical parameters, forcing terms, boundary conditions, and geometry. In purely data-driven settings, by contrast, these characteristics are not given to the model and must be inferred directly from observations. A broader discussion of physical foundation models, self-supervised representation learning, and related approaches to learning transferable representations for physical systems is provided in Appendix A.

We propose a self-supervised framework for multi-physics data that accommodates an arbitrary number of fields in both 2D and 3D, together with a latent-space self-supervised objective that we call *regime fingerprinting*, reflecting its goal of extracting persistent dynamical characteristics directly from data. Given a temporally sparse view of a trajectory, the encoder is trained to produce a representation that matches that obtained from a longer view of the same trajectory (van den Oord et al., 2018a). Since the two views share the same underlying dynamics while containing different states, the objective encourages the encoder to capture properties that characterize the trajectory as a whole. We evaluate the resulting representations on two main tasks: physical-parameter estimation, used as a proxy for assessing how much information about the underlying dynamics is encoded, and forecasting, which remains a central task in dynamical-system modeling. The proposed approach is backbone-agnostic and can be instantiated with different architectures. We experiment with two representative backbones and compare our approach with conventional autoregressive foundation-model baselines in terms of predictive accuracy, inference cost, in- and out-of-distribution generalization, and zero-shot performance.

2 METHOD

2.1 SETTING AND NOTATION

We consider a heterogeneous collection $\mathcal{S}$ of time-dependent parametric PDE systems. Each system $s \in \mathcal{S}$ governs a state $\mathbf{u} : \Omega_s \times [0, T_s] \to \mathbb{R}^{V_s}$ defined as V_s scalar fields (we consider each scalar field components of vector and tensor fields) on a spatial domain $\Omega_s \subset \mathbb{R}^{d_s}$ of dimension $d_s \leq 3$. The evolution of this state is dictated by the partial differential equation:

$$\partial_t \mathbf{u} + \mathcal{N}_s[\mathbf{u}; \mathbf{c}, \mathbf{f}] = 0 \qquad \text{in } \Omega_s \times (0, T_s], \tag{1}$$

$$\mathcal{B}_s[\mathbf{u}] = 0 \qquad \text{on } \partial\Omega_s \times [0, T_s], \tag{2}$$

$$\mathbf{u}(\cdot, 0) = \mathbf{u}_0 \qquad \text{in } \Omega_s, \tag{3}$$

where $\mathcal{N}_s$ is a system-specific operator with coefficients $\mathbf{c}$ and forcing $\mathbf{f}$, $\mathcal{B}_s$ the boundary operator, and the initial condition $\mathbf{u}_0$. For each system s, we observe a dataset $\mathcal{D}$ of N_s trajectories $\{\mathbf{u}^{(i)}\}_{i=1}^{N_s}$ discretized on a uniform spatiotemporal grid. We denote by T the temporal resolution and by $(S_1, \ldots, S_{d_s})$ the spatial resolutions. Each observed trajectory is represented as $\mathbf{u}^{(i)} \in \mathbb{R}^{T \times S_1 \times \cdots \times S_{d_s} \times V_s}$.

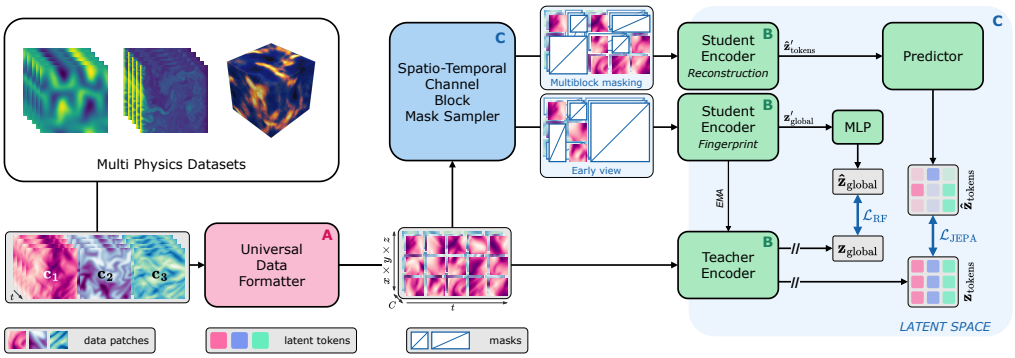


Figure 1: **Overview of the self-supervised learning framework.** A universal data formatter (**A**) maps inputs with different spatial dimensions and fields to a unified patches format. A student and teacher ViT encoders (**B**) are trained through masked reconstruction and alignment in latent space (**C**). Both students share the same weights while the teacher is an EMA of the students. Each encoder transforms the input patches into $\mathbf{z}_{\text{tokens}}$ alongside a $\mathbf{z}_{\text{global}}$ vector.

Regime. We define the *regime* of a trajectory as the trajectory-level physical characteristics that govern its evolution and remain fixed independently of the initial condition $\mathbf{u}_0$. This includes the system identity s, the coefficients $\mathbf{c}$, and the forcing $\mathbf{f}$: $r = (s, \mathbf{c}, \mathbf{f})$. Trajectories may therefore share the same regime while differing in their initial conditions.

2.2 Overall Framework

We train a multiphysics encoder with task-agnostic self-supervised objectives and evaluate whether the resulting representations capture the physical regime and can be reused for downstream tasks. In 2.3, we introduce the encoder E (Fig. 1, **B**), which operates on trajectories converted into a universal patch-based representation (Fig. 1, **A**). The encoder produces both local patch-level tokens and global tokens summarizing the trajectory. In 2.4, we describe the two latent-space pretraining objectives and their associated masking strategies (Fig. 1, **C**).

After pretraining, the encoder is frozen and used only as a context extractor; we then refer to it as **TRACE** (**T**rajectory **R**egime-**A**ware **C**onditioning **E**ncoder). Given a trajectory $\mathbf{u}$ of shape $[T, S_1, \ldots, S_{d_s}, V_s]$, TRACE produces local tokens $\mathbf{z}_{\text{tokens}}$ of shape $[\frac{T}{p_t}, \frac{S_1}{p}, \ldots, \frac{S_{d_s}}{p}, V_s]$ and n_g global tokens $\mathbf{z}_{\text{global}}$. For downstream applications, we apply an attentive pooling probe—two cross-attention layers with a single learnable query—to extract a regime-level representation from $\mathbf{z}_{\text{global}}$ or field-level representations from the corresponding components of $\mathbf{z}_{\text{tokens}}$.

2.3 Unified Encoding of Heterogeneous Trajectories

Universal Patchification. Each field of a trajectory is divided into spatio-temporal patches of fixed size (p_t, p, p, p), with $p_t = 2$ and $p = 16$. Resolutions that are not multiples of the patch size are zero-padded. For systems with $d_s < 3$, the data are lifted to three spatial dimensions by repeating values along the missing axes, each with width p. A trajectory $\mathbf{u} \in \mathbb{R}^{T \times S_1 \times \cdots \times S_{d_s} \times V_s}$ is thus represented as patches $\mathbf{u}_{\text{patches}} \in \mathbb{R}^{\frac{T}{p_t} \times \frac{S_1}{p} \times \cdots \times \frac{S_{d_s}}{p} \times V_s \times d_p}$, with $d_p = p_t \cdot p^3$.

The first stage of the encoder is a tokenizer that projects each input patch into a token and adds positional information encoding its temporal and spatial location, together with an ordinal field index. Importantly, the model is not given the semantic identity of the fields (e.g., pressure, velocity, or concentration): the index only distinguishes them. Their physical role must therefore be inferred from their observed dynamics and interactions with the other fields.

These choices allow a single set of weights to process arbitrary resolutions, spatial dimensionalities up to 3D, and numbers of fields. In contrast, existing scientific foundation models often rely on dedicated parameters for different dimensionalities—e.g., separate 2D and 3D tokenizers in Walrus

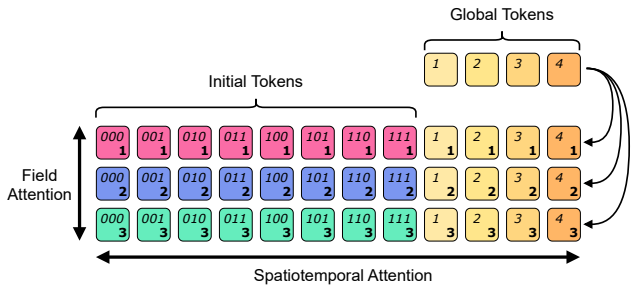


Figure 2: **Axial attention and global tokens.** A copy of the global tokens is added to each field along with the field ordinal in **bold** ; the initial tokens are flattened along each field, position encodings are in *italic* ; each attention is performed along its corresponding axis.

(McCabe et al., 2026)—or for different fields, through learnable field embeddings (Holzschuh et al., 2025) or field-wise linear projections (McCabe et al., 2024).

Axial attention and global tokens. The tokens are then processed by a Transformer architecture via attention (Vaswani et al., 2023). Since attending jointly over all field–spatiotemporal locations scales quadratically with the number of tokens, we factorize attention (Ho et al., 2019): attention is first applied across space and time within each field, and then across fields at each location. Our implementation efficiently supports arbitrary observation masks without padding; details are provided in Section C.

The encoder additionally learns n_g global tokens (Darcet et al., 2024), which are not tied to any specific location or field. Before the first attention block, a copy of these global tokens is added to the sequence of initial tokens for each field, together with the corresponding field ordinal (see Fig. 2). After the axial attention blocks, the transformed copies are averaged across the field axis, producing n_g encoded global tokens alongside the local encoded tokens. Details are in Section C.

The joint use of local and global representations is a key architectural contribution of our approach. Local tokens retain information tied to specific fields and spatiotemporal regions, whereas global tokens aggregate information across the trajectory. This separation is particularly relevant in our setting, where the encoder must capture not only local states and their evolution, but also trajectory-level characteristics of the underlying physical regime. In contrast to standard image or video representation learning, our objective explicitly requires identifying such persistent global properties of the dynamics.

2.4 Latent Objectives

We aim to learn general-purpose representations and therefore adopt latent-space objectives based on Joint Embedding Predictive Architectures (JEPA) (Assran et al., 2023; Bardes et al., 2024). A student encoder E_θ, given partial observations, is trained to predict latent targets produced by a teacher encoder E_ξ from more complete observations (Fig. 1, **C**). Both encoders share the same architecture, and the teacher weights ξ are updated as an exponential moving average (EMA) of the student weights θ. Our complete objective combines two complementary terms, $\mathcal{L} = \mathcal{L}_{\text{JEPA}} + \lambda\mathcal{L}_{\text{RF}}$, which we define below. $\mathcal{L}_{\text{JEPA}}$ targets local spatio-temporal structure by predicting masked latent content, whereas $\mathcal{L}_{\text{RF}}$ targets global, trajectory-level information associated with the underlying physical regime. Each objective uses a different masking strategy for the student input as detailed below. The two masked views are processed independently by the encoder and optimized jointly during pretraining.

2.4.1 Masked Latent Prediction

The first objective learns local spatio-temporal representations by reconstructing masked latent tokens from their observed context.

Masking. A mask is defined as a union of spatio-temporal inter-field blocks $M = \cup_k B_k$ that can span multiple contiguous timesteps, spatial positions and fields :

$$B_k = [t_k, t_k{+}\Delta t_k] \times [x_k, x_k{+}\Delta x_k] \times [y_k, y_k{+}\Delta y_k] \times [z_k, z_k{+}\Delta z_k] \times \mathcal{V}_k, \tag{4}$$

where $\mathcal{V}_k \subseteq \{1, \dots, V_s\}$ denotes a subset of physical variables. We mask up to a total of 85% of the token grid.

Objective. This objective corresponds to the original JEPA formulation ; incomplete unmasked patches encoded by the student $E_\theta^{\text{tok}}(\mathbf{u}_{\bar{M}})$ are used by a latent predictor P_ϕ to reconstruct the missing tokens ; this predictor is implemented as a ViT that decodes mask tokens along the given encoded tokens. This reconstruction is evaluated against the teacher's encoded complete view $E_\xi^{\text{tok}}(\mathbf{u})$ by a token-wise loss on the masked indices M :

$$\mathcal{L}_{\text{JEPA}} \coloneqq \mathbb{E}_{\mathbf{u}\sim\mathcal{D}, M\sim\mathcal{M}_{\text{JEPA}}} \left[\frac{1}{|M|} \sum_{m\in M} \left\| \left(P_\phi \circ E_\theta^{\text{tok}}(\mathbf{u}_{\bar{M}})\right)_m - \text{sg}\left(E_\xi^{\text{tok}}(\mathbf{u})\right)_m \right\|_2^2 \right], \tag{5}$$

with sg the stop gradients operation.

2.4.2 Regime fingerprinting

The second objective complements local representation learning by explicitly targeting global, trajectory-level information. This constitutes a key methodological contribution of our approach. As defined in 2.1, the physical regime is a global property of a trajectory that is not explicitly targeted by the localized patch-wise objective in 5. We therefore introduce a dedicated objective on the global tokens, designed to encourage the representation to capture regime-level information from only a short partial observation.

Masking. For the RF objective, only the early temporal part of the trajectory is retained: the first 30% of temporal patches remain available to the student, while all later patches are masked. Within this early window, we additionally apply the same masking strategy as in 2.4.1. This creates a short and partial student view, while the teacher observes the complete trajectory.

Objective. The global tokens produced by the student encoder from the early, partial view $E_\theta^{\text{glob}}(\mathbf{u}_{\bar{M}})$ are passed through a two-layer MLP head H_ϕ and aligned with the global tokens produced by the teacher from the complete view $E_\xi^{\text{glob}}(\mathbf{u})$ through a contrastive loss (van den Oord et al., 2018b):

$$\mathcal{L}_{\text{RF}} \coloneqq \mathbb{E}_{\mathbf{u}\sim\mathcal{D}, M\sim\mathcal{M}_{\text{RF}}} \left[-\log \frac{\exp\left(\text{sim}\left[H_\phi \circ E_\theta^{\text{glob}}(\mathbf{u}_{\bar{M}}), \text{sg}(E_\xi^{\text{glob}}(\mathbf{u}))\right]/\kappa\right)}{\sum_{\mathbf{v}\in\mathcal{D}} \exp\left(\text{sim}\left[H_\phi \circ E_\theta^{\text{glob}}(\mathbf{u}_{\bar{M}}), \text{sg}(E_\xi^{\text{glob}}(\mathbf{v}))\right]/\kappa\right)} \right], \tag{6}$$

where $\text{sim}(\cdot,\cdot)$ is a centered cosine similarity and κ is the temperature.

We refer to this alignment of global representations between early partial observations and complete trajectories as Regime Fingerprinting (RF). Unlike masked latent prediction, which primarily captures local spatio-temporal structure, RF explicitly encourages the encoder to capture persistent characteristics of the underlying dynamics. Trajectories generated under different physical regimes may have similar initial states and remain difficult to distinguish at early times, while diverging substantially later in their evolution. By requiring an early partial view to identify the representation of the corresponding complete trajectory, RF encourages the encoder to extract subtle regime-dependent information that is informative about the trajectory as a whole.

3 Experiments

We evaluate whether the self-supervised representation captures information about physical dynamics and whether this information is useful downstream. Our experiments proceed in three stages. We first probe the frozen encoder using quantitative parameter regression and qualitative visualization to assess whether it captures the underlying physical regimes. We then condition two forecasting backbones on the learned representation, testing whether it can replace part of their temporal context

Table 1: **Regime recovery for representative pretraining objectives.** Probe parameter regression (RMSE on normalized targets, mean over three probe seeds; lower is better; bold: best per column). The complete objective study is in Table 9.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Objective	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
MAE (field tube)	5.72E-2	1.89E-2	2.54E-2	1.37E-1	**7.75E-1**	1.49E-1	4.06E-1	3.81E-3	3.98E-2
JEPA (multiblock)	5.64E-2	1.58E-2	2.31E-2	8.31E-2	8.03E-1	1.50E-1	**3.93E-1**	5.51E-3	4.39E-2
TRACE (JEPA + RF)	**4.89E-2**	**1.46E-2**	**1.88E-2**	**7.79E-2**	8.77E-1	**1.47E-1**	4.16E-1	**3.34E-3**	**1.83E-2**

to accelerate inference while preserving or improving performance. Finally, controlled replacements of the conditioning vector test whether the downstream forecasters actively exploit the physical information contained in the learned representation. Detailed protocols and per-dataset results are reported in Sections E to G.

3.1 Experimental setup

Data and pretraining. We pretrain on nine datasets from The Well (Ohana et al., 2024), spanning 2D and 3D systems with different governing equations, physical fields, and resolutions (Section B). All tested encoders share the same 88M-parameter base architecture. As detailed in Section 3.2, we selected the pretraining objective by comparing short ablation runs of this architecture (Table 8). Having identified masked latent prediction (JEPA) with regime fingerprinting ($\lambda = 0.03$) as the strongest objective, we train our final model, TRACE, using this configuration for a full 100,000 steps at batch size 32 (Section E). TRACE is then frozen for all downstream experiments.

Metrics. Parameter regression is evaluated with RMSE on normalized targets (predicting the training mean yields an RMSE of 1), while forecasting uses per-field NRMSE. Forecasts are 20-step autoregressive rollouts, evaluated on the first predicted frame and over steps 1–10 and 11–20. Full metric equations and evaluation protocols are given in the appendix.

3.2 Recovering physical regimes

To assess how well a representation captures the underlying dynamics, we evaluate how accurately the physical regime can be recovered from it, using physical-parameter prediction as a proxy. We freeze the pretrained encoder and train a lightweight, dataset-specific probe consisting of an attentive-pooling module followed by an MLP head (Section E.3). The probe predicts nine governing parameters across four systems: the Prandtl and Rayleigh numbers for Rayleigh–Bénard, α and ζ for Active Matter, T_0, z, and ρ_0 for Turbulence Gravity Cooling (TGC), and the Mach numbers Ma and Ms for MHD 64. We use this probing setup both to select the pretraining objective and to compare the final model with existing baselines.

Pretraining objective and configuration selection. We proceed in two stages. First, we compare several self-supervised pretraining objectives using the frozen-probe protocol described above. Masked latent prediction (JEPA) and masked reconstruction (MAE) provide the strongest parameter recovery overall (Table 9), with JEPA outperforming MAE on five of the nine parameters (Table 1). We then augment JEPA with several auxiliary objectives, including regime fingerprinting, which combines local and global latent information through a contrastive objective. The combination of JEPA and regime fingerprinting gives the strongest results on a majority of parameters (Tables 1 and 9), and we therefore select $\mathcal{L} = \mathcal{L}_{\mathrm{JEPA}} + \lambda \mathcal{L}_{\mathrm{RF}}$. Second, having selected the pretraining objective, we tune its configuration. A sweep over the relative weight λ selects $\lambda = 0.03$ (Table 10), while additional ablations on the remaining architectural and training choices (Section F.3) determine the final TRACE configuration used in the downstream experiments.

Comparison with pretrained representations. We next assess whether TRACE encodes more recoverable physical information than representations learned by representative forecasting-based foundation models. We compare TRACE with two recent foundation models for physical dynamics:

the AViT encoder from MPP (McCabe et al., 2024) and the encoder of PDE-Transformer (Holzschuh et al., 2025).

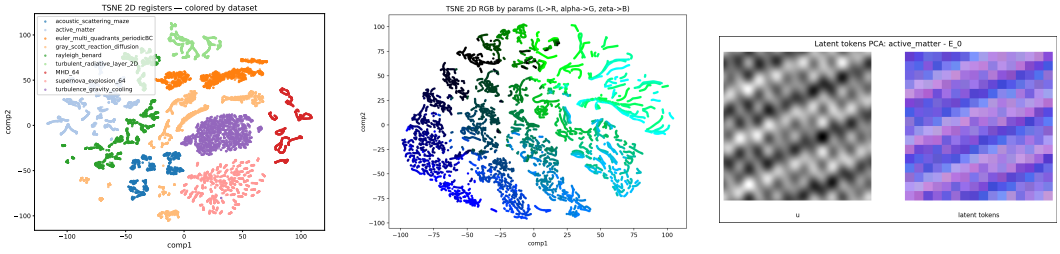
Table 2: **Parameter regression with frozen pretrained representations.** RMSE on normalized targets (lower is better; bold: best per column). TRACE is compared with MPP and PDE-Transformer using the same frozen parameter-regression probe. Predicting the training mean gives an RMSE of 1. Probe and encoder details are given in Section F.4.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Model	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
MPP	8.85E-2	3.56E-2	1.23E-1	1.94E-1	1.02E+0	3.38E-1	5.84E-1	2.41E-2	1.54E-1
PDE-Transformer	9.79E-2	2.62E-2	**8.38E-2**	1.75E-1	9.83E-1	2.73E-1	5,28E-1	3.93E-3	7.83E-2
TRACE (ours)	**7.00E-2**	**8.25E-3**	1.02E-1	**9.69E-2**	**9.43E-1**	**1.98E-1**	**4.71E-1**	**3.70E-3**	**4.68E-2**

TRACE achieves the lowest error on eight of the nine parameters (Table 2). PDE-Transformer generally outperforms MPP and achieves the lowest error on the Active Matter α parameter, but TRACE retains substantial margins on the remaining variables. The difference is particularly pronounced for the Rayleigh number, where TRACE achieves an error roughly $3\times$ lower than PDE-Transformer and $4\times$ lower than MPP. All three representations struggle to recover T_0. Because the models differ substantially in architecture and capacity (212M parameters for MPP, 211M for PDE-Transformer, and 88M for TRACE), this comparison should be interpreted as an evaluation of the information accessible from their frozen representations rather than as a controlled architectural comparison. Overall, the results indicate that TRACE captures more readily recoverable physical information than the forecasting-based foundation models considered here, despite using less than half as many parameters.

3.3 Structure of the frozen representation

To complement the quantitative parameter-recovery analysis, we qualitatively examine the structure of the frozen TRACE representation using two complementary visualizations (Figure 3). First, we apply t-SNE to the global tokens to analyze how trajectory-level representations are organized across physical systems and parameter regimes. Second, we apply PCA to the local encoded tokens of a single trajectory and map the first three principal components to RGB, allowing us to visualize how the latent representation is organized over the spatial domain.



(a) Multiphysics. (b) Active matter. (c) PCA of encoded frame tokens.

Figure 3: Visualization of the learned latents. 3a: 2D t-SNE of the global tokens across datasets, with each point corresponding to a trajectory. 3b: 2D t-SNE of Active Matter trajectories, colored by the underlying physical parameters (L, α, ζ) mapped to RGB (see Section H.1 for detailed parameter mapping). 3c: PCA of encoded frame tokens (first three components mapped to RGB).

The global representations exhibit a clear organization with respect to the physical regime, both across heterogeneous physical systems (Figure 3a) and within a single parametric system (Figure 3b). The visualization suggests that the global tokens encode both system identity and variations in the governing parameters. Additional t-SNE projections for other physical systems, together with qualitative autoregressive rollout visualizations for the forecasting models, are provided in Section H.

At the scale of a single trajectory, the PCA visualization shows that the local latent representation preserves spatial structure from the input (Figure 3c). It also suggests that tokens associated with

different physical fields develop distinct latent structures, some of which can be related qualitatively to physical features of the corresponding fields. Additional visualizations and discussion are provided in Section H.2.

3.4 DYNAMICS FORECASTING

We next evaluate whether the TRACE representation can provide the dynamical context that a forecasting model would otherwise infer from a sequence of past states. We use two backbones, PDE-Transformer and MPP, and train two versions of each. The baseline model is a standard forecaster that directly receives frames 1–10, whereas the TRACE-conditioned model receives only frame 10 together with the frozen TRACE representation extracted from frames 1–10. Apart from this conditioning mechanism, both versions share the same architecture. Both models are evaluated on the same forecasting task, predicting frames 11–30 by autoregressive rollout. We consider models trained from scratch on eight datasets, as well as models fine-tuned on both in-distribution and unseen systems. Section F details the conditioning mechanisms, additional variants, dataset assignments, and training protocols.

Table 3: **Dynamics forecasting results on in-ditribution datasets.** First-frame per-field NRMSE on the test split (lower is better). **base**: backbone receiving ten input frames. **TRACE**: same backbone receiving one input frame and the TRACE context vector. (a) Models trained from scratch on the eight training datasets. (b) Models fine-tuned for 10k steps on each dataset from the pretrained model. Bold marks report best performance.

	(a) Trained from scratch				**(b) Fine-tuned on one dataset**			
	PDE-Transformer		MPP		PDE-Transformer		MPP	
Dataset	base	TRACE	base	TRACE	base	TRACE	base	TRACE
Acoustic Scatt. Maze	0.200	**0.147**	**0.062**	0.097	–	–	–	–
Active Matter	0.119	**0.108**	**0.140**	0.142	0.119	**0.103**	**0.128**	0.132
Euler Multi-Quad.	0.245	**0.191**	0.159	**0.149**	–	–	–	–
Rayleigh-Bénard	0.224	**0.183**	0.168	**0.149**	0.226	**0.174**	0.151	**0.140**
Turb. Rad. Layer 2D	0.133	**0.118**	**0.280**	0.287	–	–	–	–
MHD 64	0.333	**0.273**	**0.441**	0.458	–	–	–	–
Supernova Expl. 64	0.745	**0.588**	**0.642**	0.720	0.756	**0.577**	**0.642**	0.717
Turb. Grav. Cool.	0.099	**0.061**	0.173	**0.168**	0.102	**0.055**	0.160	**0.155**
Rollout time (ms)	**341**	387	1249	**483**	**341**	387	1249	**483**

In-distribution results The two backbones handle temporal context in opposite ways, and TRACE improves each one in a different direction (Table 3). MPP keeps the full ten-frame history as context: its AViT backbone processes all ten frames again at every rollout step, which is accurate but expensive. With TRACE, MPP receives a single frame and a compact context vector instead, and reaches similar accuracy while being 2.6× faster (483 vs. 1249 ms per rollout). The base model is slightly better at the first frame on five of the eight datasets, and after fine-tuning each model is better on two of the four (full fine-tuning results in Table 19). PDE-Transformer takes the opposite approach: it compresses the ten frames directly at the input, which keeps it fast but loses information. At a similar cost (387 vs. 341 ms), TRACE lowers its first-frame NRMSE on all eight training datasets, by 9–38%, and on all four fine-tuned datasets, by 13–46%. Over longer horizons, this gain holds on six of the eight datasets for rollout steps 1–10, but reverses on five datasets for steps 11–20 (per-horizon results in Tables 16 and 17). TRACE therefore balances the two strategies: it keeps the accuracy of a model that reads the full history, at a cost close to that of a model that compresses it.

Out-of-distribution results on unseen systems. TRACE is the only representation that enables direct evaluation on unseen systems, since base models rely on static, dataset-specific lookup tables for their field embeddings (Section C.2). In contrast, TRACE dynamically generates per-context field representations on the fly from the input frames, enabling zero-shot generalization without parameter updates. After fine-tuning, TRACE gives 31% and 52% lower error than the base on Rayleigh-Taylor and Shear Flow with PDE-Transformer, respectively, while the base is slightly better on both with MPP. Overall, the TRACE representation compresses temporal context effectively, offers a competitive trade-off between accuracy and computational cost, and is the only option that allows zero-shot evaluation on new systems.

Table 4: **Dynamics forecasting on unseens systems.** First-frame per-field NRMSE on the test split (lower is better).**Zero-shot**: models trained on the eight training datasets and evaluated directly on an unseen system. Only TRACE is reported, since base models rely on dataset-specific field embeddings.**Fine-tuned**: models fine-tuned for 10k steps on each unseen system. Bold marks report best score.

	Zero-shot		**Fine-tuned**			
	PDE-Transformer	MPP	PDE-Transformer		MPP	
Dataset	TRACE	TRACE	base	TRACE	base	TRACE
Rayleigh-Taylor	0.667	0.758	0.650	**0.447**	**0.425**	0.429
Shear Flow	0.514	0.454	0.372	**0.180**	**0.068**	0.079

3.5 Representation utilization controls

The forecasting comparison shows that the representation is useful; we next test whether the backbones actually use its trajectory-specific content. We replace the TRACE vector at evaluation time with either the vector from a different trajectory of the same regime (*Context*) or a vector from another random trajectory on the batch (*Random*), without retraining (Table 5).

Table 5: **Replacing the TRACE vector at evaluation** (per-field NRMSE on the validation split; lower is better; geometric mean over the eight training datasets, with changes from Own in parentheses). Own denotes the unmodified vector. Context uses a vector from a different trajectory of the same dataset, while Random uses a vector from another random trajectory on the batch. Per-dataset results, the dataset-mean replacement, and the D_h variant are reported in Table 20.

Model	Own	Context	Random
PDE-Transformer, TRACE	0.258	0.363 (+52%)	0.323 (+33%)
MPP, TRACE	0.272	0.365 (+52%)	0.412 (+74%)

The backbones use the vector's content. Replacements increase the first-frame error on both backbones by 33% to 74%, showing that the backbones actively exploit the conditioning vector rather than merely benefiting from an additional input. Supernova is the main exception, with neither replacement changing the error by more than 2.2% (Table 20). Additional controls, including fine-tuning with Context vectors, are reported in Section G.4.

4 Conclusion

We introduced TRACE, a self-supervised multiphysics encoder that learns reusable representations of spatio-temporal physical systems independently of downstream tasks. Its key component, *regime fingerprinting*, complements local masked latent prediction with a global objective that encourages the representation to capture persistent characteristics of the underlying dynamics from partial observations. Our experiments show that TRACE representations improve physical-parameter recovery over forecasting-based foundation models and provide compact conditioning for downstream forecasting, with favorable accuracy–cost trade-offs and transfer to unseen systems. These results suggest that separating regime identification from downstream prediction is a promising direction for building more reusable physical foundation models.

Limitations. Regime fingerprinting assumes that persistent system characteristics can be inferred from a short initial observation. This assumption may fail when different regimes produce nearly indistinguishable early dynamics. Our study is also limited to the physical systems, architectures and downstream tasks considered here. In particular, we do not study generative models, which have shown strong potential for PDEs, both for forecasting dynamics and for multi-task generation (Lippe et al., 2023; Nguyen et al., 2025; Koupaï et al., 2025). In vision and video, self-supervised encoders have proven effective as representations for generative models (Zheng et al., 2026; Singh et al., 2026). Whether this carries over to physical systems remains open, and we leave it for future work.

AI Usage Statement. Generative AI tools were used to assist with software implementation and parts of the experimental workflow, as well as with English proofreading and language refinement. AI-assisted code and experimental outputs were reviewed and validated by the authors. We take responsibility for the final content of this work, including all claims, analyses, code, and experimental results.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *Advances in Neural Information Processing Systems*, volume 37, pp. 25152–25194. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2cd36d327f33d47b372d4711edd08de0-Abstract-Conference.html.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and Scalable Self-Supervised Learning Without the Heuristics, November 2025. URL http://arxiv.org/abs/2511.08544.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message Passing Neural PDE Solvers, March 2023. URL http://arxiv.org/abs/2202.03376.

Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. VICON: Vision in-context operator networks for multi-physics fluid dynamics prediction. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=6V3YmHULQ3.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624. Curran Associates, Inc., 2024. doi: 10.52202/079017-2311.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers, 2019. URL https://arxiv.org/abs/1912.12180.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-transformer: Efficient and versatile transformers for physics simulations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23562–23602. PMLR, 2025. URL https://proceedings.mlr.press/v267/holzschuh25a.html.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4140. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb3878c1dcbfff9ee95d5d033e5f5942-Abstract-Conference.html.

Armand Kassaï Koupaï, Jorge Mifsut Benet, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. GEPS: Boosting generalization in parametric PDE neural solvers through adaptive conditioning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2258. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/82844e428d9163a9f94830dc03af4f9c-Abstract-Conference.html.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Armand Kassaï Koupaï, Lise Le Boudec, Louis Serrano, and Patrick Gallinari. ENMA: Tokenwise autoregression for continuous neural PDE operators. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=3CYXSMFv55.

Armand Kassaï Koupaï, Lise Le Boudec, and Patrick Gallinari. Efficient generative transformer operators for million-point pdes, 2025. URL https://arxiv.org/abs/2512.04974.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/df438e5206f31600e6ae4af72f2725f1-Abstract.html.

Edward Li, Zichen Wang, Jiahe Huang, and Jeong Joon Park. VideoPDE: Unified Generative PDE Solving via Video Inpainting Diffusion Models, June 2025. URL http://arxiv.org/abs/2506.13754.

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

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers, August 2023. URL https://arxiv.org/abs/2308.05732v2.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119301–119335. Curran Associates, Inc., 2024. doi: 10.52202/079017-3791.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Régaldo-Saint Blancard, François Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, Francois Lanusse, Rudy Morel, Ruben Ohana, Liam Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. Walrus: A cross-domain foundation model for continuum dynamics. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026. URL https://arxiv.org/abs/2511.15684.

Rudy Morel, Jiequn Han, and Edouard Oyallon. DISCO: Learning to DISCover an evolution Operator for multi-physics-agnostic prediction, April 2025. URL http://arxiv.org/abs/2504.19496.

Tung Nguyen, Arsh Koneru, Shufan Li, and Aditya Grover. PhysiX: A Foundation Model for Physics Simulations, July 2025. URL http://arxiv.org/abs/2506.17774.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: A large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037. Curran Associates, Inc., 2024. doi: 10.52202/079017-1430. Datasets and Benchmarks Track.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, February 2024. URL http://arxiv.org/abs/2304.07193.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023.

Helen Qu, Rudy Morel, Michael McCabe, Francois Lanusse, Alberto Bietti, Shirley Ho, and Yann LeCun. Representation learning for spatiotemporal physical systems. In *AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations*, 2026. URL https://openreview.net/forum?id=rZQoZqUiNK.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, November 2017. URL https://arxiv.org/abs/1711.10561v1.

M. S. Rautela, A. Most, S. Mansingh, B. C. Love, A. Scheinker, D. Oyen, N. Debardeleben, E. Lawrence, and A. Biswas. MORPH: PDE foundation models with arbitrary data modality, 2025. URL https://arxiv.org/abs/2509.21670.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pp. 4470–4479. PMLR, 2018.

Louis Serrano, Armand Kassaï Koupaï, Thomas X. Wang, Pierre Erbacher, and Patrick Gallinari. Zebra: In-context generative pretraining for solving parametric pdes. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71242–71262. Curran Associates, Inc., 2023. doi: 10.52202/075280-3119.

Jayden Teoh, Manan Tomar, Kwangjun Ahn, Edward S. Hu, Tim Pearce, Pratyusha Sharma, Akshay Krishnamurthy, Riashat Islam, Alex Lamb, and John Langford. Next-latent prediction transformers learn compact world models, 2025. URL https://arxiv.org/abs/2511.05963.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018a. URL https://arxiv.org/abs/1807.03748.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018b. URL https://arxiv.org/abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, August 2023. URL http://arxiv.org/abs/1706.03762.

Sifan Wang, Ananyae Kumar Bhartari, Bowen Li, and Paris Perdikaris. Gradient alignment in physics-informed neural networks: A second-order optimization perspective. In *Advances in Neural Information Processing Systems*, volume 38, pp. 186682–186732, 2025. doi: 10.52202/085713-5615. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f655706547885b8e32ef46f1c067ece2-Abstract-Conference.html.

Shizheng Wen, Mingyuan Chi, Tianwei Yu, Ben Moseley, Mike Yan Michelis, Pu Ren, Hao Sun, and Siddhartha Mishra. Learning, solving and optimizing PDEs with TensorGalerkin: an efficient high-performance Galerkin assembly algorithm. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235. PMLR, 2024.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-Context Operator Learning with Data Prompts for Differential Equation Problems. *Proceedings of the National Academy of Sciences*, 120 (39):e2310142120, September 2023. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2310142120. URL http://arxiv.org/abs/2304.07993.

Yuan Yin, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Baskiotis, and Patrick Gallinari. LEADS: Learning dynamical systems that generalize across environments. In *Advances in Neural Information Processing Systems*, volume 34, pp. 7561–7573, 2021a.

Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012, 2021b. doi: 10.1088/1742-5468/ac3ae5.

Xiangyu Zhao, Zhiwang Zhou, zhangwenlong, Yihao Liu, Xiangyu Chen, Junchao Gong, Hao Chen, Ben Fei, Shiqi Chen, Wanli Ouyang, Xiao-Ming Wu, and LEI BAI. WeatherGFM: Learning a weather generalist foundation model via in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=izjNI5bcOV.

Yanshun Zhao, Xiaoyu Peng, Jiamin Jiang, Congcong Zhu, and Jingrun Chen. MeshTok: Efficient multi-scale tokenization for scalable PDE transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0u1LigJaab.

Anthony Zhou and Amir Barati Farimani. Masked autoencoders are PDE learners. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=rZNuiFwXVs.

Anthony Zhou and Amir Barati Farimani. Masked autoencoders are PDE learners. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=rZNuiFwXVs.

CONTENTS OF THE APPENDICES

Appendix A:	Related Work	14
Appendix B:	In-Context Weight Conditioning and Channel Embeddings	20
Appendix C:	Training Details and Hyperparameters	22
Appendix D:	Parameter Regression: Full Study and Ablations	24
Appendix E:	Dynamics Forecasting: Full Results and Controls	28
Appendix F:	Extended Visualizations	33

A RELATED WORK

Neural PDE solvers. Neural approaches to spatiotemporal dynamics span a spectrum of inductive biases, determined by how explicitly they encode physical knowledge (Yin et al., 2021b). Physics-Informed Neural Networks (PINNs) incorporate the governing equations directly in the loss (Raissi et al., 2017), which can be effective in data-scarce regimes but typically assumes access to an accurate PDE specification and often faces optimization difficulties, especially for stiff systems (Wang et al., 2025; Krishnapriyan et al., 2021). In parallel, purely data-driven surrogates learn evolution directly from observations (Li et al., 2021; Alkin et al., 2024; Wu et al., 2024), a setting that is natural when the underlying physics is unknown, observations are partial, or measurements are noisy (Huang et al., 2024; Yin et al., 2021a; Koupaï et al., 2025). This shift has driven substantial architectural progress, most notably neural operators and scalable Transformer variants, aimed at improving efficiency (Wen et al., 2026; Zhao et al., 2026).

Learning the latent physical context. The price of being data-driven is that the solver must infer the unknown *physical context* that generates the observations, e.g., coefficients, forcing terms, boundary conditions, or other regime descriptors, which is particularly challenging under distribution shift. Many methods therefore entangle two objectives: *dynamics estimation* (inferring context from limited observations) and *dynamics forecasting* (predicting future states under that context). Existing strategies estimate context implicitly by conditioning on observed state sequences (McCabe et al., 2024), or introduce explicit adaptation variables such as conditioning codes or learned latent parameters (Yin et al., 2021a; Kassaï Koupaï et al., 2024); related directions explore in-context adaptation mechanisms that infer the regime from a short prompt trajectory (Serrano et al., 2025; Yang et al., 2023).

Foundation models for spatiotemporal physics. Building on scalable architectures, recent work has pursued PDE foundation models that learn transferable representations by pretraining on large, heterogeneous collections of physical systems. Early efforts primarily scaled *next-step (next-frame) prediction* across diverse PDE families to enable transfer and fine-tuning (Subramanian et al., 2023; McCabe et al., 2024; Herde et al., 2024). More recent approaches have expanded the pretraining signal beyond one-step forecasting, including denoising (Hao et al., 2024), generative objectives (Holzschuh et al., 2025; Li et al., 2025), and masking-based training (Zhou & Barati Farimani, 2024), as well as meta-learning and adaptation-oriented objectives (Morel et al., 2025; Cao et al., 2026). Despite this progress, much of the literature still couples context inference with forecasting through state-space prediction losses. Masked reconstruction methods (e.g., MAE-style objectives) learn representations by filling in missing spatiotemporal regions (Zhou & Barati Farimani, 2024), but predicting raw states can over-emphasize high-frequency, instance-specific details, and existing implementations often struggle with heterogeneous data (e.g., varying geometries and field counts). In contrast, we propose TRACE: a unified architecture designed for heterogeneity, trained with a latent-space predictive objective (JEPA) augmented by regime fingerprinting. This decouples context learning from signal reconstruction and yields reusable, physics-grounded context embeddings for downstream tasks such as forecasting and zero-shot adaptation.

B DATASET DETAILS

All experiments use The Well (Ohana et al., 2024), which contains a diverse collection of 2D and 3D physical simulation datasets. We pretrain on nine systems spanning two and three spatial dimensions, heterogeneous field structures, and different resolutions. Two additional systems, outside the pretraining mixture, are used only for held-out forecasting and fine-tuning. Table 6 summarizes the role of each dataset. We refer to Ohana et al. (2024) for full descriptions of the underlying systems and report here only the preprocessing choices needed to reproduce our experiments.

Table 6: **Datasets and their roles.** Pretraining denotes the mixture used for our encoder and the pretrained MPP baseline (Section D). Parameters denotes datasets used for physical-parameter regression (Sections 3.2 and F). the forecasting backbones. Fine-tuning denotes single-dataset fine-tuning of the forecasting models (Section G.3).

Dataset	Dim.	Pretraining	Parameters	Forecasting	Fine-tuning
Acoustic Scattering Maze	2D	✓		train	
Active Matter	2D	✓	✓	train	✓
Euler Multi-Quadrants	2D	✓		train	
Gray-Scott	2D	✓		held out	
Rayleigh-Bénard	2D	✓	✓	train	✓
Turbulent Radiative Layer 2D	2D	✓		train	
MHD 64	3D	✓	✓	train	
Supernova Explosion 64	3D	✓		train	✓
Turbulence Gravity Cooling	3D	✓	✓	train	✓
Shear Flow	2D			held out	✓
Rayleigh-Taylor Instability	3D			held out	✓

B.1 DATASET PREPROCESSING

The time ranges, spatial and temporal subsampling factors, and window strides below apply to pretraining and parameter-regression probes, which use windows of ten frames. Forecasting experiments use the same time ranges and subsampling factors but divide each trajectory into non-overlapping 30-frame chunks instead of using these window strides (Section G.1).

Acoustic Scattering Maze. We use the *Acoustic Scattering Maze* dataset (`acoustic_scattering_maze`), which describes acoustic wave propagation through media with spatially varying density. We use the full spatial resolution and restrict the time range to timesteps $[20, 90]$, with a window stride of 15 frames.

Active Matter. The `active_matter` dataset contains simulations of rod-like active particles immersed in a Stokes flow. The particles convert chemical energy into mechanical work, generating stresses that couple through the fluid and produce collective dynamics. We use the complete dataset at full spatio-temporal resolution, with a window stride of 1 frame.

Euler Multi-Quadrants. Euler Multi-Quadrants (`euler_multi_quadrants_periodicBC`) consists of 2D inviscid compressible Euler simulations with quadrant-wise piecewise-constant initial conditions and periodic boundary conditions. We use the full dataset and downsample both space and time by a factor of 2, with a window stride of 15 frames.

Gray-Scott. The Gray-Scott system (`gray_scott_reaction_diffusion`) consists of two coupled reaction-diffusion equations describing the concentrations of two chemical species. We use timesteps in the range $[0, 300]$ and subsample the temporal axis by a factor of 10, with a window stride of 1 frame. Gray-Scott is also held out from the training of the forecasting backbones; since it belongs to the pretraining mixture, we draw no generalization conclusion from this system.

Rayleigh-Bénard. The `rayleigh_benard` dataset contains 2D horizontally periodic Rayleigh-Bénard convection simulations. We use the full dataset without spatial or temporal subsampling, with a window stride of 10 frames.

MHD 64. The `MHD_64` dataset contains 3D magnetohydrodynamic turbulence simulations. We use the complete 64^3 dataset with a window stride of 1 frame.

Supernova Explosion. The `supernova_explosion_64` dataset models a supernova blastwave propagating through a dense, cold, turbulent interstellar medium. We use the complete 64^3 dataset without temporal subsampling, with a window stride of 2 frames.

Turbulence Gravity Cooling. The `turbulence_gravity_cooling` dataset comprises 3D simulations of a turbulent interstellar gas cloud with hydrodynamics, self-gravity, and radiative heating and cooling. We use the full dataset without additional preprocessing, with a window stride of 5 frames.

Turbulent Radiative Layer 2D. The `turbulent_radiative_layer_2D` dataset contains 2D compressible hydrodynamic simulations of a radiatively cooling turbulent mixing layer. We use the complete dataset at full resolution, with a window stride of 1 frame.

B.2 Held-out evaluation datasets

The two datasets below lie outside both the pretraining mixture and the training data of the forecasting backbones. They are used for held-out zero-shot forecasting and as the two unseen systems in the fine-tuning experiment (Section G.3). Neither is used for pretraining or parameter regression.

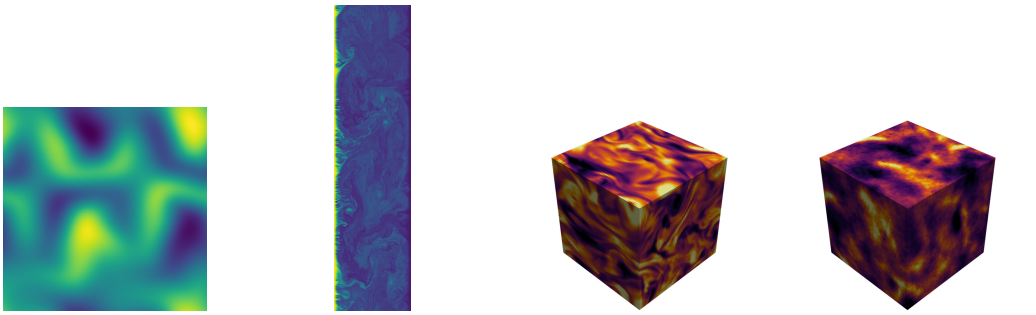
Shear Flow. The `shear_flow` dataset consists of 2D periodic incompressible shear-flow simulations coupled with a passive tracer. We downsample both space and time by a factor of 2.

Rayleigh-Taylor Instability. The `rayleigh_taylor_instability` dataset contains 3D simulations of the instability that develops at the interface between two fluids of different densities under gravity. We use the full time range without temporal subsampling and downsample space by a factor of 2.

B.3 Multi-physics data loading

The nine pretraining datasets have different field counts and spatial resolutions, so each mini-batch is restricted to a single physical system. At each step, the loader samples a system with probability proportional to its remaining batch count, so that each dataset is visited in proportion to its size over an epoch. Every rank draws the same system at a given step, while samples within each dataset are sharded across ranks. This avoids padding across systems and keeps every global batch internally consistent, while allowing the physical system to change between batches.

Dataset-level normalization is disabled. Inputs are normalized per sample and per field using reversible instance normalization (Kim et al., 2022). The forecasting backbones use the same single-dataset batch construction on their eight training datasets.



(a) Active Matter (b) Rayleigh-Bénard (c) MHD 64 (d) Turbulence Gravity Cooling

Figure 4: Samples from four systems in the pretraining mixture. Spatial resolutions and dimensions differ across systems. We show a single field per system; field counts also vary across datasets.

C ARCHITECTURE DETAILS

In our implementation, the downstream conditioned forecasting pipeline is instantiated with three distinct components:

(*i.*) the pretrained TRACE context encoder, shared across tasks;

(*ii.*) a task-specific lightweight extractor;

(*iii.*) a task-specific autoregressive forecasting backbone (PDE-Transformer or MPP).

The architectural details of the forecasting backbones and how the conditioning is injected are provided in Section G, and a full summary of model capacities is provided in Section D. Below, we detail the internal components of the TRACE encoder and the extractor.

C.1 ENCODER DETAILS

The context encoder maps partial spatiotemporal observations to a latent representation and is trained using our augmented JEPA-style objective. The backbone is a Transformer with 12 layers, an embedding dimension of 384, and 12 attention heads. Each block follows a pre-norm design with RMSNorm, and the MLP uses an expansion ratio of 4 and SwiGLU activations. All dropout rates are set to 0.

Heterogeneous Attention. The observation masks used during pretraining can result in irregular sets of visible tokens, with different spatial locations and physical fields containing different numbers of observed tokens. To efficiently process these arbitrary masks without padding to a common shape, visible tokens are flattened into a single sequence and processed using block-diagonal attention masks. The block structure preserves the axial attention pattern described in Section 2.3: for spatiotemporal attention, tokens belonging to the same field form an attention block, while for field attention, tokens corresponding to the same spatiotemporal location form a block. This allows ragged views to be packed into a single sequence while preventing interactions between unrelated groups.

Global Tokens. The encoder maintains 8 learnable global tokens that aggregate system-level information. These act as the carrier for the regime-fingerprinting auxiliary loss (Section 2.1). The encoder ultimately outputs the sequence of encoded patch-level tokens together with the updated global tokens.

Patchification. Patchification is implemented via a *repeat patchifier*, which enables a unified treatment of 2D and 3D systems. Abstractly, concurrent cross-domain models (such as Walrus and MORPH; McCabe et al., 2026; Rautela et al., 2025) unify data by treating 3D inputs as (B, C, D, H, W) and expanding 2D inputs to $(B, C, 1, H, W)$. However, extracting spatial patches of size p from these grids naturally yields different raw feature sizes: p^2 values for 2D inputs and p^3 values for 3D inputs.

While Walrus handles this by maintaining separate, dimensionality-specific projection weights, TRACE duplicates the 2D unitary dimension p times, expanding 2D patches to exactly p^3 values (Figure 6). Consequently, TRACE uses the *exact same projection weights* for both 2D and 3D data. Our rationale is to achieve complete weight-sharing across all network components, avoiding dimensionality-specific layers entirely. We empirically validate that this unified projection matches or exceeds the performance of dimensionality-specific projections in Section F.5.

Tokens are extracted per field at a fixed spatial patch shape, allowing several grid resolutions to be admitted without resampling. Patchification is applied independently for each physical field, producing separate patch sequences per variable (Figure 5). Each patch is then flattened and projected to the embedding dimension through the shared, learnable two-layer MLP with SwiGLU activation.

Positional Encoding. We employ axial positional encodings over the temporal, spatial, and field axes. Encodings are normalized and scaled by a factor of 50, with a temperature of 1024, and are summed across axes before being added to the projected patch embeddings.

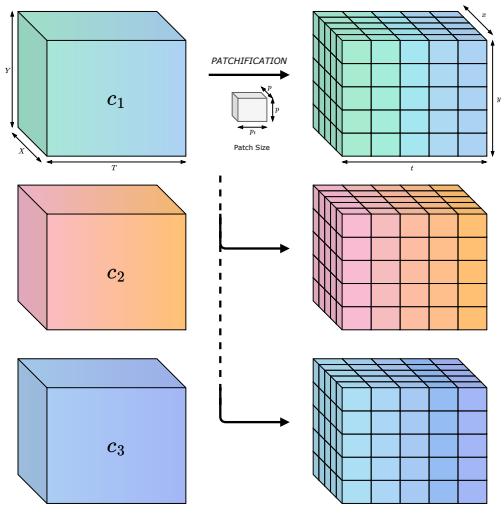


Figure 5: Patchification is applied independently for each physical field, yielding separate patch sequences per variable.

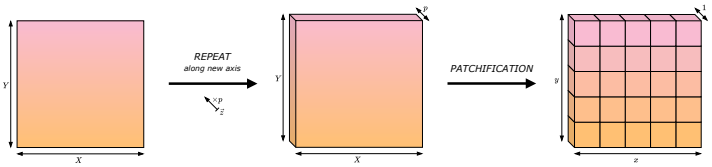


Figure 6: For 3D systems, patches of shape $(2, 16, 16, 16)$ are extracted directly, while 2D systems are lifted by repeating the missing spatial axis to match the patch depth.

Masking Strategy. During pretraining, we apply structured spatiotemporal-field masking using a multi-block mask generator. The target masking ratio is set to 0.85 (Section F.3). A binary mask over the token grid is formed by the union of axis-aligned blocks, each spanning an interval in time and in each spatial axis, and carrying its own field subset. The mask need not be the same in every field. Each block is constructed by sampling contiguous regions whose relative scale is drawn from a distribution with mean $\mu_s = 0.40$ and variance $\sigma_s = 0.05$, with an aspect-ratio jitter of 0.10.

Figure 7 shows a real mask sampled by our generator, visualized at a lower masking ratio of 20% for clarity. Masked tokens exhibit structured continuity across all axes: when moving along the temporal dimension T, similar spatial regions remain masked over several steps, often with small displacements; likewise, when moving across fields V, analogous spatial blocks are masked in multiple variables. This block-wise masking enforces partial observability at the level of coherent spatiotemporal regions, encouraging the encoder to rely on long-range context and cross-field information to infer the missing representations.

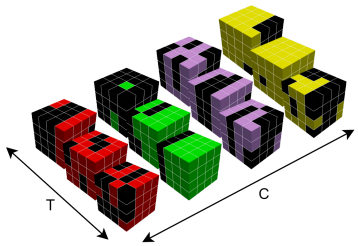


Figure 7: Example of structured multi-block masking on a spatiotemporal sequence. Each cube represents a token indexed by space, time T, and field V. Colored small cubes denote observed tokens, while black small cubes indicate masked tokens. The visualization uses a 20% masking ratio for readability. Masked regions appear as contiguous spatial blocks that persist across time and fields with small shifts.

C.2 In-context weight conditioning and field embeddings

To understand why **TRACE** successfully generalizes zero-shot to out-of-distribution (OOD) systems while standard baselines fail, we must look at how physical variables are mapped into the network.

Standard forecasting models (such as the base MPP or PDE-Transformer) rely on a static, learned lookup table of field embeddings. Each physical variable (e.g., density, velocity components) is tied to a fixed vector learned during training. When presented with an unseen physical system containing novel fields or dimensions, these models break because the new variables have no entry in their static lookup table.

In contrast, the **TRACE** conditioned models perform in-context adaptation via two dynamic mechanisms:

1. **Dynamic Field Embeddings:** For the standard **TRACE** variant, the pretrained extractor reads the context frames and directly generates the per-field embeddings on the fly. When encountering an unseen physical system, the frozen encoder synthesizes appropriate field representations dynamically rather than retrieving them from a fixed table.
2. **Activation Modulation (AdaLN-Zero):** The extracted regime embedding is projected into scale and shift parameters that modulate every block of the backbone through zero-initialized adaptive layer normalization (AdaLN-Zero; Peebles & Xie, 2023).

This combination allows the architecture to adapt its internal representations in-context to novel physical regimes without requiring any updates to its static parameters or lookup tables.

D MODEL CAPACITIES

For clarity and ease of comparison across the pretraining and downstream tasks, Table 7 summarizes the trainable parameter counts for all primary encoders, forecasting backbones, and extractors used throughout the experiments.

Detailed training protocols for these architectures are provided in Section E. Specific baseline comparisons for the parameter regression task are detailed in Section F, and the conditioned forecasting protocols are detailed in Section G.

Table 7: **Model sizes used in the experiments.** Parameter counts represent the trainable weights for each corresponding architecture.

Model	Parameters
TRACE Encoder (Base)	88.3M
MPP Encoder (AViT baseline)	212M
PDE-Transformer (forecasting backbone)	210M
MPP (AViT forecasting backbone)	221M
Parameter-regression probe	11.3M–42M
Forecasting conditioning extractor	13.5M–14M

E TRAINING DETAILS

All experiments are implemented in PyTorch (Paszke et al., 2019). We distinguish between the study configuration, used for the objective comparison and ablations, and the final configuration, used to produce the encoder that is evaluated in the downstream experiments.

E.1 PRETRAINING

All pretraining runs use the same encoder architecture, data mixture, and optimizer configuration, and differ only in the self-supervised objective or the experimental setting under study. We use two training configurations, summarized in Table 8. The study configuration is used for the objective comparison and ablations, while the final configuration is used for the encoder evaluated in the downstream experiments.

Table 8: **Pretraining configurations.** The study configuration is used for the objective study and ablations; the final configuration is used for the encoder evaluated downstream.

Hyperparameter	**Study**	**Final**
Optimization steps	30,000	100,000
Global batch size	16	32
Optimizer	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}
Schedule	cosine	cosine
Warmup steps	8,000	8,000
Weight decay	10^{-4}	10^{-4}
Precision	bfloat16	bfloat16
Gradient checkpointing	yes	yes
Masking ratio	0.85	0.85
Teacher EMA decay	0.99	0.99
Auxiliary weight λ	0.03	0.03
Auxiliary warmup	10,000	10,000
Auxiliary decay window	none	70,000–95,000
Gradient clipping	1.0	1.0
Learning-rate floor	10^{-7}	10^{-7}

The multi-physics loader constructs homogeneous batches because the datasets differ in spatial resolution and field structure. Each mini-batch therefore contains samples from a single physical system. At each step, the loader samples a system proportionally to its remaining number of batches in the current epoch. The same system is selected on every distributed rank, while samples within that system are sharded across ranks. This avoids padding between incompatible systems while preserving dataset-proportional sampling. Inputs are normalized per sample and per field using reversible instance normalization (Kim et al., 2022); no dataset-level normalization is applied. The forecasting backbones use the same single-dataset batch construction on their training datasets.

Masking and teacher. The student uses the multiblock masking scheme described in Section C, with a masking ratio of 0.85. The teacher is an exponential moving average of the student with decay 0.99. It processes the unmasked input and provides the latent targets at masked positions, while receiving no gradient.

Regime-fingerprinting auxiliary. The regime-fingerprinting auxiliary is weighted by $\lambda = 0.03$ and is linearly warmed up during the first 10,000 steps. In the final configuration, the weight is subsequently decayed to zero between steps 70,000 and 95,000. The corresponding ablations are reported in Section F.2.

E.2 BASELINE FORECASTING MODELS

To obtain baseline representations for the parameter regression task, we utilize the fully trained ten-frame (base) forecasting models of MPP (McCabe et al., 2024) and PDE-Transformer

(Holzschuh et al., 2025). Both models are trained from scratch on the nine-dataset mixture using their native forecasting objectives given a ten-frame context.

For MPP, we use a training schedule of 100,000 steps with a global batch size of 32 and the Adan optimizer (Xie et al., 2024) with weight decay 10^{-3}; the parameter-regression probe reads its checkpoint at step 96,000. To evaluate their captured physical information, we extract the encoder-side of both fully trained base models, freeze them, and apply the parameter-regression probe.

E.3 Frozen parameter-regression probe

For parameter regression, the encoder (whether TRACE, MPP, or PDE-Transformer) is frozen and an attentive pooling extractor and MLP head are trained separately for each dataset. The probe receives windows of ten frames and is trained for 10,000 steps with a learning rate of 5×10^{-4}. The exact same probe architecture and training procedure are used for every evaluated representation.

The physical parameters are transformed using fixed per-parameter statistics. We apply a $\log_{10}$ transform to Prandtl, Rayleigh, T_0, ρ_0, and Ms, followed by z-scoring. The remaining parameters are only z-scored. Default configurations are evaluated with three probe seeds; non-default ablation settings use a single probe seed. Thus, the variability reported for the default configurations reflects probe fitting on a fixed pretrained encoder rather than variation across pretraining runs.

E.4 Conditioned forecasting

The downstream forecasting backbones are trained from scratch while the pretrained TRACE encoder remains frozen. Only the conditioning extractor and the backbone are optimized. Each trajectory is divided into non-overlapping 30-frame chunks, with the first ten frames providing context and the remaining 20 frames serving as prediction targets.

The complete forecasting training schedules, validation procedure, checkpoint selection, and fine-tuning protocol are reported in Section G.

F PARAMETER REGRESSION: FULL STUDY AND ABLATIONS

This section complements Section 3.2. Section F.1 specifies the nineteen objectives and reports all of them (Table 9); Section F.2 isolates the weight of the regime-fingerprinting auxiliary (Table 10); Section F.3 reports the mechanism control, along with the global tokens and masking-ratio ablations; and Section F.4 details the comparison with MPP and simple baselines.

Except in Section F.4, all runs use the study configuration (Table 8) and the probe protocol of Section E.

For the single-setting ablations, we modify only one hyperparameter at a time from the default TRACE configuration (JEPA with regime fingerprinting at λ=0.03). In the ablation tables, the "default" row simply reproduces the results of this standard TRACE baseline for easy comparison.

Furthermore, the results for the default configurations are averaged over three different probe initializations (seeds) evaluated on a single pretraining run. Therefore, the reported variability reflects the variance of the probe fitting process itself, rather than variance across multiple pretraining runs. All other non-default ablation rows report results from a single probe seed.

F.1 THE STUDIED OBJECTIVES

The studied objectives fall into six families. Masked reconstruction (MAE; He et al., 2022; Zhou & Barati Farimani, 2024) decodes raw field values at masked positions with no teacher. Masked latent prediction (JEPA; Assran et al., 2023; Bardes et al., 2024), the backbone of Section 2, predicts an EMA teacher's latents at masked positions. Both families run under three masking geometries: independent random patches, field-consistent tubes (random in space and time, repeated across fields), and contiguous multiblocks. A coarse-view objective predicts the teacher's full-resolution latents from a spatially coarsened input. View-invariance objectives align embeddings of global and local views: JEPA global (crops with an EMA teacher and a contrastive loss) and LeJEPA (Balestriero & LeCun, 2025) with crops and with multiblock masking. Future-prediction objectives predict either the raw next-frame tendency $\mathbf{u}_T - \mathbf{u}_{T-1}$ or the teacher's latents of the future frame (causal temporal). The last family adds an auxiliary to JEPA (multiblock): regime fingerprinting (Section 2.1) at $\lambda \in \{0.01, 0.03, 0.1, 0.3, 1\}$, a pixel-space reconstruction auxiliary, and a latent-dynamics auxiliary inspired by NextLat (Teoh et al., 2025) that enforces a direct forward-prediction loss on the encoded tokens. No family received per-objective hyperparameter tuning beyond the variants shown; all rankings are read at the shared operating point of the study configuration.

Table 9 reports every objective, and Table 10 reports the different weights of the regime auxiliary. Every coarse-view, view-invariance, and future-prediction objective performs substantially worse on parameter recovery than the masked-prediction variants (with the exception of JEPA with random masking, which struggles heavily). The physical causal-temporal objective and JEPA global prove stronger than random-mask JEPA, further highlighting JEPA's heavy dependence on masking geometry.

F.2 WEIGHT OF THE REGIME-FINGERPRINTING AUXILIARY

Table 10 sweeps the weight λ of the InfoNCE auxiliary of Equation (6) and marks the boundary of its validity. The parameter recovery error is minimized across the majority of parameters at λ=0.03. At λ=0.01 and λ=0.1 the errors remain highly competitive, staying below plain JEPA (i.e., λ=0) on the low-variance parameters (Prandtl, Rayleigh, ζ, T_0, z, ρ_0). Larger weights degrade performance significantly, becoming worse than plain JEPA across the board at λ=0.3 and λ=1. We retain λ=0.03 as our final configuration. Within the $[0.01, 0.1]$ range, the low-variance parameters differ only modestly, all improving upon plain JEPA, indicating that the exact weight in this range is loosely determined rather than strictly tuned.

F.3 MECHANISM CONTROL, GLOBAL TOKENS, AND MASKING RATIO

Stable-target control. The study shows that the auxiliary helps, not why: the gain could come from the correspondence signal it carries, or simply from adding a second head and gradient path to the optimization. A stable-target control test removes trajectory correspondence while retaining

Table 9: **Full objective study** (frozen-probe parameter regression, NRMSE on normalized targets averaged over three probe seeds; lower is better; bold: best per column). Rows are grouped by family: masked reconstruction, masked latent prediction, coarse view, view invariance, future prediction, and auxiliaries on JEPA (multiblock). The regime auxiliary is shown at the retained λ=0.03 (TRACE); its other four weights are in Table 10.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Objective	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
MAE (field tube)	5.72E-2	1.89E-2	2.54E-2	1.374E-1	7.75E-1	1.49E-1	4.06E-1	3.81E-3	3.98E-2
MAE (random)	5.41E-2	1.84E-2	4.09E-2	1.059E-1	7.91E-1	1.68E-1	4.19E-1	3.61E-3	5.11E-2
MAE (multiblock)	6.14E-2	1.75E-2	3.27E-2	1.268E-1	8.06E-1	1.52E-1	4.26E-1	4.28E-3	6.37E-2
JEPA (field tube)	8.55E-2	1.71E-2	7.31E-2	1.078E-1	7.74E-1	1.44E-1	4.05E-1	3.97E-3	4.57E-2
JEPA (random)	3.332E-1	8.76E-2	9.15E-2	1.027E-1	8.31E-1	3.21E-1	4.65E-1	5.24E-3	2.677E-1
JEPA (multiblock)	5.64E-2	1.58E-2	2.31E-2	8.31E-2	8.03E-1	1.50E-1	3.93E-1	5.51E-3	4.39E-2
JEPA coarse view	1.595E-1	3.59E-2	1.603E-1	2.468E-1	1.018E+0	6.07E-1	6.92E-1	4.09E-3	3.588E-1
JEPA global	1.230E-1	2.25E-2	1.181E-1	2.112E-1	7.81E-1	2.44E-1	4.60E-1	**3.26E-3**	9.57E-2
LeJEPA global	3.122E-1	4.52E-2	1.333E-1	2.881E-1	8.26E-1	4.99E-1	5.99E-1	4.25E-3	4.155E-1
LeJEPA multiblock	3.705E-1	8.71E-2	1.739E-1	2.742E-1	7.84E-1	4.47E-1	5.15E-1	4.26E-3	1.440E-1
Causal temp. (physical)	1.529E-1	4.48E-2	1.672E-1	1.999E-1	8.32E-1	3.85E-1	5.46E-1	4.04E-3	1.787E-1
Causal temp. (latent, local)	3.668E-1	1.261E-1	2.462E-1	4.746E-1	8.60E-1	7.06E-1	6.65E-1	7.92E-3	5.980E-1
JEPA + reconstruction	6.01E-2	1.54E-2	3.55E-2	1.269E-1	**7.57E-1**	**1.42E-1**	3.94E-1	3.96E-3	5.03E-2
JEPA + latent dynamics	6.00E-2	2.20E-2	4.49E-2	**6.37E-2**	8.71E-1	1.69E-1	**3.88E-1**	3.43E-3	5.27E-2
TRACE (JEPA + RF)	**4.89E-2**	**1.46E-2**	**1.88E-2**	7.79E-2	8.77E-1	1.47E-1	4.16E-1	3.34E-3	**1.83E-2**

Table 10: **Weight λ of the regime-fingerprinting auxiliary** (frozen-probe parameter regression, NRMSE averaged over three probe seeds; lower is better; bold: best per column). All configurations use the plain JEPA (multiblock) baseline, corresponding to λ=0. The row at λ=0.03 is the configuration retained for TRACE.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Configuration	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
Plain JEPA (λ=0)	5.64E-2	1.58E-2	2.31E-2	8.31E-2	**8.03E-1**	1.50E-1	3.93E-1	5.51E-3	4.39E-2
+ RF (λ=0.01)	**4.38E-2**	1.81E-2	4.17E-2	**6.97E-2**	8.41E-1	**1.42E-1**	**3.78E-1**	3.44E-3	5.17E-2
+ RF (λ=0.03) (**TRACE**)	4.89E-2	1.46E-2	**1.88E-2**	7.79E-2	8.77E-1	1.47E-1	4.16E-1	3.34E-3	**1.83E-2**
+ RF (λ=0.1)	4.60E-2	1.56E-2	2.16E-2	8.77E-2	8.57E-1	1.48E-1	3.96E-1	4.21E-3	4.69E-2
+ RF (λ=0.3)	6.01E-2	**1.18E-2**	5.04E-2	1.080E-1	8.42E-1	1.65E-1	4.30E-1	3.70E-3	5.47E-2
+ RF (λ=1.0)	5.96E-2	1.21E-2	6.52E-2	1.394E-1	8.35E-1	1.98E-1	4.36E-1	**2.60E-3**	6.63E-2

the auxiliary head, weight, schedule, and gradient path: each teacher summary is replaced by a fixed random vector, drawn once per batch slot and held for the entire run, so the signal is withheld without creating a moving target. Table 11 reports the comparison. The control does not reproduce the gains of the full auxiliary, and the per-parameter picture is graded rather than uniform: it recovers part of the Prandtl improvement (5.14E-2, between plain JEPA's 5.64E-2 and the auxiliary's 4.89E-2), yet it is worse than plain JEPA on six of the nine parameters (for example, z: 1.62E-1 against 1.50E-1; ρ_0: 4.16E-1 against 3.93E-1). Since the control keeps the added machinery but carries no regime signal, the comparison is consistent with the correspondence signal contributing the bulk of the gain, with the machinery possibly contributing an optimization-side remainder.

Table 11: **Mechanism controls** (parameter regression NRMSE on normalized targets, lower is better; bold: best per column). The plain JEPA and auxiliary rows are 3-seed means; the control is a single run. The stable-target control keeps the head, weight, schedule, and gradient path of the auxiliary and removes the regime signal.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Auxiliary	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
Plain JEPA (multiblock)	5.64E-2	1.58E-2	2.31E-2	8.31E-2	8.03E-1	1.50E-1	**3.93E-1**	5.51E-3	4.39E-2
+ stable-target control	5.14E-2	1.64E-2	3.27E-2	9.28E-2	**7.99E-1**	1.62E-1	4.16E-1	3.98E-3	5.73E-2
+ regime (λ=0.03)	**4.89E-2**	**1.46E-2**	**1.88E-2**	**7.79E-2**	8.77E-1	**1.47E-1**	4.16E-1	**3.34E-3**	**1.83E-2**

Global Tokens. Table 12 varies the number of global tokens, the carrier of the auxiliary. Differences between 4, 8, and 16 global tokens are small on most parameters; the clearest movement is on the MHD Mach numbers, and since those columns carry large seed variance we keep that observation qualitative. The 0-global tokens row does not remove the auxiliary: its view pooling falls back to the mean of the patch tokens, so the same objective trains without a dedicated carrier. Global tokens count alone moves performance only modestly, while the control above links the improvement to the auxiliary signal; we therefore describe global tokens as the carrier the auxiliary trains, not as beneficial slots in themselves. The probes read the encoder's full token output rather than the global tokens alone, so this concerns where the auxiliary applies its pressure; the information itself need not live there exclusively. We retain 8 rather than paying for 16.

Table 12: **Ablation: number of global tokens** (parameter regression NRMSE on normalized targets, lower is better; bold: best per column). Non-default rows are single runs; the default (8) is a 3-seed mean.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Global tokens	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
0	4.67E-2	1.81E-2	2.33E-2	8.00E-2	9.03E-1	**1.38E-1**	4.24E-1	9.97E-3	4.64E-2
4	**4.21E-2**	1.87E-2	2.57E-2	1.05E-1	9.26E-1	1.56E-1	4.46E-1	4.29E-3	3.65E-2
8 (default)	4.89E-2	**1.46E-2**	**1.88E-2**	**7.79E-2**	**8.77E-1**	1.47E-1	4.16E-1	**3.34E-3**	**1.83E-2**
16	4.71E-2	1.78E-2	2.23E-2	9.69E-2	8.96E-1	1.49E-1	**4.06E-1**	4.03E-3	2.80E-2

Masking ratio. Table 13 varies the masking ratio. Performance degrades sharply below the default ratio of 0.85: at 0.5 and 0.75 the Prandtl error rises by factors of seven and three, and Active Matter's parameters degrade in the same direction. Raising the ratio to 0.95 is less harmful but still worse than the default on most parameters. Latent prediction requires a large masked fraction: leaving too much of the field visible may make the task solvable without encoding regime structure. We retain 0.85.

F.4 Comparison with MPP and simple baselines

When comparing TRACE against the baseline representations in the main text, we use one probe recipe for every backbone: an attentive pooler (8 queries, depth 2) followed by an MLP head, one per dataset, trained with AdamW at learning rate 5×10^{-4} (500 warmup steps, cosine decay) for 20,000 steps at batch size 16 on windows of ten frames, with one probe seed. Each entry in the parameter regression tables is the evaluation at the end of training, never a best-of-test pick.

Table 13: **Ablation: masking ratio** (parameter regression NRMSE on normalized targets, lower is better; bold: best per column). Non-default rows are single runs; the default (0.85) is a 3-seed mean.

	Rayleigh-Bénard		Active Matter		Turb. Grav. Cool.			MHD 64	
Masking ratio	**Prandtl**	**Rayleigh**	α	ζ	T_0	z	ρ_0	**Ma**	**Ms**
0.5	3.43E-1	7.97E-2	7.15E-2	1.34E-1	**7.67E-1**	2.12E-1	4.43E-1	**2.99E-3**	4.24E-2
0.75	1.44E-1	3.43E-2	6.39E-2	1.07E-1	8.90E-1	1.83E-1	4.66E-1	3.96E-3	7.76E-2
0.85 (default)	**4.89E-2**	**1.46E-2**	**1.88E-2**	**7.79E-2**	8.77E-1	**1.47E-1**	4.16E-1	3.34E-3	**1.83E-2**
0.95	5.54E-2	2.04E-2	5.40E-2	1.00E-1	8.02E-1	1.68E-1	**3.97E-1**	4.23E-3	4.36E-2

TRACE is the final encoder, read through its patch tokens and global tokens. MPP is the pretrained AViT (212M parameters, 12 processor blocks of width 1024), read after its last processor block, before the decoder; it has no global tokens. The pooler runs at the width of the backbone, so the MPP probe has about 42M trainable parameters against about 11M for TRACE. The random-weight reference re-initializes the TRACE architecture and freezes it. The training-mean reference predicts, for each parameter, the mean of the normalized targets over the training split, which scores an NRMSE of 1.00 on all nine parameters after rounding. The three Turbulence Gravity Cooling probes reached the time limit at step 18,501 (TRACE, random weights) and 19,001 (MPP) of 20,000 and are evaluated there; every other probe completed its 20,000 steps.

F.5 Patchification dimensionality ablation

To justify the use of the unified repeat patchifier introduced in Section C, we ablate it against a standard multidimensional baseline ("Multidim"), which maintains completely separate linear projections for 2D and 3D inputs. We evaluate both approaches using the MPP forecasting backbone.

As shown in Table 14, the repeat patchifier matches the performance of the separate multidim projections on 3D datasets (MHD 64, Turbulence Gravity Cooling) exactly, while actually yielding a slight improvement across all horizons on the 2D datasets (Active Matter, Euler Periodic). This confirms that a single, unified projection can effectively handle varying dimensionalities without sacrificing downstream accuracy.

Table 14: **Patchifier ablation on the MPP forecasting backbone** (NRMSE on the test split; lower is better; bold: better of the two). Multidim uses separate linear projections for 2D and 3D inputs. Repeat uses a single projection by lifting 2D inputs to 3D.

	Active Matter (2D)			Euler Periodic (2D)			MHD 64 (3D)			Turb. Gravity (3D)		
Encoder	1st	1:10	10:20	1st	1:10	10:20	1st	1:10	10:20	1st	1:10	10:20
Multidim	4.87E-2	2.81E-1	6.87E-1	7.33E-2	1.92E-1	3.29E-1	**3.53E-1**	**6.29E-1**	8.08E-1	**1.11E-1**	**3.82E-1**	**5.28E-1**
Repeat	**4.62E-2**	**2.79E-1**	**6.87E-1**	**7.15E-2**	**1.89E-1**	**3.20E-1**	**3.53E-1**	**6.29E-1**	**8.06E-1**	**1.11E-1**	**3.82E-1**	**5.28E-1**

G DYNAMICS FORECASTING: PROTOCOL DETAILS AND ADDITIONAL RESULTS

This section completes Sections 3.4 and 3.5: the protocol and the training budget of each model (Section G.1), per-dataset results after training from scratch (Section G.2), the fine-tuning experiment (Section G.3), and the controls (Section G.4).

G.1 PROTOCOL

Backbones and data. The two backbones are PDE-Transformer (Holzschuh et al., 2025) and the AViT of MPP (McCabe et al., 2024), both run on 2D and 3D inputs. Neither architecture is changed beyond the conditioning slot described below, and every model trains its backbone from scratch.

The eight training datasets are the pretraining systems of Section B other than Gray-Scott: Acoustic Scattering Maze, Active Matter, Euler Multi-Quadrants, Rayleigh-Bénard, Turbulent Radiative Layer 2D, MHD 64, Supernova Explosion 64, and Turbulence Gravity Cooling. Rayleigh-Taylor instability and Shear Flow are held out from training for generalization capacities evaluation. All datasets keep the preprocessing of Section B (Euler subsampled by two in space and time, Acoustic Scattering Maze restricted to frames 20–90, Gray-Scott to frames 0–300 subsampled by ten in time, Shear Flow subsampled by two in space and time), with Rayleigh-Taylor subsampled by two in space; the per-dataset window strides are replaced by the chunk stride of 30.

Chunks and models. Each trajectory is cut into consecutive, non-overlapping 30-frame chunks. Frames 1–10 of a chunk form its context and frames 11–30 its 20 targets, so every model trains and is evaluated on the same (chunk, target) pairs. All models are trained on one-step prediction.

As introduced in Section 3.4, we compare the standard ten-frame model (**base**) with our conditioned approach. We evaluate two variants of the conditioned model: $\textbf{TRACE}_\textbf{r}$ and $\textbf{TRACE}_\textbf{r}^\textbf{f}$. Both variants produce a regime embedding but $\textbf{TRACE}_\textbf{r}$ retains the backbone's native field embeddings, while the standard $\textbf{TRACE}_\textbf{r}^\textbf{f}$ variant instead replaces the backbone's field embeddings with its own extracted embeddings. $\textbf{TRACE}_\textbf{r}^\textbf{f}$ is the final version reported in the main results as **TRACE**.

Each variant is trained with the same NRMSE loss on next frame prediction. At validation, each model rolls out the 20 targets 11–30 autoregressively on its own predictions ; for conditioned variants the regime embedding and field embeddings are computed once, from frames 1–10.

Conditioning pathway. The conditioned models use our final pretrained encoder TRACE. For each chunk, the encoder's output tokens (patch tokens and global tokens) for the ten context frames are precomputed once and cached during training, so the encoder never runs in the training loop and is never updated. A trainable attentive extractor, optimized jointly with the backbone, reads these tokens and returns the regime embedding (together with one embedding per field for the $\textbf{TRACE}_\textbf{r}^\textbf{f}$ variant).

In PDE-Transformer, both the regime embedding and field embeddings are added to the backbone's native conditioning embedding, which modulates every block through zero-initialized adaptive layer normalization (AdaLN-Zero; Peebles & Xie, 2023). In **MPP** the regime embedding modulates each spatial axial-attention block through AdaLN-Zero modulation, and the field embeddings set the per-field weights of the patch embedding and output projection.

Budget and learning rates. Every model runs on one node of four H100 GPUs. The cosine learning-rate schedule anneals over the total number of optimization steps. Table 15 lists the configurations each model used; two main differences remain. (i) Learning rate: all PDE-Transformer TRACE models were trained with a learning rate of 10^{-4}, whereas the base model was trained at 2.5×10^{-5}; all MPP models use 2×10^{-4}. (ii) Steps: the total number of training steps varies per model depending on the architecture and throughput.

Validation and selection. Validation uses the validation split, the first ten trajectories of each dataset, evaluated every 25k steps on PDE-Transformer and every 12k to 15k steps on MPP; PDE-Transformer validates on all eleven datasets, and MPP on its eight training sets. Every model is read at the validated step with the lowest one-step validation error: on PDE-Transformer, the teacher-forced one-step validation NRMSE over the eight training sets; on MPP, the mean first-frame rollout NRMSE

Table 15: **Training budget of each forecasting model** (one node of four H100 GPUs). Steps: total optimization steps. Selected: the validated step with the lowest one-step validation error.

Backbone	Model	LR	Steps	Selected	Score
PDE-Transformer	base	2.5×10^{-5}	275k	275k	0.269
	TRACE_r	10^{-4}	220k	200k	0.219
	$\text{TRACE}_\text{r}^\text{f}$	10^{-4}	220k	200k	0.208
MPP (AViT)	base	2×10^{-4}	116k	108k	0.258
	TRACE_r	2×10^{-4}	93k	90k	0.270
	$\text{TRACE}_\text{r}^\text{f}$	2×10^{-4}	122k	120k	0.271

over its eight datasets. The rule is the same for every model of a backbone and looks only at next-step accuracy. Since selection and reporting use the same split, the numbers are not held-out test estimates. All downstream experiments, including zero-shot evaluation and fine-tuning, proceed directly from the models at these selected steps.

Rollout time. Timing runs the validation code path of each backbone with batch size 1 on one H100, at the precision used for validation, on the checkpoints of the downstream experiments. For the$\text{TRACE}_\text{r}^\text{f}$models the encoder runs live (no token cache), and the reported time includes computing the conditioning once per rollout (about 20 ms); we report the median of repeated timed passes after warm-up passes. The base and TRACE_r models cannot run on the held-out sets.

G.2 Per-dataset results after training from scratch

Tables 16 and 17 give every horizon for the three models detailed in Table 15. On PDE-Transformer, D has a lower first-frame error than the base on ten of the eleven datasets (all but Gray-Scott). On MPP, only D can be evaluated on the held-out sets. Table 18 gives the rollout time per dataset: the MPP TRACE models are 2.2 to 2.8 times faster than the base on every dataset, and the PDE-Transformer TRACE models 2% to 23% slower.

Table 16: **PDE-Transformer, per-dataset NRMSE after training from scratch** (test split, each model at its selected step; lower is better). **base**: ten input frames; **TRACE_r**, **$\text{TRACE}_\text{r}^\text{f}$**: one input frame plus the learned regime embedding, using native backbone or extractor field embeddings, respectively. 1st: first predicted frame; 1:10, 10:20: means over rollout steps 1–10 and 11–20. Held-out rows: datasets absent from the backbone's training; Gray-Scott, unlike the other two, belongs to the encoder's pretraining mixture. Bold: best per dataset and horizon.

	1st			1:10			10:20		
Dataset	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$
Acoustic Scatt. Maze	0.202	0.148	**0.146**	1.046	0.871	**0.839**	**1.162**	1.224	1.412
Active Matter	0.152	0.150	**0.138**	1.557	0.670	**0.655**	2.793	**1.618**	1.741
Euler Multi-Quad.	0.368	0.300	**0.292**	**1.047**	1.154	1.052	**1.435**	3.837	6.499
Rayleigh-Bénard	0.181	0.139	**0.132**	**0.697**	0.733	0.761	**1.278**	10.435	16.405
Turb. Rad. Layer 2D	0.336	0.302	**0.295**	1.325	1.516	**1.154**	**2.061**	2.592	4.422
MHD 64	0.399	0.335	**0.328**	1.198	0.988	**0.933**	**1.946**	2.689	2.317
Supernova Expl. 64	0.757	0.615	**0.596**	3.262	**0.926**	0.927	4.215	**1.025**	1.088
Turb. Grav. Cool.	0.244	0.155	**0.141**	1.124	0.437	**0.428**	1.595	**0.869**	0.915
Held out from the backbone's training (zero-shot)									
Rayleigh-Taylor	–	–	**0.382**	–	–	**0.590**	–	–	**0.776**
Shear Flow	–	–	**0.360**	–	–	**0.841**	–	–	1.866

G.3 Fine-tuning on single datasets

Protocol. Every model detailed in Table 15 is fine-tuned separately on each of six datasets: four of the backbones' training sets (Active Matter, Rayleigh-Bénard, Supernova Explosion 64, TGC) and two systems outside both the backbones' training and TRACE's pretraining (Rayleigh-Taylor,

Table 17: **MPP (AViT), per-dataset NRMSE after training from scratch** (validation split; same horizons and bold rule as Table 16). Held-out rows: zero-shot evaluation of the final weights of TRACE; the other models take their field embeddings from a per-dataset table that has no entry for an unseen system (–).

	1st			1:10			10:20		
Dataset	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$
Acoustic Scatt. Maze	**0.062**	0.075	0.097	**0.126**	0.175	0.190	**0.245**	0.349	0.341
Active Matter	**0.140**	0.148	0.142	0.419	0.381	**0.358**	1.10	0.972	**0.924**
Euler Multi-Quad.	0.159	0.165	**0.149**	0.369	0.417	**0.352**	0.743	0.775	**0.638**
Rayleigh-Bénard	0.168	0.170	**0.149**	0.315	0.358	**0.261**	0.553	0.682	**0.471**
Turb. Rad. Layer 2D	**0.280**	0.294	0.287	**0.491**	0.547	0.524	**0.836**	0.897	0.871
MHD 64	**0.441**	0.452	0.458	**0.778**	0.795	0.809	**1.02**	1.05	1.20
Supernova Expl. 64	**0.642**	0.668	0.720	0.811	**0.805**	0.871	1.01	**0.950**	1.09
Turb. Grav. Cool.	0.173	0.188	**0.168**	0.471	0.481	**0.415**	0.906	0.918	**0.862**
Held out from the backbone's training (zero-shot; defined for $\text{TRACE}_\text{r}^\text{f}$ only)									
Rayleigh-Taylor	–	–	**0.758**	–	–	**1.10**	–	–	**3.10**
Shear Flow	–	–	**0.454**	–	–	**0.948**	–	–	**4.26**

Table 18: **Rollout time per dataset** (ms for one 20-step rollout, batch 1, one H100, median of repeated passes;$\text{TRACE}_\text{r}^\text{f}$context included). MPP's base and TRACE_r cannot run on the held-out sets (–).

	PDE-Transformer			MPP (AViT)		
Dataset	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$	base	TRACE_r	$\text{TRACE}_\text{r}^\text{f}$
Acoustic Scatt. Maze	327	387	395	1311	520	526
Active Matter	507	621	624	1311	577	582
Euler Multi-Quad.	327	389	395	1307	519	485
Rayleigh-Bénard	291	341	343	1299	508	491
Turb. Rad. Layer 2D	265	298	302	1087	484	437
MHD 64	364	372	375	1194	487	442
Supernova Expl. 64	322	332	333	1241	484	457
Turb. Grav. Cool.	322	331	333	1244	484	442
Held out						
Rayleigh-Taylor	276	288	291	–	–	449
Shear Flow	236	265	266	–	–	440

3D, and Shear Flow, 2D). Runs start from the weights listed in Table 15 and use one protocol per backbone: 10,000 steps on one node of four H100 GPUs, global batch 16, a fresh AdamW optimizer, 500 linear warmup steps followed by cosine decay, and learning rate 2.5×10^{-5} on PDE-Transformer and 5×10^{-5} on MPP for all three models. Validation uses the windows of the 24-hour runs, at step 0 and at the end of fine-tuning (every 1,000 steps for the PDE-Transformer's base runs), and each run is read at its best validated step, step 0 included. On an unseen system, MPP's base and TRACE_r receive a new, randomly initialized entry of MPP's per-dataset field table, trained with the rest; their step-0 evaluation is undefined. Table 19 gives the step-0 and final first-frame errors and the longer horizons of every run.

Table 19: **Fine-tuning on single datasets, all runs** (NRMSE on the validation split, lower is better). 1st@0, 1st@10k: first-frame error before and after 10k steps of fine-tuning; 1:10, 10:20: rollout errors at the step that is read (best validated step, step 0 included). –: step 0 undefined (new field-table entry). Rayleigh-Taylor and Shear Flow are unseen by the backbones and by TRACE.

	base				**TRACE_r**				$\text{TRACE}_\text{r}^\text{f}$			
Dataset	1st@0	1st@10k	1:10	10:20	1st@0	1st@10k	1:10	10:20	1st@0	1st@10k	1:10	10:20
PDE-Transformer (lr 2.5×10^{-5})												
Active Matter	0.152	0.361	1.518	2.823	0.150	0.141	0.619	1.497	0.138	0.131	0.597	1.549
Rayleigh-Bénard	0.184	0.332	0.760	1.535	0.139	0.128	0.741	12.723	0.132	0.122	0.763	14.710
Supernova Expl. 64	0.767	1.255	2.488	2.685	0.615	0.594	0.903	1.044	0.596	0.585	0.908	1.053
Turb. Grav. Cool.	0.243	1.037	1.056	1.357	0.155	0.136	0.401	0.800	0.141	0.129	0.409	0.873
Rayleigh-Taylor	–	0.588	0.883	0.999	0.432	0.111	0.389	0.713	0.382	0.110	0.361	0.666
Shear Flow	–	0.214	0.798	0.891	0.397	0.024	0.228	0.622	0.360	0.022	0.185	0.538
MPP (AViT) (lr 5×10^{-5})												
Active Matter	0.140	0.128	0.352	0.955	0.150	0.135	0.338	0.898	0.141	0.132	0.331	0.927
Rayleigh-Bénard	0.168	0.151	0.279	0.509	0.166	0.145	0.248	0.445	0.151	0.140	0.242	0.440
Supernova Expl. 64	0.642	0.645	0.810	1.008	0.671	0.668	0.800	0.934	0.718	0.717	0.874	1.142
Turb. Grav. Cool.	0.172	0.160	0.436	0.895	0.185	0.164	0.395	0.801	0.168	0.155	0.386	0.844
Rayleigh-Taylor	–	0.425	0.726	3.626	–	0.421	0.672	1.452	0.766	0.429	0.726	2.111
Shear Flow	–	0.068	0.272	0.665	–	0.079	0.209	0.318	0.453	0.079	0.160	0.268

G.4 CONTROLS

To ensure the forecasting backbones actively utilize the regime-specific information rather than treating the regime embedding as a static bias, we conduct controls that replace or shuffle the regime embeddings, both at evaluation and during fine-tuning.

Replacement at evaluation. As introduced in Section 3.5, we take the models trained in Section G.2 and replace their regime embedding during the test rollout. We compare the model's true embeddings against two replacements:

1. *Random*: A Gaussian draw matching the per-dimension statistics of the dataset's evaluated windows.
2. *Shuffled*: An embedding drawn from a randomly chosen trajectory within the same dataset.

For $\textbf{TRACE}_\textbf{r}^\textbf{f}$, the extractor field embeddings are replaced in the same way.

Table 20 reports the first-frame error per dataset under these replacements. Consistent with the aggregate findings in the main text, replacing the model's true regime embedding generally degrades predictive performance, confirming that the backbones actively rely on the trajectory-specific physical conditioning provided by the encoder.

Replacement during fine-tuning. We extend this control to the fine-tuning regime to verify the networks do not simply learn to ignore the regime embedding when adapting to a specific dataset. Following the *Shuffled* replacement strategy described in Section 3.5, we fine-tune each conditioned model using the standard protocol, but shuffle the learned regime embeddings (and extractor field embeddings for the $\textbf{TRACE}_\textbf{r}^\textbf{f}$ variant) across the batch during both training and evaluation. Table 21 demonstrates that fine-tuning with these shuffled embeddings yields consistently higher errors than standard fine-tuning across all tested datasets, further validating that the backbones actively leverage the trajectory-specific physical representations to guide their dynamics.

Table 20: **Replacement of the regime embedding at evaluation, per dataset** (first-frame NRMSE, test split, lower is better). Own: the model's true embeddingss.

	$\textbf{TRACE}_\textbf{r}$			$\textbf{TRACE}_\textbf{r}^\textbf{f}$		
Dataset	Own	Random	Shuffled	Own	Random	Shuffled
PDE-Transformer						
Acoustic Scatt. Maze	0.148	0.195	0.207	0.146	0.215	0.252
Active Matter	0.150	0.230	0.272	0.138	0.228	0.268
Euler Multi-Quad.	0.300	0.397	0.472	0.292	0.410	0.507
Rayleigh-Bénard	0.139	0.240	0.269	0.132	0.238	0.287
Turb. Rad. Layer 2D	0.302	0.361	0.393	0.295	0.350	0.383
MHD 64	0.335	0.375	0.399	0.328	0.380	0.416
Supernova Expl. 64	0.615	0.618	0.617	0.596	0.598	0.599
Turb. Grav. Cool.	0.155	0.159	0.160	0.141	0.162	0.195
MPP (AViT)						
Acoustic Scatt. Maze	0.072	0.132	0.130	0.097	0.488	0.176
Active Matter	0.150	0.303	0.297	0.141	0.287	0.300
Euler Multi-Quad.	0.167	0.269	0.370	0.155	0.271	0.381
Rayleigh-Bénard	0.166	0.238	0.266	0.151	0.387	0.282
Turb. Rad. Layer 2D	0.297	0.388	0.377	0.287	0.425	0.369
MHD 64	0.458	0.476	0.488	0.460	0.525	0.509
Supernova Expl. 64	0.673	0.673	0.673	0.716	0.732	0.715
Turb. Grav. Cool.	0.186	0.195	0.202	0.168	0.179	0.191

Table 21: **Replacement of the regime embedding during fine-tuning, all horizons** (NRMSE after 10k steps, test split, lower is better). TRACE: regular fine-tune; shuf.: fine-tune with shuffled embeddings in training and evaluation. Bold: the lower of each pair.

	$\textbf{TRACE}_\textbf{r}$						$\textbf{TRACE}_\textbf{r}^\textbf{f}$					
	1st		1:10		10:20		1st		1:10		10:20	
Dataset	TRACE	Shuffled	TRACE	Shuffled	TRACE	Shuffled	TRACE	Shuffled	TRACE	Shuffled	TRACE	Shuffled
PDE-Transformer												
Active Matter	**0.141**	0.149	**0.619**	0.758	1.497	**1.318**	**0.131**	0.140	**0.597**	0.810	1.549	**1.417**
Rayleigh-Bénard	**0.128**	0.142	0.741	**0.537**	12.723	**1.741**	**0.122**	0.139	0.763	**0.470**	14.710	**1.419**
Supernova Expl. 64	0.594	0.594	**0.903**	0.917	**1.044**	1.082	**0.585**	0.590	**0.908**	0.956	**1.053**	1.218
Turb. Grav. Cool.	**0.136**	0.142	**0.401**	0.424	**0.800**	0.851	**0.129**	0.142	**0.409**	0.414	0.873	**0.824**
MPP (AViT)												
Active Matter	**0.135**	0.146	**0.338**	0.358	**0.898**	0.947	**0.132**	0.143	**0.331**	0.343	**0.927**	0.947
Rayleigh-Bénard	**0.145**	0.153	**0.248**	0.266	**0.445**	0.469	**0.140**	0.167	**0.242**	0.300	**0.440**	0.537
Supernova Expl. 64	0.668	0.668	0.800	**0.796**	0.934	**0.920**	0.717	**0.660**	0.874	**0.804**	1.142	**0.949**
Turb. Grav. Cool.	**0.164**	0.175	**0.395**	0.443	**0.801**	0.827	**0.155**	0.177	**0.386**	0.447	0.844	**0.841**

H EXTENDED VISUALIZATIONS

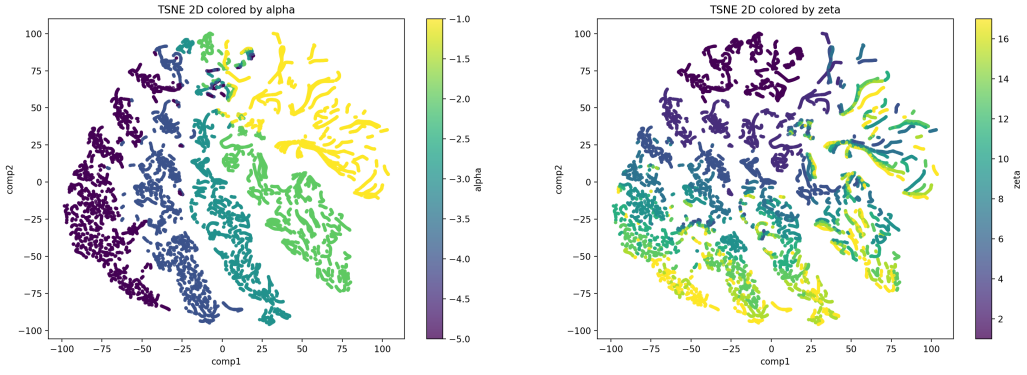
This appendix provides extended qualitative evaluations of the TRACE representation and its downstream forecasting performance, complementing the analysis in the main text. To thoroughly examine both the latent space geometry and the physical fidelity of the surrogate, we divide these visualizations into three parts:

- **Global Latent Space (t-SNE):** In Section H.1, we project the global tokens of the frozen encoder into 2D to observe how the model organizes physical regimes. We detail the RGB parametric mapping used for Active Matter and explore additional systems to visualize how the encoder captures continuous physical parameters versus highly entangled ones.
- **Patch-Level Features (PCA):** In Section H.2, we analyze the encoded frame tokens using Principal Component Analysis (PCA) to investigate how the model captures spatial structures, inter-field correlations, and physical discontinuities across different datasets.
- **Forecasting Rollouts:** Finally, in Section H.3, we present visual comparisons between the ground truth trajectories and the autoregressive predictions of the TRACE-conditioned downstream models, demonstrating the practical efficacy of the learned representations.

H.1 GLOBAL LATENT SPACE (T-SNE)

Active Matter and RGB Parametric Mapping. In the main text (Figure 3b), we visualized the Active Matter trajectories using an RGB composite colormap. This system is governed by three primary parameters: the domain size L, the activity parameter α, and the active tension ζ. To visualize how the encoder captures the simultaneous variation of these parameters, we map them directly to color channels: $L \rightarrow$ Red, $\alpha \rightarrow$ Green, and $\zeta \rightarrow$ Blue.

To unpack this composite visualization, Figure 8 shows the exact same t-SNE projection colored by α and ζ individually (L is actually constant). The projection reveals clear, orthogonal gradients for each parameter. The smooth transition of colors confirms that the encoder does not merely memorize distinct environments, but learns a continuous, structured manifold representing the parametric physical variations.



(a) Colored by α.

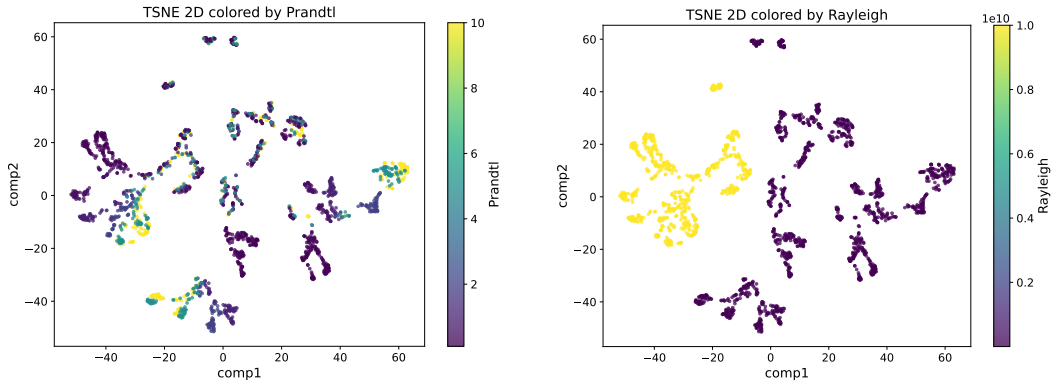
(b) Colored by ζ.

Figure 8: **Active Matter latent space.** Individual parameter gradients composing the RGB visualization from the main text. The latent space exhibits smooth, continuous transitions corresponding to the underlying physics.

Regime Structuring in Additional Systems. We extend this analysis to Rayleigh-Bénard convection (Figure 9) and Turbulence Gravity Cooling (Figure 10).

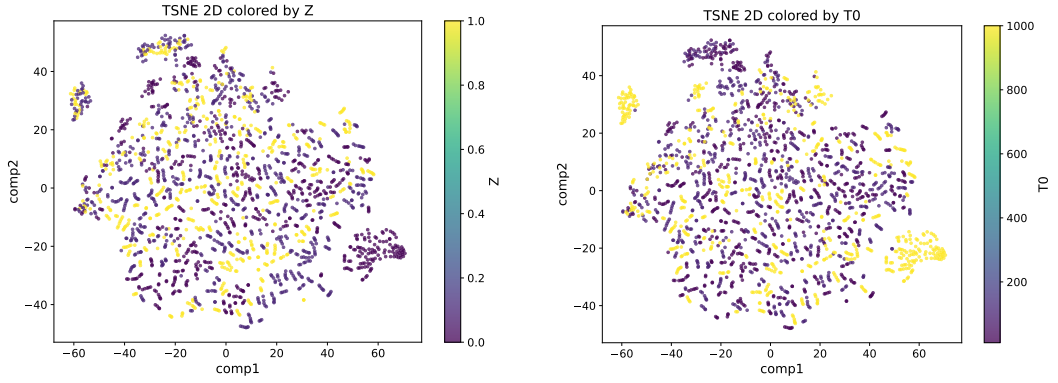
For Rayleigh-Bénard, the system's underlying dynamics are clearly separable: the trajectories perfectly cluster according to the Rayleigh number, making it very easy to classify, while the Prandtl number reveals a continuous, structured gradient across the latent space.

In stark contrast, the latent space for Turbulence Gravity Cooling is highly entangled and poorly structured. Both parameters are completely entangled and lack any clean structural separation. This visual entanglement perfectly aligns with our quantitative findings in Table 2, where all baseline models, including TRACE, struggled to accurately regress T_0. Furthermore, this disorganized representation directly relates to the difficulties observed in our forecasting results (Section G.2): Turbulence Gravity Cooling proved to be an exceptionally challenging system to model, suffering from severe error accumulation and instability at longer rollout horizons. The messy representation reflects the model's inability to cleanly decouple the driving physical parameters of this highly chaotic 3D system, which inevitably hurts downstream forecasting.



(a) Colored by Prandtl. (b) Colored by Rayleigh.

Figure 9: **Rayleigh-Bénard latent space.** The system is easily separable, with Rayleigh forming distinct clusters and Prandtl driving a continuous gradient.



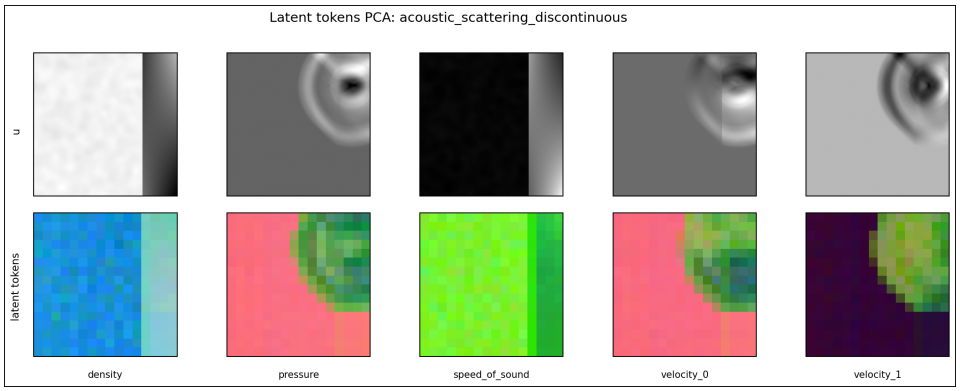
(a) Colored by metallicity (z). (b) Colored by initial temperature (T_0).

Figure 10: **Turbulence Gravity Cooling latent space.** The latent space is highly entangled with poorly separated parameters like T_0, correlating with the system's poor downstream forecasting performance.

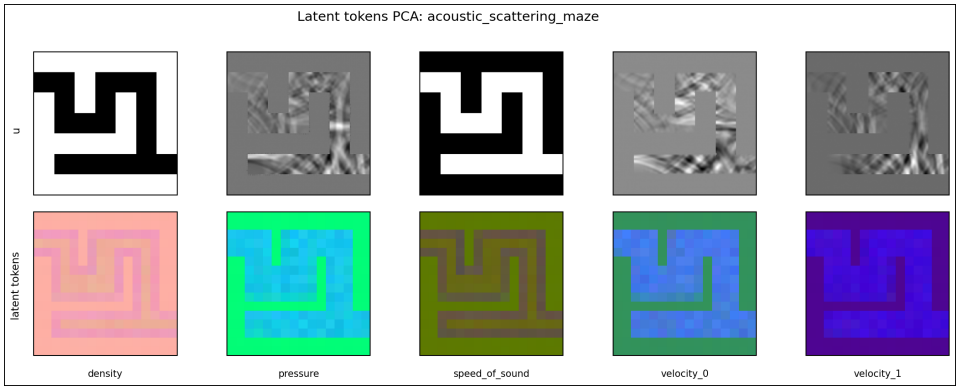
H.2 PCA ON THE LATENT TOKENS

In addition to preserving spatial features in latent space, it is worth noting that encoded features have a much higher variance across fields than across spatial positions, and that despite these large differences, certain fields share similarities:

- **Acoustic Scattering**: propagating waves in pressure and velocities share a green (resp. blue) component in 11a (resp. 11b) ;



(a)



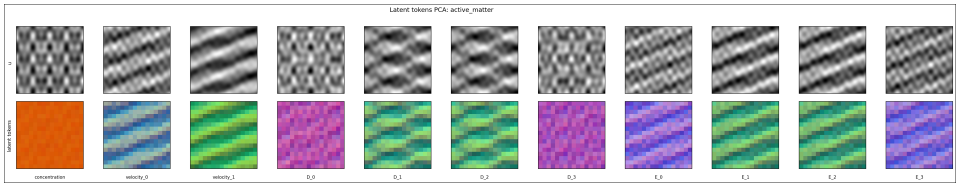
(b)

Figure 11: PCA on the latent tokens on Acoustic Scattering.

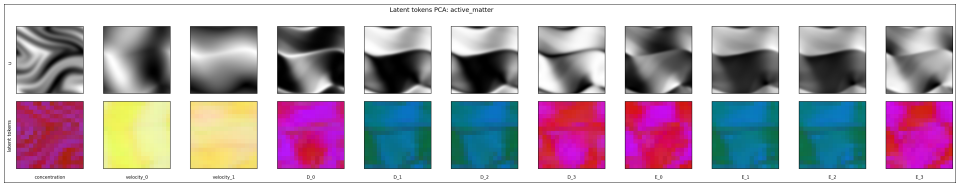
- **Active Matter**: rather than fields' nature, it seems to be the axis of the orientation tensor D and strain-rate tensor E. Axes 1 and 2 consistently sharing the same average feature across trajectories, and axes 0 and 3 having a close one;
- **Euler Multi Quadrants**: energy and pressure, as well as momentum 0 and 1, seem correlated in feature space;

It is also interesting to notice:

- in **Acoustic Scattering**, how borders between discontinuities in density or speed of sound seem to have a special focus in latent space that is not present on the time-dependent fields;
- in 11a, how inter-field attention allows for the discontinuity known in the density and speed of sound fields to be shared to tokens of the other fields that should not be aware of this discontinuity until the acoustic wave reaches their position;
- a clear shock line identification in 13b (in green in the energy and pressure latents);

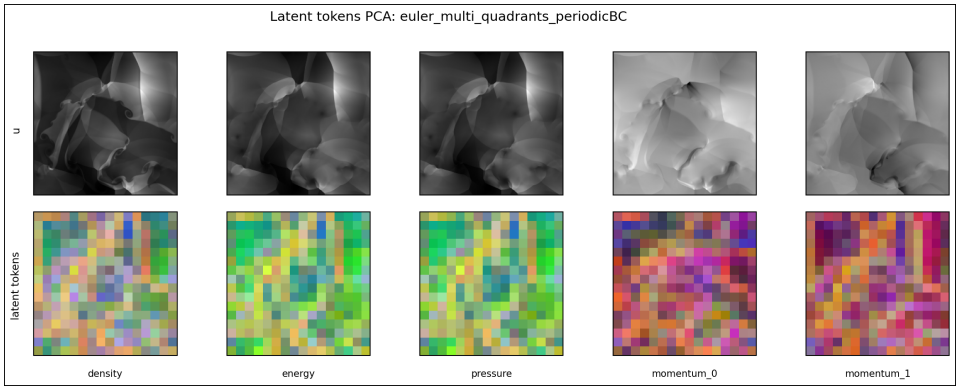


(a)

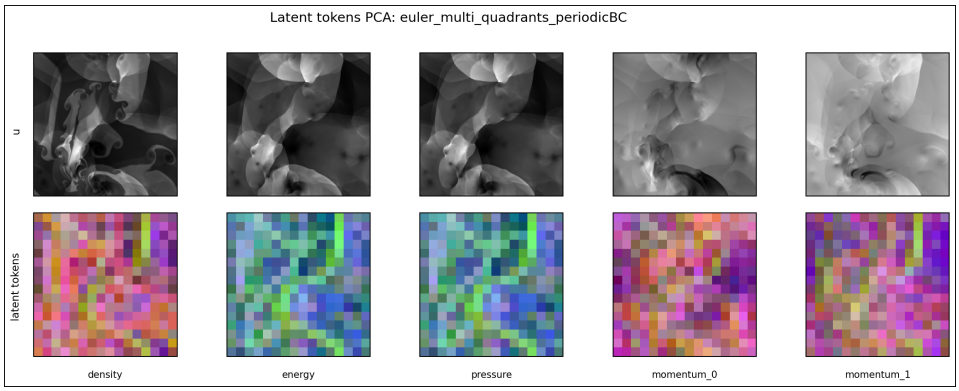


(b)

Figure 12: PCA on the latent tokens on Active Matter.



(a)



(b)

Figure 13: PCA on the latent tokens on Euler Multi Quadrants Periodic BC.

H.3 Forecasting Rollout Visualizations

To qualitatively assess the forecasting performance of the downstream models conditioned on the TRACE representation, we visualize the autoregressive rollouts against the ground truth. We present results for two structurally diverse systems: Rayleigh-Bénard Convection (Figure 14) and Euler Multi-Quadrants (Figure 15).

In each figure, we compare the ground truth trajectory against the predictions of the four downstream architectures: PDE-Transformer (base and TRACE) and MPP (base and TRACE). Recall that the standard base models process a full window of ten past frames, whereas the TRACE-conditioned models only receive the single preceding frame alongside the latent TRACE representation.

Observation of Physical Dynamics. Across both datasets, the TRACE-conditioned models successfully capture both the large-scale macroscopic structures and the fine-grained dynamics of the underlying physics:

- **Rayleigh-Bénard** (Figure 14): The system is dominated by thermal convection plumes. We observe a clear degradation in the standard PDE-Transformer (base) over time, as it struggles to maintain the structural integrity of the plumes. In contrast, the TRACE-conditioned PDE-Transformer retains a much better form and avoids this rapid smoothing over long horizons. For the MPP architecture, the visual fidelity remains comparable between the base and TRACE-conditioned variants, with both successfully evolving the convection dynamics at a similar level of accuracy.
- **Euler Multi-Quadrants** (Figure 15): This system features sharp shockwaves interacting under periodic boundary conditions. The standard PDE-Transformer (base) struggles early on, producing initial predictions that rapidly deteriorate and introduce severe unphysical oscillations. However, the TRACE-conditioned variant starts with much sharper, accurate shockfronts and degrades at a significantly slower rate. This visual evidence perfectly aligns with our quantitative forecasting results, confirming that the task-agnostic TRACE embeddings provide a more robust physical context for the downstream solver than a dense temporal history.

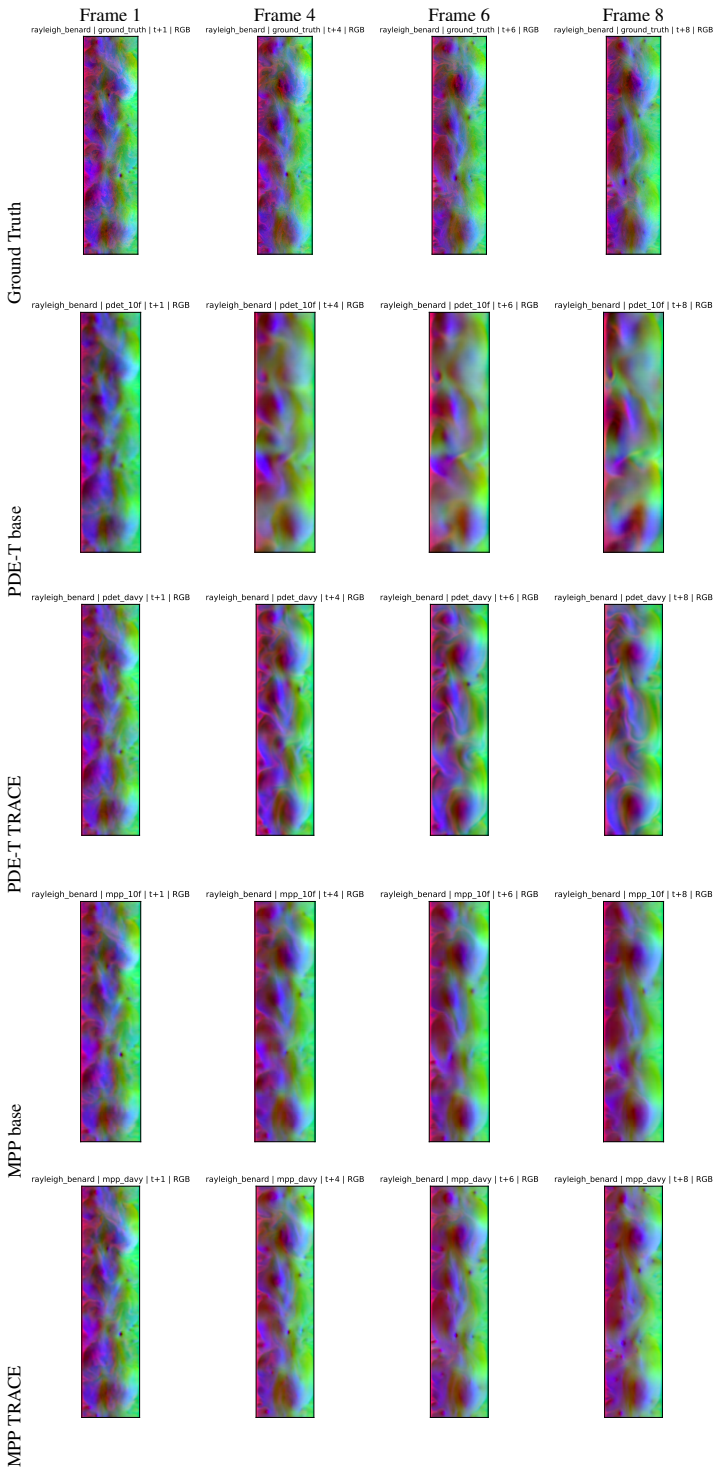


Figure 14: **Rayleigh-Bénard Rollout.** Visualizing the evolution of thermal convective plumes comparing 10-frame baseline models against TRACE-conditioned variants.

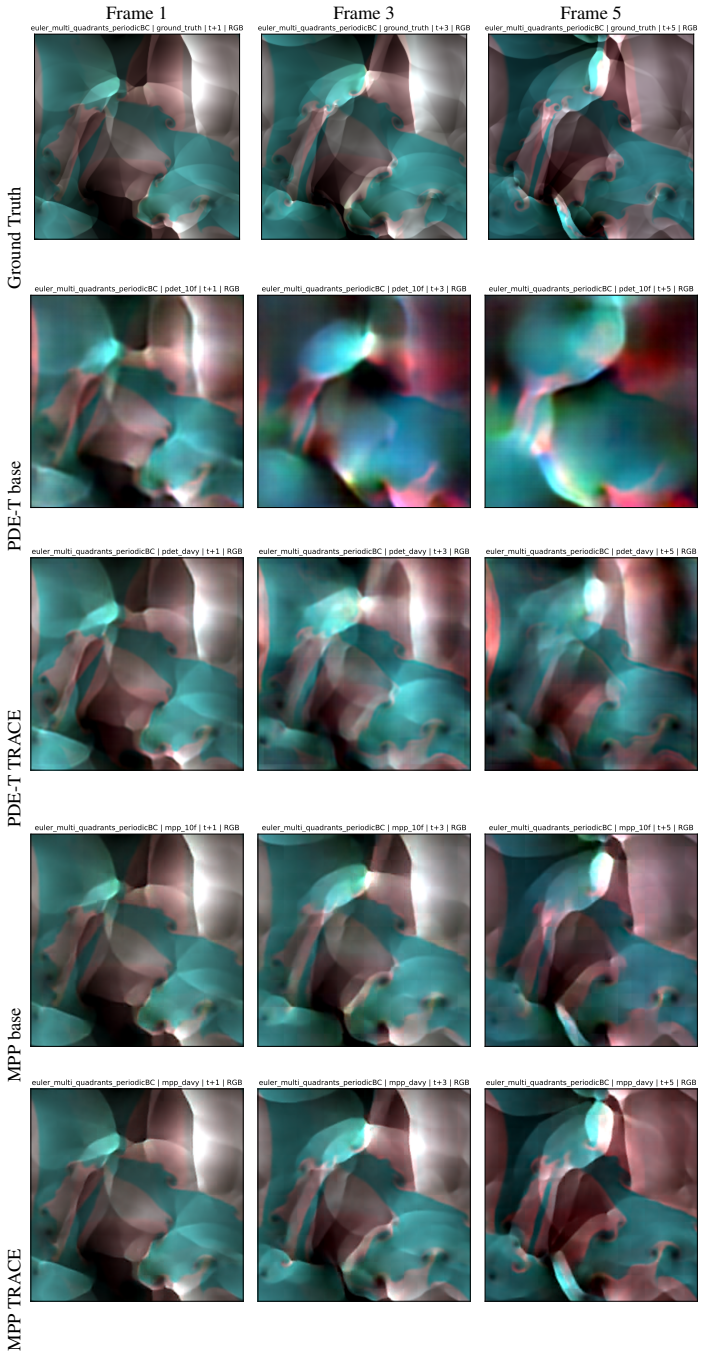


Figure 15: **Euler Multi-Quadrants Rollout.** Evaluation of sharp shockwave discontinuities interacting over time.