LEARNING BEFORE PREDICTING: PHYSICS-GUIDED JOINT-EMBEDDING LEARNING FOR SCIENTIFIC SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific systems are heterogeneous twice: they live in different spatial dimensions, and their states contain quantities with different geometric laws. Density is a scalar, velocity is a vector, and orientational order is a tensor, yet learning pipelines often flatten them into interchangeable channels and ask a downstream predictor to recover both dynamics and structure from scarce labels. We ask a different question: *can the physical organization of a state be learned before forecasting?* We introduce UNIJEPA, a physics-guided joint-embedding framework that keeps field type explicit at the interface while a shared predictive core learns coupled spatiotemporal dynamics. During pretraining, lightweight field decoders expose predicted streams to system-specific equations, differential relations, realizability constraints, and transformation laws; the decoders are removed before downstream prediction. Across 1D–3D scalar PDEs, 2D coupled flow, and scalar–vector–tensor Active Matter, the same parameterization principle improves few-shot transfer. On Active Matter with 16 labeled trajectories, UNIJEPA reduces full-state VRMSE and velocity error by 13.8% and 20.4%, respectively, and outperforms parameter-matched Scratch FNO by 10.1%. Frozen probes and controlled ablations further support the benefits of physically organized representations.

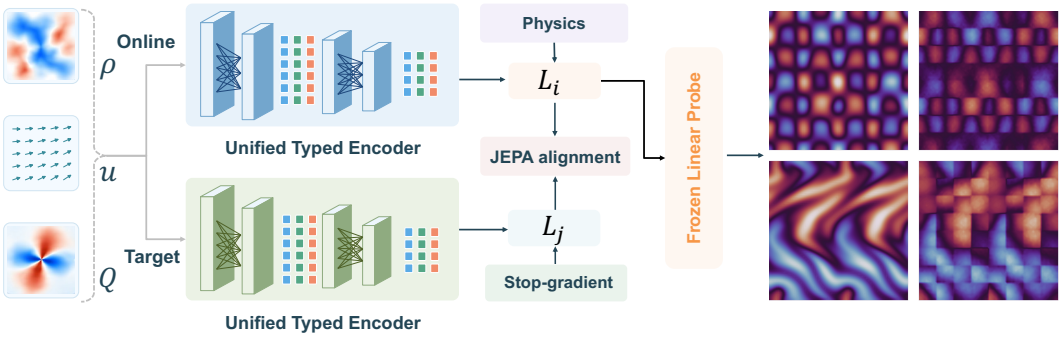

Figure 1: **Learning physical organization before forecasting.** UNIJEPA combines typed field encoding with physics-guided joint-embedding learning to organize scalar, vector, and tensor fields in a shared predictive space. Physical-field readouts illustrate structure retained in the learned representations.

1 INTRODUCTION

AI is rapidly advancing scientific modeling, from learning solution operators to building transferable models across physical systems (Li et al., 2021; McCabe et al., 2024; Herde et al., 2024). Yet scientific states vary across spatial dimensions and geometric types. Density is a scalar, velocity components define a vector, and orientation components jointly describe a tensor. These quantities follow different transformation laws and interact through physical relations: for example, strain

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.

depends on spatial derivatives of velocity. Forecasting such systems requires learning both temporal dynamics and the relationships among heterogeneous fields.

Many learning pipelines represent these quantities as ordinary input channels, retaining their numerical values while leaving their physical relationships implicit. Expressive neural operators and Transformers can learn these relationships from data. Under few-shot transfer, however, the limited labeled trajectories must support learning the target dynamics while also identifying how field components relate and transform together. Known variable semantics provide information that need not be rediscovered entirely through downstream supervision. This raises our central question: *Can the physical organization needed for forecasting be learned before fitting a predictor with scarce labels?*

We separate reusable predictive representation learning from fitting the final forecaster. The representation must capture both what each field is and how the fields interact. Scalars, vectors, and tensors follow different transformation rules, yet jointly describe the evolution of the same physical system. This motivates a shared predictive space in which field identities remain explicit and interactions can be learned across fields. Temporal prediction gives this space a concrete learning objective: using past observations to anticipate the system's future state.

Joint-embedding predictive learning offers a formulation in which the transferable representation itself is the prediction target (Assran et al., 2023; Bardes et al., 2024). A context encoder processes past observations, a target encoder represents the observed future state, and a predictor learns to map the former representation to the latter. Their agreement directly supervises future-feature prediction in the learned representation space. This is particularly relevant to our transfer setting because both the encoder and predictor are reused: the downstream neural operator receives features of an anticipated future state alongside the raw observations. JEPA thus connects the pretraining task to the information supplied at transfer, allowing temporal predictive structure to be learned before the final state-space forecaster is fitted with limited labels.

The remaining difficulty is that agreement in representation space does not by itself specify physical consistency. A latent distance does not explicitly test whether vector components transform jointly or whether predicted strain agrees with velocity derivatives. Leaving these properties entirely implicit would leave part of the original learning burden unresolved. Field types provide a way to retain component grouping and geometric identity, while decoding predicted features into physical quantities makes their relationships directly testable during pretraining. Transformation behavior is one such requirement and can also be enforced through equivariant architectures (Satorras et al., 2021; Helwig et al., 2023). Differential consistency and realizability impose additional requirements that equivariance alone does not supply. This analysis motivates coupling future-representation prediction with explicit field organization and physical supervision. Through the decoders, physical constraints provide a training signal to the encoder and predictor, encouraging future representations to retain the relationships among fields. These relationships are learned during pretraining and carried into downstream forecasting through the predictive features. The few-shot forecaster can therefore build on an existing organization of the physical state when learning the target dynamics.

We introduce UNIJEPA, a physics-guided joint-embedding framework that implements this principle. Each named field enters through an interface matched to its scalar, vector, or tensor type. Cross-field interaction and a shared Transformer then learn the coupled predictive representation. During pretraining, lightweight decoders map predicted future features to physical quantities, allowing equations, differential relations, realizability constraints, and transformation laws to supervise the representation. At transfer time, these decoders are removed. The pretrained encoder and predictor supply field-level predictive streams to a downstream FNO through bounded residual adapters, alongside the raw context states. Figure 2 summarizes how this design connects physically organized pretraining with few-shot forecasting.

Across 1D–3D scalar systems, coupled flow, and Active Matter, UNIJEPA consistently improves few-shot transfer. At the 16-trajectory budget, it reduces error by 13.5% on the 3D system and lowers CFD2D full-state error by 8.0% over Scratch FNO. On Active Matter, it reduces five-seed full-state VRMSE and velocity error by 13.8% and 20.4%, remains 10.1% better than a parameter-matched Scratch control, and improves over an exactly C_4-equivariant $p4$-GFNO by 8.2% on full-state VRMSE. Frozen linear probes further reduce mean orientation and strain readout error from 0.264 (JEPA) to 0.154, directly revealing physical organization in the learned representation.

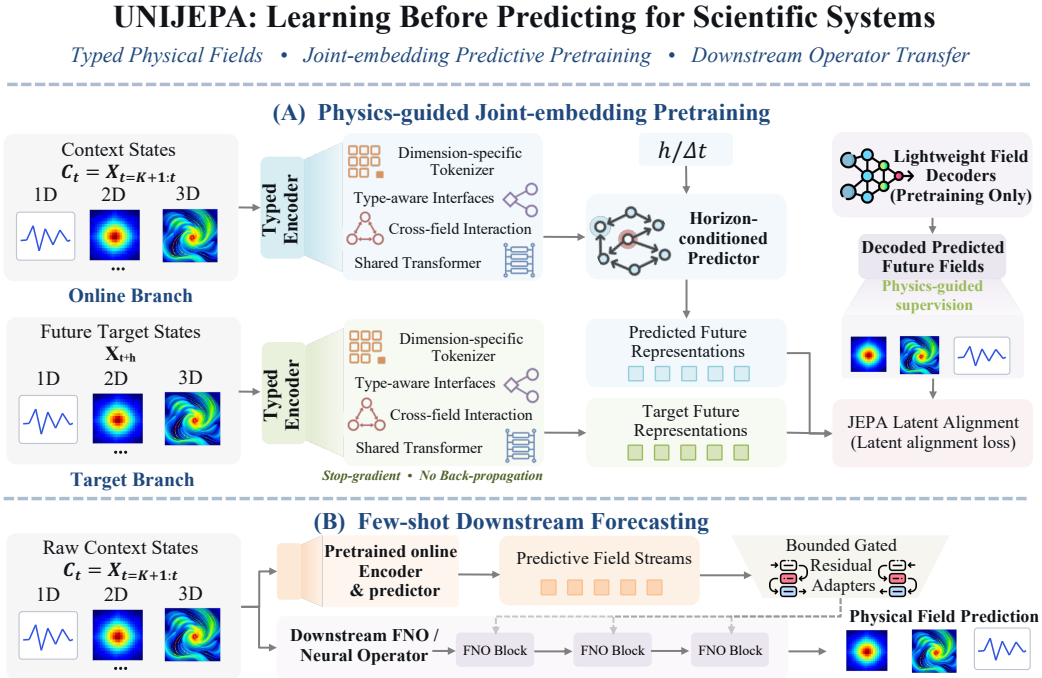

Figure 2: Overview of UNIJEPA. (A) A shared JEPA predicts future representations of typed 1D–3D fields, while pretraining-only decoders apply system-specific physical supervision. (B) During few-shot transfer, the decoders and target branch are removed, and typed predictive streams enter an FNO through bounded residual adapters.

Our contributions are:

- We formulate heterogeneous scientific states as typed physical fields and map scalar, vector, and tensor quantities into a shared predictive space using one parameterized core with type-aware interfaces.
- We use applicable physical relations and transformation laws to supervise predicted field representations during JEPA pretraining, allowing physics to shape the latent space *before* downstream forecasting.
- We validate the same modeling principle across scalar, coupled-flow, and scalar–vector–tensor regimes, and use five-seed, capacity-matched, exactly equivariant, frozen-probe, and field-typing controls to isolate where the gain comes from.

2 Related Work

Scientific forecasting and transferable solvers. PINNs constrain solutions with governing equations (Raissi et al., 2019), while graph simulators predict particle or mesh dynamics (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021). DeepONet and FNO learn operators between function spaces (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023), and PINO combines operator learning with PDE constraints (Li et al., 2024). MWT learns operators in multiwavelet bases (Gupta et al., 2021), while CNO uses convolutional operator layers (Raonic et al., 2023). PDE-Transformer uses channel-wise attention (Holzschuh et al., 2025), and Transolver attends over learned mesh slices (Wu et al., 2024). Multiple Physics Pretraining, DPOT, and Poseidon transfer across equations (McCabe et al., 2024; Hao et al., 2024; Herde et al., 2024); MORPH handles heterogeneous dimensions and modalities (Rautela et al., 2026). Our complementary goal is to learn reusable future representations that preserve field geometry before downstream forecasting.

Self-supervised and latent dynamics learning. Representation learning for dynamics includes reduced latent evolution, masked reconstruction, and predictive feature learning. Koopman autoencoders seek coordinates with approximately linear evolution (Lusch et al., 2018), whereas masked autoencoders learn transferable PDE features by reconstructing missing observations (Zhou & Farimani, 2024). I-JEPA predicts target-region embeddings from visible context (Assran et al., 2023), and V-JEPA extends feature prediction to video (Bardes et al., 2024). Recent work compares JEPA with masked reconstruction on spatiotemporal physical systems (Qu et al., 2026), while AeroJEPA adapts predictive latent learning to three-dimensional aerodynamic surrogates (Giral et al., 2026). Feature prediction can emphasize evolution-relevant structure without reconstructing every state value or prescribing linear latent dynamics. However, generic JEPA objectives do not distinguish the transformation roles of scalar, vector, and tensor fields.

Physics-guided representation learning. Physics can supervise the latent space rather than only the decoded solution. Concurrent Phys-JEPA separates multivariate time-series latents into physical and residual components and constrains latent states and transitions (Nie et al., 2026). REPA-P attaches lightweight first-principles decoders to intermediate scientific diffusion features and removes them at inference (Jia et al., 2026). UNIJEPA instead makes field role the common organizing variable across encoding, future-latent prediction, physical supervision, and neural-operator transfer, while retaining the efficiency of removable pretraining-only decoders.

Geometric and typed physical fields. Equivariant networks encode prescribed responses to group actions (Satorras et al., 2021; Helwig et al., 2023). Tensor Field Networks explicitly propagate scalar, vector, and higher-order features under rotations (Thomas et al., 2018), while Clifford neural layers represent multiple geometric grades within a structured algebra (Brandstetter et al., 2023). These approaches impose geometric behavior within particular architectures. UNIJEPA addresses a complementary problem: maintaining named field identity inside a shared predictive representation and exposing each typed future stream to a downstream operator. We separately evaluate an exactly equivariant $p4$-GFNO control to distinguish physically organized pretraining from hard-coded symmetry alone.

3 METHOD

3.1 ONE PREDICTIVE SPACE FOR HETEROGENEOUS FIELDS

Let s index a system on spatial domain $\Omega_s \subset \mathbb{R}^{d_s}$, with dimension $d_s \in \{1, 2, 3\}$. We represent its state as named physical fields,

$$X_t^{(s)} = \left\{ \left(x_t^{(s,f)}, \iota_f, \tau_f \right) : f \in \mathcal{F}_s \right\}, \qquad \tau_f \in \{\text{scalar, vector, tensor}\}, \tag{1}$$

where t indexes time, $\mathcal{F}_s$ fields, $x_t^{(s,f)}$ field f, and ι_f, τ_f its identity and type. Vector and tensor components are grouped before tokenization so that their joint transformation law is not reduced to independent scalar channels.

For K context states $C_t^{(s)} = X_{t-K+1:t}^{(s)}$, a dimension-specific tokenizer P_{d_s} maps each native field to tokens,

$$U_t^{(s,f)} = P_{d_s}\left(x_{t-K+1:t}^{(s,f)} \right) + e_{\text{coord}} + e_{d_s} + e_{\iota_f} + e_{\tau_f}. \tag{2}$$

Here, K is context length, $C_t^{(s)}$ its window, P_{d_s} the tokenizer, and $U_t^{(s,f)}$ its tokens; the four e terms encode coordinates, dimension, identity, and type in subscript order. The field tokens pass through a known type-specific interface,

$$V_t^{(s,f)} = E_{\tau_f}\left(U_t^{(s,f)} \right), \qquad Z_t^{(s)} = T_\theta\left(\mathcal{A}_{\text{cross}}\left(\{V_t^{(s,f)}\}_{f \in \mathcal{F}_s} \right) \right) \in \mathcal{Z}. \tag{3}$$

Here, E_{τ_f} is the type interface, $V_t^{(s,f)}$ typed tokens, $\mathcal{A}_{\text{cross}}$ aggregation, T_θ the shared Transformer, θ its parameters, and $Z_t^{(s)} \in \mathcal{Z}$ the latent. Field semantics determine the interface, while cross-field interaction and the shared Transformer provide one parameterized core in which the coupled state is represented and evolved. We pretrain a separate checkpoint for each benchmark family; within each checkpoint, the Transformer and predictive space are shared across all fields.

3.2 JOINT-EMBEDDING PREDICTION

UNIJEPA uses an online context encoder f_θ, a momentum target encoder f_ξ, and a horizon-conditioned predictor g_ϕ:

$$Z_t^{(s)} = f_\theta(C_t^{(s)}), \tag{4}$$

$$Z_{t+h}^{(s)} = f_\xi(X_{t+h}^{(s)}), \tag{5}$$

$$\widehat{Z}_{t+h}^{(s)} = g_\phi\Big(Z_t^{(s)}, e_h, e_{\text{param}}\Big). \tag{6}$$

Here, θ, ξ, ϕ parameterize the online encoder, target encoder, and predictor; h is horizon, e_h its embedding, and e_{param} the parameter embedding. $Z_{t+h}^{(s)}$ is the target latent and $\widehat{Z}_{t+h}^{(s)}$ its prediction. The stop-gradient target follows $\xi \leftarrow \rho\xi + (1-\rho)\theta$ with momentum $\rho \in [0, 1)$. The objective over streams r is

$$\mathcal{L}_{\text{JEPA}} = \sum_r \lambda_r \frac{1}{|\mathcal{M}_r|} \sum_{i \in \mathcal{M}_r} \Big\| \text{LN}(\widehat{Z}_{t+h,i}^{(s,r)}) - \text{LN}(\text{sg}[Z_{t+h,i}^{(s,r)}]) \Big\|_2^2. \tag{7}$$

Here, λ_r weights stream r, $\mathcal{M}_r$ is its matched-token set, and i a token index; LN, sg, $|\cdot|$, and $\|\cdot\|_2^2$ are layer normalization, stop-gradient, set size, and squared Euclidean distance. Retaining field-level targets prevents a low-dimensional but dynamically important variable from disappearing inside one global average.

3.3 PHYSICS AS A LEARNING SIGNAL

Small field decoders q_f expose predicted physical quantities during pretraining,

$$\widehat{x}_{t+h}^{(s,f)} = q_f\Big(\widehat{Z}_{t+h}^{(s,f)}\Big), \qquad \mathcal{L}_{\text{field}} = \sum_{f \in \mathcal{F}_s} \alpha_f \, \ell_f(\widehat{x}_{t+h}^{(s,f)}, x_{t+h}^{(s,f)}). \tag{8}$$

Here, q_f decodes field f, $\widehat{x}_{t+h}^{(s,f)}$ is its prediction, ℓ_f its loss, and $\alpha_f \geq 0$ its weight. When a governing or conservation relation is available, decoded fields define a system-specific structural loss

$$\mathcal{L}_{\text{phys}}^{(s)} = \Big\| \mathcal{N}_s\Big(\widehat{X}_{t+h}^{(s)}; C_t^{(s)}, \mu_s\Big) \Big\|_2^2. \tag{9}$$

Here, $\widehat{X}_{t+h}^{(s)}$ collects decoded fields, $\mathcal{N}_s$ is the physical residual operator, and μ_s known parameters. For coupled flow, structural signals include continuity and derived velocity quantities,

$$\partial_t \rho + \nabla\cdot(\rho\mathbf{v}) = 0, \quad \nabla\cdot\mathbf{v} = \partial_x V_x + \partial_y V_y, \quad \omega = \partial_x V_y - \partial_y V_x. \tag{10}$$

Here, ρ is density, $\mathbf{v} = (V_x, V_y)$ velocity, $\nabla\cdot$ divergence, $\partial_t, \partial_x, \partial_y$ derivatives, and ω vorticity. For a planar rotation g with matrix R_g, scalar, vector, and tensor fields transform as

$$(g\cdot a)(x) = a(R_g^{-1}x), \tag{11}$$

$$(g\cdot\mathbf{v})(x) = R_g\mathbf{v}(R_g^{-1}x), \tag{12}$$

$$(g\cdot Q)(x) = R_g Q(R_g^{-1}x)R_g^\top. \tag{13}$$

Here, x is position; $a, \mathbf{v}, Q$ are scalar, vector, and rank-two tensor fields; g is the action; and $^{-1}, ^\top$ are inverse and transpose. We supervise transformation consistency after applying the correct joint spatial and component action. For Active Matter, the symmetric orientation tensor is stored through $c = \text{tr}(D)$ and spin-2 components $q_1 = D_{xx} - D_{yy}$, $q_2 = D_{xy} + D_{yx}$, and reconstructed as

$$D = \frac{1}{2}\begin{bmatrix} c + q_1 & q_2 \\ q_2 & c - q_1 \end{bmatrix}. \tag{14}$$

Here, D is the orientation tensor, tr the trace operator, D_{ij} its entries, and c, q_1, q_2 the stored trace and spin-two components. The complete pretraining objective is

$$\mathcal{L}_{\text{pre}}^{(s)} = \mathcal{L}_{\text{JEPA}} + \lambda_{\text{field}}\mathcal{L}_{\text{field}} + \lambda_{\text{phys}}^{(s)}\mathcal{L}_{\text{phys}}^{(s)} + \lambda_{\text{eq}}^{(s)}\mathcal{L}_{\text{eq}}^{(s)} + \lambda_{\text{real}}^{(s)}\mathcal{L}_{\text{real}}^{(s)}. \tag{15}$$

Here, $\mathcal{L}_{\text{eq}}^{(s)}$ and $\mathcal{L}_{\text{real}}^{(s)}$ are consistency and realizability losses; each $\lambda \geq 0$ weights its subscripted loss. An inapplicable term receives zero weight, allowing the shared predictive parameterization to respect system-specific equations.

3.4 Transfer to a Neural Operator

After pretraining, the auxiliary physics decoders are removed. Predicted field streams are aligned with a dimension-appropriate FNO and injected through bounded residual adapters,

$$H_{\ell+1} = B_\ell(H_\ell) + \sum_{f \in \mathcal{F}_s} \sigma(\gamma_{\ell,f}) A_{\ell,f}\left(\widehat{Z}^{(f)}_{t+h}\right). \tag{16}$$

Here, ℓ indexes FNO blocks, H_ℓ hidden states, B_ℓ the native block, and $f \in \mathcal{F}_s$ a field. Alignment $A_{\ell,f}$ maps predicted stream $\widehat{Z}^{(f)}_{t+h}$ (system index suppressed), scaled by sigmoid σ of trainable logit $\gamma_{\ell,f}$. The raw state remains the primary FNO input. Pretraining augments the task-specific operator with a learned predictive representation while retaining direct observation.

4 Experiments

4.1 Systems and Protocol

Table 1: Few-shot forecasting with 16 labeled trajectories. Means over three seeds for scalar/CFD2D and five seeds for Active Matter. Large Scratch matches the transfer model's parameter count. Gains use Scratch FNO for scalar/CFD2D and Large Scratch for Active Matter.

Scalar fields 1D–3D benchmarks, three seeds				
Metric ↓	Scratch FNO	JEPA	UniJEPA	Gain ↑
Reaction–Diffusion1D	0.0978	0.1025	**0.0939**	4.0%
Diffusion–Reaction2D	0.2356	**0.2331**	0.2332	1.0%
Heat3D	0.2043	0.1772	**0.1768**	13.5%
CFD2D locked three-seed comparison				
Metric ↓	Scratch FNO	JEPA	UniJEPA	Gain ↑
Raw full-state rel-L2	0.3555	0.3283	**0.3269**	8.0%
Velocity rel-L2	0.8100	0.7869	**0.7868**	2.9%
Direction error	0.5815	0.5549	**0.5513**	5.2%
Kinetic-energy rel-L2	0.7562	0.7499	**0.7446**	1.5%
Continuity residual MSE	74.433	68.491	**68.097**	8.5%
Active Matter five-seed capacity-matched comparison				
Metric ↓	Ordinary Scratch	Large Scratch	UniJEPA	Gain ↑
Normalized full-state VRMSE	0.1510	0.1448	**0.1301**	10.2%
Raw full-state rel-L2	0.1129	0.1083	**0.0970**	10.4%
Velocity rel-L2	0.0992	0.0991	**0.0789**	20.4%
Orientation tensor rel-L2	0.0869	0.0814	**0.0740**	9.1%
Strain tensor rel-L2	0.2017	0.1931	**0.1760**	8.9%
Tensor C_4 error	0.1432	0.1215	**0.1115**	8.2%
Full-field C_4 error	0.1093	0.0932	**0.0839**	10.0%
Rotated-OOD full-state rel-L2	0.1186	0.1123	**0.0988**	12.0%
Strain-consistency rel-L2	0.2367	0.2251	**0.1568**	30.3%

The experiments follow the central claim as a sequence of tests: breadth across dimensions and field types, benefit of predictive pretraining, controls for capacity and exact equivariance, and direct evidence that physical structure is already present in the latent space.

We evaluate scalar PDEs in 1D–3D, coupled CFD2D flow (Takamoto et al., 2022), and scalar–vector–tensor Active Matter (Ohana et al., 2024), using 16 labeled trajectories. Each comparison

fixes the data splits and training budget and selects checkpoints on validation data. Table 1 reports three-seed scalar/CFD2D and five-seed Active Matter results; protocols and complete results are in Appendices C.1–E.3. All reported errors are lower-is-better.

4.2 Forecasting Across Physical Regimes

On scalar systems, UniJEPA reduces Heat3D error by 13.5%; gains vary across the nine benchmarks (Appendix E.1). On CFD2D, JEPA and UniJEPA reduce raw-state error by 7.66% and 8.0%, respectively. Both improve velocity error over Scratch in all five seeds of the extended comparison (Appendix E.2).

On Active Matter, Large Scratch matches the transfer model's parameter count within 0.58%. UniJEPA still lowers VRMSE by 10.06% and wins all five paired seeds, with gains in field-wise, physical-consistency, and rotated-input metrics (Table 1). Thus, increased capacity alone does not explain the improvement. Figure 3 shows some predictions; confidence intervals and additional field maps are in Appendices E.3 and E.3.2.

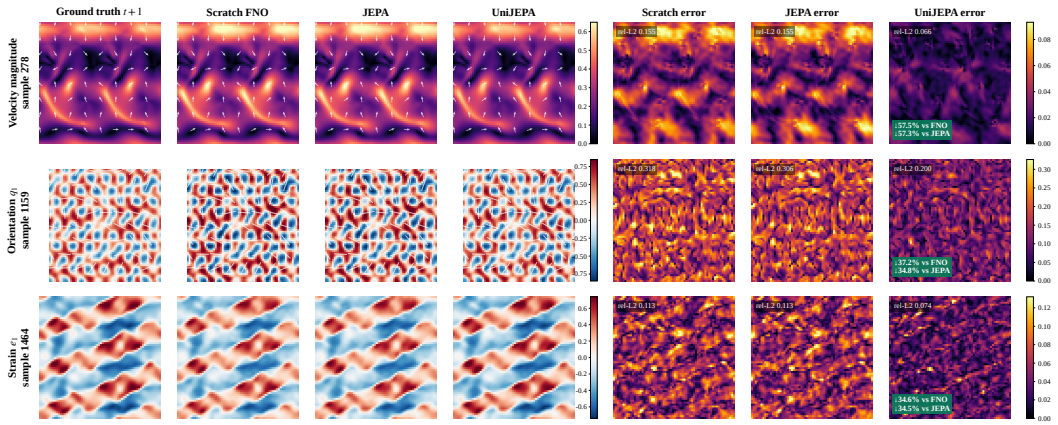

Figure 3: Held-out Active Matter predictions and error maps for velocity, orientation q_1, and strain e_1. Relative L2 reductions over Scratch FNO are 57.5%, 37.2%, and 34.6%, respectively.

4.3 Transfer Across Operator Backbones

Across MWT, CNO, PDE-Transformer, and Transolver, adding UniJEPA and downstream physics reduces VRMSE by 2.64–8.93% and strain-consistency error by 18.11–63.94% (Table 2). These matched comparisons evaluate the combined transfer-and-physics package; the controls below isolate representation learning. Appendix G gives additional results.

4.4 Controlled Attribution of the Gain

Exact-equivariance control. $p4$-GFNO, parameter-matched to Ordinary Scratch, reduces C_4 error to numerical precision, but UniJEPA achieves lower VRMSE (0.1215 vs. 0.1324) and strain-consistency error (0.1812 vs. 0.4823). Exact symmetry therefore does not replace predictive pretraining (Appendix E.3.1). The 0.1215 VRMSE belongs to the three-seed control protocol; the five-seed capacity comparison in Table 1 reports 0.1301.

Same-architecture control. Holding the complete architecture and optimization fixed, physics-guided pretraining lowers Active Matter state error from 0.1430 to 0.1215. Downstream physical losses are zero in this control.

Table 2: Cross-backbone transfer (16 trajectories; three seeds). Base+UNIJEPA includes downstream physics. Superscripts: sample SD; orange: error reduction before rounding. Errors ↓; reductions ↑. Strain consistency is a rel-L2 error.

	MWT			CNO		
Metric ↓	Base	Base+UNIJEPA	Gain ↑	Base	Base+UNIJEPA	Gain ↑
State VRMSE	$0.1320^{\pm 0.004}$	$\mathbf{0.1237}^{\pm 0.003}$	6.31%	$0.1265^{\pm 0.003}$	$\mathbf{0.1215}^{\pm 0.004}$	4.00%
Velocity rel-L2	$0.0858^{\pm 0.006}$	$\mathbf{0.0751}^{\pm 0.004}$	12.46%	$0.1019^{\pm 0.002}$	$\mathbf{0.0920}^{\pm 0.003}$	9.71%
Full-field C_4	$0.0957^{\pm 0.001}$	$\mathbf{0.0862}^{\pm 0.002}$	9.96%	$0.0951^{\pm 0.002}$	$\mathbf{0.0881}^{\pm 0.001}$	7.27%
Strain cons.	$0.4208^{\pm 0.029}$	$\mathbf{0.1517}^{\pm 0.031}$	63.94%	$0.7051^{\pm 0.057}$	$\mathbf{0.4878}^{\pm 0.021}$	30.82%
	PDE-Transformer			**Transolver**		
Metric ↓	Base	Base+UNIJEPA	Gain ↑	Base	Base+UNIJEPA	Gain ↑
State VRMSE	$0.1166^{\pm 0.004}$	$\mathbf{0.1135}^{\pm 0.002}$	2.64%	$0.1494^{\pm 0.018}$	$\mathbf{0.1361}^{\pm 0.001}$	8.93%
Velocity rel-L2	$0.0907^{\pm 0.001}$	$\mathbf{0.0868}^{\pm 0.001}$	4.25%	$0.1141^{\pm 0.005}$	$\mathbf{0.1090}^{\pm 0.002}$	4.51%
Full-field C_4	$0.0698^{\pm 0.002}$	$\mathbf{0.0671}^{\pm 0.002}$	3.86%	$0.0765^{\pm 0.005}$	$\mathbf{0.0707}^{\pm 0.010}$	7.56%
Strain cons.	$0.3918^{\pm 0.055}$	$\mathbf{0.2381}^{\pm 0.021}$	39.23%	$0.3091^{\pm 0.044}$	$\mathbf{0.2531}^{\pm 0.050}$	18.11%

Table 3: Same-architecture control: three-seed mean with sample SD in superscript. Downstream physical and transformation losses are zero.

Initialization	CFD state ↓	CFD velocity ↓	Active state ↓	Active tensor ↓
Random	$0.4807^{\pm 0.0180}$	$0.8057^{\pm 0.0320}$	$0.1430^{\pm 0.0080}$	$0.0808^{\pm 0.0040}$
JEPA	$0.4719^{\pm 0.0160}$	$0.7868^{\pm 0.0350}$	$0.1430^{\pm 0.0080}$	$0.0807^{\pm 0.0040}$
UNIJEPA	$\mathbf{0.4697}^{\pm 0.0190}$	$\mathbf{0.7868}^{\pm 0.0330}$	$\mathbf{0.1215}^{\pm 0.0020}$	$\mathbf{0.0685}^{\pm 0.0020}$

Training-signal attribution. Table 4 varies only pretraining supervision; architecture, data, and optimization are fixed, with downstream physical losses disabled.

Table 4: Pretraining-signal ablation: three-seed mean with sample SD in superscript. Aux.: decoded field supervision; rotation: transformation consistency.

Training signal	Full state ↓	Orientation ↓	Tensor C_4 ↓	Strain cons. ↓
JEPA only	$0.1430^{\pm 0.0080}$	$0.0807^{\pm 0.0040}$	$0.1385^{\pm 0.0070}$	$0.2035^{\pm 0.0670}$
JEPA + aux.	$0.1427^{\pm 0.0080}$	$0.0806^{\pm 0.0040}$	$0.1381^{\pm 0.0070}$	$0.1992^{\pm 0.0600}$
JEPA + aux. + physics	$0.1405^{\pm 0.0080}$	$0.0803^{\pm 0.0040}$	$0.1351^{\pm 0.0070}$	$\mathbf{0.1564}^{\pm 0.0590}$
JEPA + aux. + rotation	$0.1306^{\pm 0.0060}$	$0.0736^{\pm 0.0020}$	$0.1173^{\pm 0.0060}$	$0.1899^{\pm 0.0580}$
Full UNIJEPA	$\mathbf{0.1215}^{\pm 0.0020}$	$\mathbf{0.0685}^{\pm 0.0020}$	$\mathbf{0.1069}^{\pm 0.0030}$	$0.1812^{\pm 0.0200}$

Auxiliary decoding alone changes little. Physics chiefly improves strain consistency; rotation improves field geometry; combining them gives the lowest state and tensor errors. Appendix F.2 also shows more uniform physical-loss gradients at matched pretraining epoch 30.

4.5 REPRESENTATION-LEVEL EVIDENCE

Frozen latent probing. We freeze the encoder and predictor and fit local linear ridge or two-layer MLP probes. Table 5 reports future orientation and strain readout errors, without a raw-state bypass or spatial mixing.

With 1,285 fitted parameters across the three probe tasks, the Linear probe lowers mean orientation and strain readout error from 0.2645 (JEPA) to 0.1543. Both readouts improve these two quantities over JEPA and Random (Table 5), demonstrating physical information in the frozen representation.

Table 5: Frozen orientation and strain probes: test readout errors (↓). Linear probes use ridge regression; MLP results are means over three readout-training seeds. Orange: error reduction over Random (↑). Details are in Appendix F.3.

Method	Lin. Ori.↓	Lin. Str.↓	MLP Ori.↓	MLP Str.↓
Random	0.2239	0.2843	0.1556	0.2334
JEPA	0.2391	0.2898	0.2235	0.2803
UNIJEPA	**0.1312**	**0.1774**	**0.1078**	**0.1592**
Gain ↑	**41.4%**	**37.6%**	**30.7%**	**31.8%**

The physical atlas shows more coherent physical neighborhoods (Figure 4). UMAP rows are projected independently; comparisons use within-map structure and the reported scores.

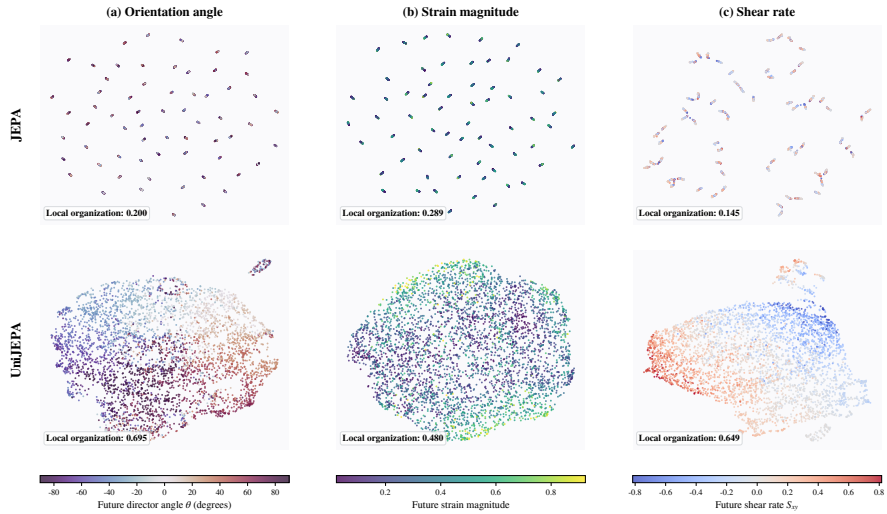

Figure 4: Frozen-latent physical atlas. Each point is a future patch colored by its ground-truth orientation, strain, or shear rate. Higher local-organization scores (↑) indicate more coherent physical neighborhoods; UNIJEPA improves all three.

Typed-interface ablation. With raw channels, architecture, and pretraining signals matched, field typing lowers state error from 0.1420 to 0.1215 and tensor C_4 error from 0.1335 to 0.1069 (Appendix F.4).

5 CONCLUSION

UNIJEPA organizes scalar, vector, and tensor fields in a shared predictive space before forecasting. Gains across physical regimes and operator backbones, together with capacity controls, signal ablations, and frozen probes, support physically organized pretraining as a reusable foundation for few-shot scientific prediction.

AI USE STATEMENT

Generative AI tools assisted with experimental planning, code development, literature discovery, and drafting and editing portions of the manuscript. The authors reviewed and revised the AI-assisted content, executed all experiments, verified the reported results and mathematical claims, and take full responsibility for the final paper.

REFERENCES

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, June 2023.

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.

Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K. Gupta. Clifford neural layers for PDE modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Francisco Giral, Abhijeet Vishwasrao, Andrea Arroyo Ramo, Mahmoud Golestanian, Federica Tonti, Adrian Lozano-Duran, Steven L. Brunton, Sergio Hoyas, Hector Gomez, Soledad Le Clainche, and Ricardo Vinuesa. Aerojepa: Learning semantic latent representations for scalable 3d aerodynamic field modeling, 2026.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24048–24062. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c9e5c2b59d98488fe1070e744041ea0e-Paper.pdf.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635. PMLR, 21–27 Jul 2024.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant Fourier neural operators for partial differential equations. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12907–12930. PMLR, 23–29 Jul 2023.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624. Curran Associates, Inc., 2024. doi: 10.52202/079017-2311.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-transformer: Efficient and versatile transformers for physics simulations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23562–23602. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/holzschuh25a.html.

Haozhe Jia, Pengyu Yin, Wenshuo Chen, Shaofeng Liang, Lei Wang, Bowen Tian, Xiucheng Wang, Nanqian Jia, and Yutao Yue. Learning to think in physics: Breaking shortcut learning in scientific diffusion via representation alignment, 2026.

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

Zongyi Li, Nikola Borislavov 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.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, 2021. doi: 10.1038/S42256-021-00302-5.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 119301–119335. Curran Associates, Inc., 2024. doi: 10.52202/079017-3791.

Weizhi Nie, Weichao Liu, Honglin Guo, and Yuting Su. Phys-jepa: Physics-informed latent world models for multivariate time-series forecasting, 2026.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, 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 Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037. Curran Associates, Inc., 2024. doi: 10.52202/079017-1430.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021.

Helen Qu, Rudy Morel, Michael McCabe, Alberto Bietti, François Lanusse, Shirley Ho, and Yann LeCun. Representation learning for spatiotemporal physical systems, 2026.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200. Curran Associates, Inc., 2023. doi: 10.52202/075280-3376. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f3c1951b34f7f55ffaecada7fde6bd5a-Paper-Conference.pdf.

Mahindra Singh Rautela, Alexander Most, Siddharth Mansingh, Bradley C. Love, Alexander Scheinker, Diane Oyen, Nathan Debardeleben, Earl Lawrence, and Ayan Biswas. Morph: Pde foundation models with arbitrary data modality, 2026.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 13–18 Jul 2020.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) Equivariant Graph Neural Networks. In *International Conference on Machine Learning*, volume 139, pp. 9323–9332, 2021.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611. Curran Associates, Inc., 2022. doi: 10.52202/068431-0117.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Anthony Zhou and Amir Barati Farimani. Masked autoencoders are PDE learners. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

A Field Actions and Tensor Parameterization

This section fixes the geometric conventions used by the Active Matter loader, transformation objectives, and tensor-valued metrics. The spatial grid and field components must be transformed jointly; changing the packing convention or rotating components alone defines a different task.

Scalar, vector, and tensor actions. Let

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{17}$$

denote a counterclockwise planar rotation through angle θ. A transformed field is evaluated by inverse-warping the coordinates and then applying the representation carried by its physical type:

$$(T_\theta a)(x) = a(R_\theta^{-1}x), \tag{18}$$

$$(T_\theta \mathbf{v})(x) = R_\theta \mathbf{v}(R_\theta^{-1}x), \tag{19}$$

$$(T_\theta Q)(x) = R_\theta Q(R_\theta^{-1}x)R_\theta^\top. \tag{20}$$

Here, x is a spatial coordinate, T_θ is the field action induced by R_θ, and a, $\mathbf{v}$, and Q are scalar, vector, and rank-two tensor fields, respectively; $^{-1}$ and $^\top$ denote matrix inverse and transpose. The action is joint: the spatial grid and the components are transformed together. Scalars are resampled but not component-mixed; vectors use the standard two-dimensional representation; rank-two tensors transform on both indices.

The discrete transformation group used in the reported Active Matter experiment is

$$C_4 = \{R_0, R_{\pi/2}, R_\pi, R_{3\pi/2}\}. \tag{21}$$

Here, C_4 is the four-element group of quarter-turn rotations. Because these rotations map the square periodic grid to itself, they introduce no interpolation ambiguity. Exact architectural equivariance is not imposed; consistency is learned through the normalized objective in Appendix D.

Symmetric tensor packing. For a symmetric planar tensor T, the stored channels use the full-difference convention

$$c = T_{xx} + T_{yy}, \qquad q_1 = T_{xx} - T_{yy}, \qquad q_2 = T_{xy} + T_{yx} = 2T_{xy}. \tag{22}$$

The inverse map is

$$T = \frac{1}{2}\begin{bmatrix} c + q_1 & q_2 \\ q_2 & c - q_1 \end{bmatrix}. \tag{23}$$

Under rotation by θ, the trace c is invariant and the pair (q_1, q_2) transforms as a spin-two vector:

$$\begin{bmatrix} q_1' \\ q_2' \end{bmatrix} = \begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \tag{24}$$

The primes denote the packed components after rotation by θ. This identity is used as an implementation check in addition to the full matrix action.

Orientation and strain tensors. The orientation second moment D is reconstructed from concentration c and the two orientational channels. Its trace is constrained by construction, $\mathrm{tr}(D) = c$. Realizability is evaluated through the eigenvalues $\lambda_i(D)$:

$$\mathcal{L}_{\mathrm{real}} = \frac{1}{N}\sum_{n,i}\left[\max(0, -\lambda_i(D_n))\right]^2. \tag{25}$$

Here, N is the number of evaluated tensors, n indexes a tensor instance, i indexes its eigenvalues, and $\lambda_i(D_n)$ is the ith eigenvalue of D_n. The archived evaluator reports both the fraction of negative eigenvalues and their mean negative magnitude.

For velocity $\mathbf{u} = (u_x, u_y)$, the stored strain target is compared to the symmetric trace-free velocity gradient

$$S(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\top\right) - \frac{1}{2}(\nabla\cdot\mathbf{u})I. \tag{26}$$

Here, $S(\mathbf{u})$ is the symmetric trace-free strain tensor, $\nabla\mathbf{u}$ is the velocity-gradient matrix, $\nabla\cdot\mathbf{u}$ is its divergence, and I is the 2×2 identity matrix. The strain-consistency metric therefore tests a differential relationship between two predicted streams rather than only comparing one output channel to its target.

B STRUCTURAL OBJECTIVES BY PHYSICAL REGIME

The model shares a predictive parameterization, not a single equation. For system s, an unavailable structural term is assigned zero weight. Table 6 records the implemented supervision map.

Table 6: Structural quantities available in each regime. "Field" denotes decoded-field supervision and "transform" denotes the correct action for each geometric type.

Regime	Field	Equation/constraint	Derived observables	Transform
Scalar PDEs	yes	PDE residual	conservation diagnostics	scalar
CFD2D	yes	continuity	div., vort., energy	scalar/vector
Active Matter	yes	tensor/strain structure	PSD, strain consistency	scalar/vector/tensor

Scalar systems. For a scalar system

$$\partial_t u = G_s(u, \nabla u, \nabla^2 u; \mu_s), \tag{27}$$

Here, u is the scalar state, G_s is the evolution operator of system s, ∇u and $\nabla^2 u$ are its spatial gradient and Laplacian, and μ_s collects known physical parameters. The discrete residual has the general form

$$r_s = D_t \widehat{u} - G_s(\widehat{u}, D_x \widehat{u}, D_{xx} \widehat{u}; \mu_s). \tag{28}$$

Here, r_s is the residual, $\widehat{u}$ is the decoded prediction, and D_t, D_x, and D_{xx} are the discrete temporal, first-spatial, and second-spatial derivative operators. For the three one-dimensional PDEBench systems this becomes

$$R_{\mathrm{adv}} = u_t + \beta u_x, \tag{29}$$

$$R_{\mathrm{burg}} = u_t + u u_x - \nu u_{xx}, \tag{30}$$

$$R_{\mathrm{rd}} = u_t - \nu u_{xx} - \rho u(1-u). \tag{31}$$

Here, R_{adv}, R_{burg}, and R_{rd} are the advection, Burgers, and reaction–diffusion residuals; subscripts on u denote partial derivatives, β is advection speed, ν is diffusivity or viscosity, and ρ is the reaction-rate coefficient in this scalar-system context. Finite differences are applied in physical coordinates after undoing normalization. Residual magnitudes are not used to rank different systems because the discretizations, time steps, and coefficient scales differ.

Coupled flow. The CFD state is $\Phi = (\rho, p, \mathbf{v})$, where Φ is the full state, ρ is density, p is pressure, and $\mathbf{v} = (V_x, V_y)$ is planar velocity. The continuity relation is

$$R_{\mathrm{cont}} = \partial_t \rho + \nabla \cdot (\rho \mathbf{v}), \tag{32}$$

where R_{cont} is the continuity residual. The separately evaluated derived fields are

$$\nabla \cdot \mathbf{v} = \partial_x V_x + \partial_y V_y, \qquad \omega = \partial_x V_y - \partial_y V_x, \qquad k = \tfrac{1}{2}\|\mathbf{v}\|_2^2. \tag{33}$$

Here, ω is scalar vorticity and k is kinetic-energy density per unit mass. Reporting state, velocity, direction, divergence, vorticity, and kinetic-energy errors prevents a low scalar-state error from hiding poor vector dynamics.

Active Matter. The seven-channel downsampled state contains one concentration channel, two velocity channels, two orientation channels, and two strain channels. During representation pretraining, the complete protocol combines predictive feature matching, decoded-field supervision, tensor/strain structure, and C_4 consistency. The released state does not contain the fourth-moment closure needed to evaluate the full kinetic-theory residual. We therefore report only quantities supported by the released fields: incompressibility-related diagnostics, strain consistency, tensor realizability, and discrete rotation consistency.

C Experimental Details

Following the task-wise presentation used in the main paper, we first specify data construction and split discipline, then summarize model and optimization settings. Complete numerical results and controlled studies are reported in Appendices E and F.

C.1 Data, Preprocessing, and Split Discipline

Scalar suite. The scalar comparison uses nine tasks spanning native one-, two-, and three-dimensional grids. Every system uses 512 training, 128 validation, and 256 test trajectories, including the five controlled 3D systems in the fixed Scalar16 split. A budget of 16 selects complete training trajectories; it does not count cropped windows as independent labels. Downstream seeds are 0, 1, and 2.

Table 7: Scalar suite used by the canonical three-seed aggregate. The five 3D rows belong to the fixed Scalar16 suite stored under the project PDEBench data tree.

System	Dim.	Principal structure	Downstream epochs
Advection	1D	transport	200
Burgers	1D	nonlinear transport/diffusion	200
Reaction–diffusion	1D	diffusion/reaction	200
Diffusion–reaction	2D	coupled spatial operators	200
Heat	3D	diffusion	100
Advection–diffusion	3D	transport/diffusion	100
Allen–Cahn	3D	phase field	100
Wave	3D	propagation	100
Cahn–Hilliard	3D	phase separation	100

Joint scalar pretraining runs for 30 epochs. All three methods use the same budget and split for a given downstream seed. Checkpoints are selected on validation loss before the test split is opened.

CFD2D. The loaded tensor has shape $(N, 21, 32, 32, 4)$ with channel order $[\rho, p, V_x, V_y]$. The fixed split contains 512 training, 128 validation, and 256 test trajectories. Five context frames predict the following 16 frames directly. The main table is the locked three-seed comparison; a five-seed extension is used only for the velocity reliability statement.

Active Matter. The Well Active Matter data contain 225 trajectories on a periodic 256×256 domain with 81 time steps. Fourier low-pass downsampling produces a 64×64 cache while preserving periodicity. Four context frames predict one next frame; longer horizons are obtained autoregressively. At budget 16, the archived split contains 1,232 training windows. Validation uses 24 trajectories and 1,848 windows. Test metrics use 2,002 one-step windows, while rollout metrics use 78 starting sequences.

Table 8: Representation-pretraining pools and downstream labeled budgets. A trajectory is selected before any temporal examples are formed, so multiple windows from one trajectory are not treated as independent labels.

System family	Representation-pretraining pool	Downstream labeled pool
Scalar suite	512 trajectories per system; 4,608 trajectory items across nine systems	16 complete trajectories per system
CFD2D	512 trajectories; 512 fixed clips with 5 context and 16 target frames	16 complete trajectories
Active Matter	175 trajectories; 13,475 sliding windows with 4 context frames and 1 target frame	16 trajectories; 1,232 windows

The budget trajectory list is fixed across methods within each comparison. Model seed changes initialization and minibatch order, not the labeled subset. The principal five-seed capacity experiment uses downstream seeds 0–4. The matched control and representation analyses use seeds 0–2.

Checkpoint selection and test access. Each seed is trained independently. The Active Matter selection score is

$$S_{\mathrm{val}} = \tfrac{1}{2} E^{\mathrm{val}}_{\mathrm{state}} + \tfrac{1}{2} E^{\mathrm{val}}_{\mathrm{orientation}}. \tag{34}$$

Here, S_{val} is the checkpoint-selection score, while $E^{\mathrm{val}}_{\mathrm{state}}$ and $E^{\mathrm{val}}_{\mathrm{orientation}}$ are the validation full-state and orientation errors. The selected epoch minimizes this validation score. Stored training summaries record `test_accessed=false`; the final evaluator sets `allow_test=true` only after checkpoint selection. No test metric is used to choose an epoch, weight, method variant, or random seed.

Table 9 summarizes which random-seed set supports each appendix result. Seed counts refer to independently trained downstream models; examples or rollout starts within one seed are not treated as independent replicates.

Table 9: Evaluation protocols used by the reported aggregates. "Val." indicates validation-only checkpoint or hyperparameter selection before one locked test pass.

Study	Seeds	Primary evaluation unit	Selection
Scalar transfer	0–2	test trajectory	Val.
CFD2D main comparison	0–2	test trajectory	Val.
CFD2D velocity extension	0–4	test trajectory	Val.
Active Matter capacity control	0–4	2,002 windows / 78 rollouts	Val.
Active Matter signal ablation	0–2	test window	Val.
Frozen latent probes	MLP: 0–2	non-overlapping patch target	Val.
Channel-blind comparison	0–2	test window	Val.
C_4-regularized Scratch	0–4	test window / rollout	Val. weight

Execution order. Representation pretraining, downstream fitting, checkpoint selection, and final testing are separate phases. Pretraining learns the online encoder and predictor against an exponential-moving-average target encoder. Downstream fitting then initializes the task predictor from the selected representation checkpoint, retains the raw physical state as the direct FNO input, and injects representation streams only through learned adapters. Validation chooses one checkpoint per seed. The final evaluator reloads that checkpoint, enables test access once, and writes immutable per-seed metrics. This phase separation prevents a favorable test horizon, seed, or auxiliary weight from being selected after test inspection.

C.2 Architectures, Optimization, and Parameters

Table 10: Principal model and optimization settings.

Setting	Scalar	CFD2D	Active Matter
Temporal input and target			
Context frames	task-specific	5	4
Target frames	task-specific	16 direct	1
Predictive representation			
JEPA latent width	shared	256	256
Encoder / predictor layers	shared	4 / 2	4 / 2
Attention heads	shared	8	8
Target EMA	shared	0.996	0.996
Downstream operator			
FNO width	task-specific	128	64
Fourier modes	task-specific	12	16
FNO depth	task-specific	4	4
Optimization			
Downstream epochs	200 or 100	300	80
Batch size	task-specific	task-specific	4
Learning rate	task-specific	task-specific	2×10^{-4}
Weight decay	task-specific	task-specific	10^{-4}
Gradient clip	enabled	enabled	1.0

Table 11: Unified representation stack. P: pretraining; T: downstream transfer. "Included denotes parameters already counted in the online stack.

Component	Structure	Parameters	Stage
Tokenizer P_{d_s}	Scalar: patch projection; 2D roles: 3×3 + strided conv.	Included	P, T
Role experts	Four routes; two 256×256 maps, residual + LN.	530,432	P, T
Cross-role $\mathcal{A}_{\text{cross}}$	8-head attention + projection.	922,368	P, T
Encoder T_θ	4 Transformer layers; width 256; FFN 512; 8 heads.	2,108,416	P, T
Predictor G_ϕ	2 Transformer layers; width 256; final LN.	1,054,720	P, T
EMA encoder	Online-encoder copy; momentum 0.996.	16,121,600	P
Field decoders	LN + Linear + GELU + Linear.	2,143,360	P
Transfer adapters	20 maps $(256\to64)$ + 20 sigmoid gates.	328,960 + 20	T

The four expert routes are scalar, vector, tensor, and structure; structure processes derived divergence and vorticity. Active Matter retains five predictive streams: concentration, velocity, orientation, strain, and structure. Orientation and strain share the tensor expert while remaining separate streams.

The component rows are not intended to sum to the online encoder–predictor total: the archived 17,176,320 parameters also include native tokenizers, coordinate/type/system conditioning, and associated normalization parameters. During pretraining, the 17,176,320-parameter online encoder–predictor, the 16,121,600-parameter EMA target encoder, and 2,143,360 pretraining-only decoder parameters give a resident model footprint of 35,441,280 parameters. The target branch and field decoders are discarded before downstream transfer.

The Active Matter predictor uses 64 channels, 16 Fourier modes, four FNO blocks, cosine learning-rate decay to 10^{-6}, and validation at every epoch. Large Scratch changes only FNO width from 64 to 112. The reported complete transfer stack contains 25,922,267 trainable parameters; Ordinary Scratch contains 8,416,967. The width-112 control was chosen to match the complete stack within 0.58%.

Table 12: Archived Active Matter parameter accounting. "Rep." includes the predictive encoder and predictor; adapters align predictive streams with the FNO.

Configuration	Total / trainable	Base FNO	Rep.	Adapters
Forecasting models				
Ordinary Scratch	8,416,967	8,416,967	0	0
C_4-regularized Scratch	8,416,967	8,416,967	0	0
Complete UNIJEPA	25,922,267	8,416,967	17,176,320	328,960
Frozen representation and probe heads				
Frozen probe representation	17,250,048	0	17,250,048	0
Linear probe head	1,285	0	0	0
Two-layer MLP probe head	49,669	0	0	0

Bounded transfer. Each representation stream enters an FNO block through a learned alignment map multiplied by a sigmoid gate. The five streams enter each of four FNO blocks through separate adapters, giving 20 maps and 20 scalar gates. In the primary Active Matter transfer model, the online encoder and predictor are fine-tuned jointly with the adapters and FNO. The raw state remains the FNO input, so transfer supplements rather than replaces direct observation.

Hardware and software. The evidence archive was collected on a node with eight NVIDIA H100 80GB HBM3 GPUs. The recorded driver is 535.230.02 and the driver-reported CUDA compatibility version is 12.2. Runs use Python 3.10.20, PyTorch 2.5.1+cu121, NumPy 2.2.6, and cuDNN 9.1.0.70. Table 13 reports coarse archival intervals from resolved-configuration creation to the final stored artifact. They include initialization, validation, evaluation, and file I/O; because the node was shared and several GPUs were occupied concurrently, they are not isolated throughput benchmarks.

Table 13: Archived H100 training intervals and deterministic parameter footprint. Times are medians when five downstream seeds are available. Inference latency was not isolated in the archive, so the inference comparison is limited to exact resident parameter counts.

Operation	Archived scope	Recorded interval	Parameter footprint
Unified representation pretraining	30 epochs, 144 steps/epoch, batch 32	8.6 min, one run	35.44M resident
Active Matter Scratch fitting	80 epochs, seeds 0–4	15.8 min median	8.42M trainable
Active Matter UNIJEPA fitting	80 epochs, seeds 0–4	66.4 min median	25.92M trainable
Scratch inference footprint	Selected downstream model	Latency not isolated	8.42M resident
UNIJEPA inference footprint	Selected downstream model	Latency not isolated	25.92M resident (3.08×)

D Evaluation Metrics and Statistical Reporting

For prediction $\widehat{u}$ and target u, relative L_2 error is

$$E_{\mathrm{rel}} = \frac{\|\widehat{u} - u\|_2}{\|u\|_2 + \epsilon}. \tag{35}$$

Here, E_{rel} is relative error, $\|\cdot\|_2$ is the Euclidean norm over evaluated entries, and $\epsilon > 0$ is a fixed numerical stabilizer. Let K be the number of physical variables, N_k the number of evaluated entries for variable k, and (μ_k, σ_k) its training-set statistics. The normalized full-state VRMSE is

$$E_{\mathrm{VRMSE}} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N_k}\sum_{i=1}^{N_k}\left(\frac{\widehat{u}_{k,i} - u_{k,i}}{\sigma_k + \epsilon}\right)^2}. \tag{36}$$

Here, k indexes the K physical variables, i indexes their evaluated entries, $u_{k,i}$ and $\widehat{u}_{k,i}$ are target and predicted values, and σ_k is the training-set standard deviation of variable k; μ_k is used when undoing the stored normalization. The corresponding raw full-state error is evaluated after undoing normalization. Reporting both prevents low-variance fields from disappearing in a joint raw norm while also exposing whether normalized gains survive in physical units. Vector and tensor errors use their joint Euclidean or Frobenius norms after physical reconstruction, rather than averaging independent channel errors.

Velocity direction error is $1 - \cos(\widehat{\mathbf{v}}, \mathbf{v})$ averaged over valid sites, where $\widehat{\mathbf{v}}$ and $\mathbf{v}$ are predicted and target velocities and $\cos(\cdot,\cdot)$ denotes cosine similarity. High-frequency error applies the fixed Fourier mask before computing relative L_2. Continuity residual is a mean squared residual and therefore differs in scale from the relative-error rows.

For tensor-valued quantities, channel errors are never interpreted independently. Orientation and strain channels are first reconstructed into symmetric matrices using Appendix A; the relative error is then computed with the Frobenius norm. Strain consistency compares the predicted strain tensor with $S(\widehat{\mathbf{u}})$, so it measures agreement between two predicted physical streams. The 20-step Active Matter metric is an endpoint error after autoregressively feeding predictions back as inputs; it is therefore a stability diagnostic rather than another one-step accuracy measure.

The normalized transformation-consistency error is

$$E_{\mathrm{eq}} = \mathbb{E}_{X,g}\left[\frac{\|F(T_g X) - T_g F(X)\|_2}{\|T_g F(X)\|_2 + \epsilon}\right], \tag{37}$$

where X is an input context, F is the forecasting model, g is sampled from C_4, T_g is its action on fields, and $\mathbb{E}_{X,g}$ denotes expectation over inputs and rotations. Rotated-distribution forecasting evaluates accuracy against a rotated target:

$$E_{\mathrm{rot}} = \mathbb{E}_{X,Y,g}\left[\frac{\|F(T_g X) - T_g Y\|_2}{\|T_g Y\|_2 + \epsilon}\right]. \tag{38}$$

Here, Y is the unrotated future target and $\mathbb{E}_{X,Y,g}$ averages over input–target pairs and rotations.

For n independent downstream seeds, tables report the arithmetic mean and sample standard deviation,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{39}$$

Here, x_i is the metric from seed i, $\bar{x}$ is its arithmetic mean, and s is the sample standard deviation. Paired comparisons use $d_i = x_i^{(A)} - x_i^{(B)}$, where A and B denote the two compared methods and d_i is their paired difference for seed i. The five-seed confidence interval in the main text is computed from the empirical paired-difference distribution and is reported together with the paired mean; it is not an interval formed by subtracting two marginal intervals.

Relative reduction is

$$\mathrm{Reduction}(\%) = 100\frac{E_{\mathrm{baseline}} - E_{\mathrm{method}}}{E_{\mathrm{baseline}}}. \tag{40}$$

Here, E_{baseline} and E_{method} are the errors of the reference baseline and the evaluated method on the same metric. Prediction errors, physical residuals, and consistency errors are lower-is-better (↓); relative error reductions and latent local-organization scores are higher-is-better (↑). Raw physical field values have no preferred direction. Boldface marks the best displayed mean, not statistical significance.

E Additional Benchmark Results

This section reports the complete benchmark metrics underlying the condensed results in the main paper.

E.1 Scalar Systems

Table 14 exposes all three seed values behind the scalar means in the main text. The seed order is 0/1/2. This table also makes negative transfer visible: Scratch is best for Advection, ordinary JEPA is best for Burgers, and UniJEPA is not uniformly best on every system.

Table 14: Per-seed few-shot relative L_2 error (↓) for the canonical scalar run. Methods form column groups; each group reports seeds 0, 1, and 2 separately.

System	Scratch FNO ↓			JEPA ↓			UniJEPA ↓		
	0	1	2	0	1	2	0	1	2
Advection1D	0.1822	0.2547	0.2215	0.1891	0.2665	0.2086	0.1884	0.2670	0.2085
Burgers1D	0.3354	0.2733	0.3514	0.3176	0.2663	0.3441	0.3328	0.2647	0.3433
Reaction–Diffusion1D	0.1484	0.0969	0.0482	0.1291	0.0978	0.0807	0.1242	0.0962	0.0613
Diffusion–Reaction2D	0.2379	0.2415	0.2274	0.2349	0.2396	0.2250	0.2354	0.2394	0.2249
Heat3D	0.1912	0.2366	0.1851	0.1498	0.2274	0.1544	0.1496	0.2260	0.1548
Advection–Diffusion3D	0.6603	0.6273	0.6735	0.6610	0.6267	0.6675	0.6576	0.6215	0.6681
Allen–Cahn3D	0.1902	0.1864	0.1928	0.1819	0.1765	0.1894	0.1828	0.1763	0.1904
Wave3D	0.6715	0.6926	0.6084	0.6547	0.6904	0.6047	0.6591	0.6928	0.6066
Cahn–Hilliard3D	0.3322	0.3640	0.3745	0.3305	0.3542	0.3761	0.3312	0.3546	0.3786

Observations. Across the nine means, predictive pretraining improves on Scratch in eight systems. The differences are modest for Diffusion–Reaction2D and several 3D systems, while Heat3D shows the clearest transfer gain. The result supports breadth of transfer, not universal dominance of physics-guided pretraining over generic JEPA.

E.2 COUPLED FLOW

Table 15: All twelve locked CFD2D test metrics from the original three-seed comparison. Mean with sample standard deviation; the velocity row retains five decimals to distinguish close means.

Metric ↓	Scratch FNO	JEPA	UNIJEPA
State prediction			
Normalized full-state rel-L2	$0.4817^{\pm 0.0257}$	$0.4719^{\pm 0.0159}$	$\mathbf{0.4697}^{\pm 0.0188}$
Raw full-state rel-L2	$0.3555^{\pm 0.0288}$	$0.3283^{\pm 0.0298}$	$\mathbf{0.3269}^{\pm 0.0357}$
Scalar rel-L2	$0.3560^{\pm 0.0289}$	$0.3287^{\pm 0.0299}$	$\mathbf{0.3274}^{\pm 0.0358}$
Velocity diagnostics			
Velocity rel-L2	$0.8100^{\pm 0.0331}$	$0.7869^{\pm 0.0349}$	$\mathbf{0.7868}^{\pm 0.0333}$
Speed-magnitude rel-L2	$0.5405^{\pm 0.0130}$	$0.5258^{\pm 0.0154}$	$\mathbf{0.5219}^{\pm 0.0148}$
Velocity direction error	$0.5815^{\pm 0.0280}$	$0.5549^{\pm 0.0309}$	$\mathbf{0.5513}^{\pm 0.0293}$
Velocity high-frequency rel-L2	$0.9930^{\pm 0.0374}$	$\mathbf{0.9769}^{\pm 0.0277}$	$0.9798^{\pm 0.0335}$
Derived physical quantities			
Divergence rel-L2	$0.9011^{\pm 0.0234}$	$\mathbf{0.8908}^{\pm 0.0280}$	$0.8918^{\pm 0.0282}$
Vorticity rel-L2	$0.7771^{\pm 0.0221}$	$\mathbf{0.7647}^{\pm 0.0224}$	$0.7655^{\pm 0.0218}$
Kinetic-energy rel-L2	$0.7562^{\pm 0.0279}$	$0.7499^{\pm 0.0267}$	$\mathbf{0.7446}^{\pm 0.0305}$
Continuity residual MSE	$74.4331^{\pm 10.8267}$	$68.4906^{\pm 4.5714}$	$\mathbf{68.0973}^{\pm 5.5655}$
Long-horizon prediction			
Far-horizon velocity rel-L2	$0.8901^{\pm 0.0212}$	$\mathbf{0.8688}^{\pm 0.0245}$	$0.8702^{\pm 0.0279}$

Observations. Both predictive methods improve Scratch on all twelve displayed means. The strict ordering UNIJEPA $<$ JEPA $<$ Scratch holds for eight metrics. JEPA is slightly better for far-horizon velocity, high-frequency velocity, divergence, and vorticity. The additional five-seed velocity check gives $0.8013^{\pm 0.0277}$ for Scratch, $\mathbf{0.7810}^{\pm 0.0261}$ for JEPA, and $0.7816^{\pm 0.0246}$ for UNIJEPA; both predictive methods beat Scratch on all five paired seeds.

E.3 ACTIVE MATTER

Table 16: Five-seed Active Matter comparison used for the primary heterogeneous-field result.

Metric ↓	Ordinary Scratch	Large Scratch	UNIJEPA
State and field prediction			
Normalized full-state VRMSE	$0.1510^{\pm 0.0184}$	$0.1448^{\pm 0.0184}$	$\mathbf{0.1301}^{\pm 0.0154}$
Raw full-state rel-L2	$0.1129^{\pm 0.0134}$	$0.1083^{\pm 0.0135}$	$\mathbf{0.0970}^{\pm 0.0111}$
Velocity rel-L2	$0.0992^{\pm 0.0124}$	$0.0991^{\pm 0.0136}$	$\mathbf{0.0789}^{\pm 0.0095}$
Orientation tensor rel-L2	$0.0869^{\pm 0.0117}$	$0.0814^{\pm 0.0110}$	$\mathbf{0.0740}^{\pm 0.0096}$
Strain tensor rel-L2	$0.2017^{\pm 0.0231}$	$0.1931^{\pm 0.0233}$	$\mathbf{0.1760}^{\pm 0.0195}$
Rotation consistency and generalization			
Tensor C_4 error	$0.1432^{\pm 0.0101}$	$0.1215^{\pm 0.0061}$	$\mathbf{0.1115}^{\pm 0.0080}$
Full-field C_4 error	$0.1093^{\pm 0.0076}$	$0.0932^{\pm 0.0049}$	$\mathbf{0.0839}^{\pm 0.0059}$
Rotated-OOD full-state rel-L2	$0.1186^{\pm 0.0122}$	$0.1123^{\pm 0.0125}$	$\mathbf{0.0988}^{\pm 0.0107}$
Physical consistency			
Strain consistency rel-L2	$0.2367^{\pm 0.0790}$	$0.2251^{\pm 0.0815}$	$\mathbf{0.1568}^{\pm 0.0500}$
Long-horizon prediction			
Far-horizon raw full-state rel-L2	$0.9765^{\pm 0.0406}$	$\mathbf{0.9698}^{\pm 0.0327}$	$0.9879^{\pm 0.0395}$

Observations. Large Scratch improves normalized VRMSE by 4.14% relative to Ordinary Scratch. The complete UNIJEPA protocol improves a further 10.06% relative to Large Scratch and wins all five paired seeds. The paired mean difference (UNIJEPA minus Large Scratch) is -0.01468, with paired 95% confidence interval $[-0.01759, -0.01262]$.

The far-horizon row is a deliberate negative result. UNIJEPA improves one-step accuracy and most structure-sensitive diagnostics but does not improve the 20-step autoregressive endpoint. This separates short-horizon representation quality from rollout stability.

E.3.1 EXACTLY C_4-EQUIVARIANT OPERATOR BASELINE

Does exact rotational equivariance alone account for the transfer gains? We answer this with a $p4$-GFNO trained from scratch on Active Matter. The baseline is C_4-equivariant by construction, uses no auxiliary physics or equivariance loss, and matches the Ordinary Scratch FNO within 0.10% in trainable parameters (8.43M versus 8.42M). All three methods use the same budget-16 data split, four-frame context, validation-only model selection, and test evaluator; results below use seeds 0–2 and are kept separate from the five-seed comparison above.

Table 17: Active Matter comparison with an exactly C_4-equivariant operator matched to Ordinary Scratch in parameter count. We report mean ± standard deviation over three seeds; lower is better and the best value is bold.

Metric ↓	Ordinary Scratch	$p4$-GFNO	UNIJEPA
State and field prediction			
Full-state VRMSE	$0.1407^{\pm 0.0031}$	$0.1324^{\pm 0.0059}$	$\mathbf{0.1215}^{\pm 0.0020}$
Velocity rel-L2	$0.0936^{\pm 0.0026}$	$0.0843^{\pm 0.0064}$	$\mathbf{0.0743}^{\pm 0.0007}$
Orientation rel-L2	$0.0803^{\pm 0.0020}$	$0.0807^{\pm 0.0060}$	$\mathbf{0.0685}^{\pm 0.0020}$
Strain rel-L2	$0.1883^{\pm 0.0029}$	$0.1738^{\pm 0.0085}$	$\mathbf{0.1647}^{\pm 0.0014}$
Rotation consistency and generalization			
Full-field C_4 error	$0.1049^{\pm 0.0043}$	$(\mathbf{3.9 \pm 0.2}) \times \mathbf{10^{-7}}$	$0.0805^{\pm 0.0027}$
Rotated-OOD rel-L2	$0.1116^{\pm 0.0009}$	$0.0985^{\pm 0.0044}$	$\mathbf{0.0927}^{\pm 0.0004}$
Physical consistency			
Strain consistency	$0.2693^{\pm 0.0689}$	$0.4823^{\pm 0.2560}$	$\mathbf{0.1812}^{\pm 0.0197}$

Observations. The $p4$-GFNO improves over Ordinary Scratch and reduces the measured C_4 error to numerical precision, confirming the value of hard rotational symmetry. UNIJEPA nevertheless achieves the best forecasting, rotated-input, and strain-consistency results, including an 8.2% lower full-state VRMSE than the $p4$-GFNO. Exact equivariance and typed predictive pretraining are therefore complementary: the former enforces a transformation law, while the latter improves the organization and transfer of physical representations.

Seed-level audit subset. Table 18 lists the seed-level test JSON values available for seeds 0–2 in the exported evidence bundle. The principal claim remains the locked five-seed aggregate above.

Table 18: Active Matter seed-level audit subset. Each seed has separate Scratch and UNIJEPA columns.

Metric ↓	Seed 0		Seed 1		Seed 2	
	Scratch	UNIJEPA	Scratch	UNIJEPA	Scratch	UNIJEPA
Normalized state	0.13976	0.12179	0.13826	0.11938	0.14420	0.12335
Raw state	0.10514	0.09084	0.10358	0.08965	0.10735	0.09148
Velocity	0.09391	0.07511	0.09081	0.07391	0.09596	0.07379
Orientation	0.07916	0.06785	0.07913	0.06699	0.08255	0.07073
Strain	0.18822	0.16524	0.18544	0.16315	0.19115	0.16572
Tensor C_4	0.13238	0.10332	0.13661	0.10754	0.14258	0.10984
Rotated OOD	0.11197	0.09286	0.11055	0.09220	0.11223	0.09306
Strain consistency	0.19609	0.17374	0.27906	0.16639	0.33276	0.20362
Far-horizon raw	0.93152	0.95011	0.95116	1.00144	1.01802	1.01253

E.3.2 QUALITATIVE FIELD COMPARISONS

Figure 5 compares the full UNIJEPA forecasting model with a frozen-encoder Channel-blind model at budget 16 and seed 0. The full model fine-tunes its encoder and uses downstream physics and rotation losses, so this visualization complements the matched interface ablation in Appendix F.4 as a full-method comparison.

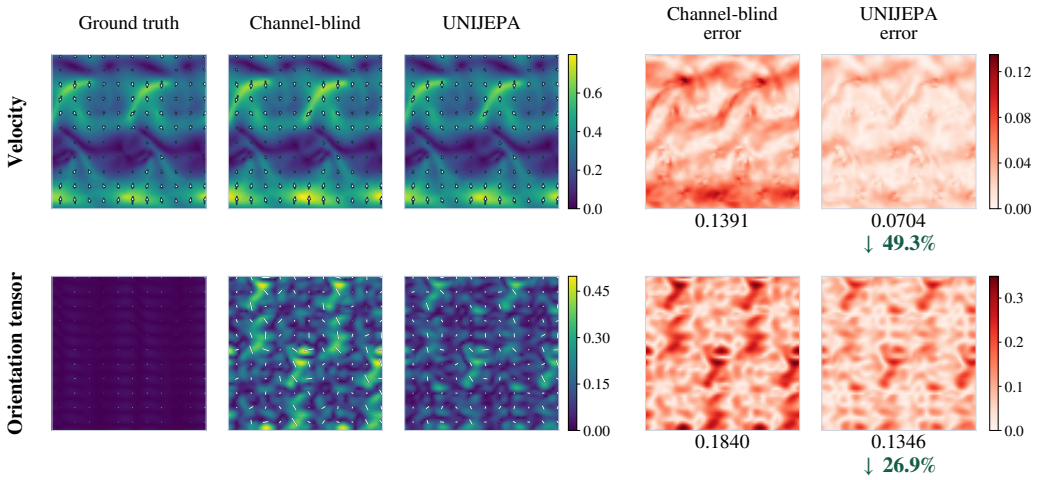

Figure 5: Selected one-step Active Matter predictions and spatial errors. The velocity and orientation rows use different test windows, each selected from eight fixed windows for the largest relative error reduction in that field. Velocity colors and arrows show speed and direction; tensor colors and rods show anisotropy magnitude and principal direction, with rod length proportional to anisotropy. Error maps show $\|\widehat{\mathbf{u}} - \mathbf{u}\|_2$ and $\|\widehat{D} - D\|_F$, with shared scales within each row. Numbers report sample-level relative errors; downward arrows indicate reductions of 49.3% and 26.9% over Channel-blind.

F ADDITIONAL CONTROLLED STUDIES

Following the main comparisons, we separate training-signal effects, representation readability, interface choice, and downstream symmetry regularization. The training-signal and interface studies

use matched downstream protocols with downstream physics and transformation losses set to zero; the separate C_4-regularized Scratch control evaluates downstream symmetry regularization.

F.1 Training-Signal Ablations

Every row uses the same architecture, labeled subset, frozen target encoder, optimization schedule, and downstream seeds. Downstream physics and transformation losses are zero for all rows. The weights in Table 19 denote only the corresponding representation-pretraining signals, so the comparison isolates what those signals contribute before downstream forecasting.

Table 19: Three-seed Active Matter representation-pretraining signal ablation. The displayed structural weights are used only during pretraining; downstream physics and transformation losses are zero for every row.

Training signal	Pretraining weights	
	$\lambda_{\text{phys}}^{\text{pre}}$	$\lambda_{\text{eq}}^{\text{pre}}$
JEPA only	0	0
JEPA + auxiliary	0	0
+ physics	.02	0
+ rotation	0	.05
Complete UniJEPA	.02	.05

Training signal	State ↓	Orientation ↓	Tensor C_4 ↓	Strain cons. ↓
JEPA only	$0.1430^{\pm 0.0077}$	$0.0807^{\pm 0.0037}$	$0.1385^{\pm 0.0066}$	$0.2035^{\pm 0.0673}$
JEPA + auxiliary	$0.1427^{\pm 0.0076}$	$0.0806^{\pm 0.0035}$	$0.1381^{\pm 0.0066}$	$0.1992^{\pm 0.0600}$
+ physics	$0.1405^{\pm 0.0078}$	$0.0803^{\pm 0.0037}$	$0.1351^{\pm 0.0069}$	$\mathbf{0.1564}^{\pm 0.0585}$
+ rotation	$0.1306^{\pm 0.0059}$	$0.0736^{\pm 0.0022}$	$0.1173^{\pm 0.0056}$	$0.1899^{\pm 0.0582}$
Complete UniJEPA	$\mathbf{0.1215}^{\pm 0.0020}$	$\mathbf{0.0685}^{\pm 0.0020}$	$\mathbf{0.1069}^{\pm 0.0033}$	$0.1812^{\pm 0.0197}$

The corresponding per-seed normalized-state errors (↓) are:

Configuration	Seed 0 ↓	Seed 1 ↓	Seed 2 ↓
JEPA only	0.14029	0.15175	0.13699
JEPA + auxiliary	0.13985	0.15135	0.13687
+ physics	0.13744	0.14933	0.13459
+ rotation	0.12882	0.13712	0.12578
Complete UniJEPA	0.12179	0.11938	0.12335

Observations. Auxiliary decoding alone changes little. The physics term has its largest relative effect on strain consistency, the rotation term improves state and C_4 errors broadly, and their combination gives the best state, orientation, and C_4 means.

F.2 Physical-Loss Gradient Diagnostics on Active Matter

Protocol. We compare JEPA + auxiliary with JEPA + auxiliary + physics at seed 0, pretraining epoch 30, with rotation supervision disabled in both. For the same 24 validation trajectories, we average physical-loss gradients over four windows per trajectory with respect to the online encoder parameters. The physical loss includes tensor realizability and strain consistency, and is evaluated with the same multiplier 0.02 for both checkpoints. No parameters are updated during this analysis.

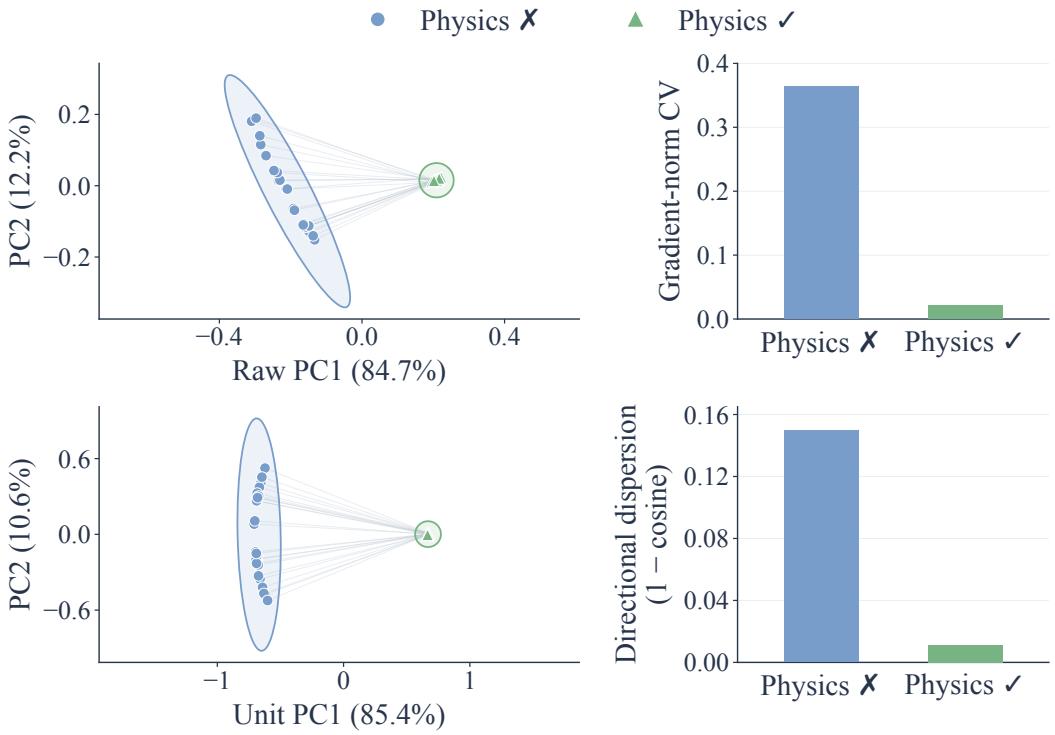

Figure 6: Physical-loss gradients with (green) and without (blue) physics supervision. Left: joint PCA of raw gradients (top) and unit-normalized gradients (bottom). Each point represents one validation trajectory; lines connect matched trajectories, and ellipses enclose all points rather than indicate confidence regions. Right: gradient-norm CV and directional dispersion, computed in the original gradient space; lower is better for both.

Results. Figure 6 shows a more concentrated gradient distribution with physics supervision. Gradient-norm CV (standard deviation divided by mean) decreases from 0.3641 to 0.0218, a 94.0% reduction. Directional dispersion (one minus mean pairwise cosine) decreases from 0.1497 to 0.0111, a 92.6% reduction. Raw PCA reflects magnitude and direction together, while unit-normalized PCA removes magnitude differences. Together, these results show more balanced physical-gradient magnitudes and more aligned directions across trajectories in this seed-0 comparison.

F.3 Frozen Predictive-Latent Probes

The probe consumes predicted future field tokens. The online encoder and matching online JEPA predictor are loaded from a fixed representation checkpoint and placed in evaluation mode. Every representation parameter has `requires_grad=false`; only the readout is fitted. The three MLP seeds vary readout initialization and minibatch order, with the representation checkpoint and labeled trajectories fixed.

Targets are non-overlapping patch means at the next time step: orientation (q_1, q_2), strain (e_1, e_2), and scalar vorticity ω. The probe has no raw-state path, convolution, Fourier layer, trainable adapter, or spatial token mixing. Table 20 focuses on orientation and strain. Across all three tasks, the linear readouts have 1,285 fitted coefficients; the MLP heads have 49,669 trainable parameters with hidden width 64. The labeled budget remains 16 trajectories, and feature standardization uses training statistics only; MLP targets are also standardized using training statistics.

Linear probes use ridge regression with regularization strengths selected from $\{0, 10^{-10}, 10^{-9}, \ldots, 10^{1}\}$: 10^{-4} for Random, 10^{-5} for JEPA, and 10^{-4} for UniJEPA. MLPs use AdamW, batches of 4,096 patch tokens, and at most 300 epochs. All methods share

learning-rate/weight-decay candidates $(10^{-3}, 0)$, $(3 \times 10^{-3}, 0)$, $(10^{-2}, 0)$, and $(3 \times 10^{-3}, 10^{-4})$. Seed-0 validation selects the configuration, reused for seeds 1 and 2. Validation every five epochs controls the plateau learning-rate schedule, early stopping, and checkpoint selection. The stopping patience is 60 epochs after a minimum of 100. Selection minimizes mean validation relative error across the three tasks; all choices are fixed before testing. Ridge results are deterministic; MLP results report three-seed mean and sample standard deviation.

Table 20: Frozen orientation and strain readout errors (↓). Linear probes use deterministic ridge regression; MLP results are three-seed means with sample standard deviations over readout training. Mean error averages orientation and strain.

Method	Orientation ↓	Strain ↓	Mean error ↓
Linear probe			
Random	0.2239	0.2843	0.2541
JEPA	0.2391	0.2898	0.2645
UNIJEPA	**0.1312**	**0.1774**	**0.1543**
Two-layer MLP probe			
Random	$0.1556^{\pm 0.0005}$	$0.2334^{\pm 0.0002}$	$0.1945^{\pm 0.0003}$
JEPA	$0.2235^{\pm 0.0051}$	$0.2803^{\pm 0.0024}$	$0.2519^{\pm 0.0036}$
UNIJEPA	$\mathbf{0.1078}^{\pm 0.0003}$	$\mathbf{0.1592}^{\pm 0.0004}$	$\mathbf{0.1335}^{\pm 0.0003}$

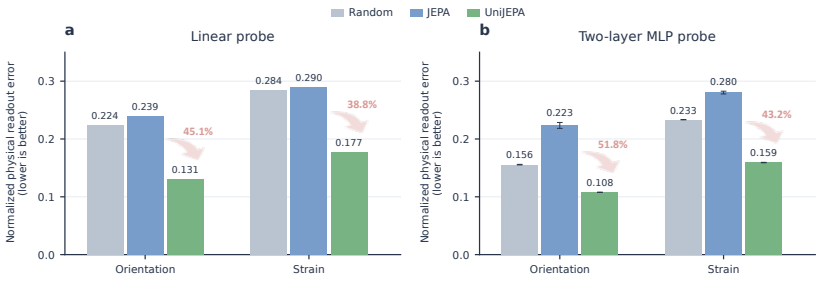

Figure 7: Frozen orientation and strain readout errors (lower is better). UNIJEPA achieves the lowest errors for both quantities under linear ridge and two-layer MLP readouts. MLP error bars show sample standard deviations over three readout-training seeds.

Observations. Relative to JEPA, linear UNIJEPA probes reduce orientation and strain errors by 45.1% and 38.8%; MLP probes reduce them by 51.8% and 43.2%. Relative to Random, the corresponding reductions are 41.4% and 37.6% for linear probes, and 30.7% and 31.8% for MLP probes. These comparisons use the same readout architectures and validation-selection procedure, with the encoder and predictor frozen throughout.

F.4 CHANNEL-BLIND COMPARISON

The channel-blind representation keeps all seven raw channels but removes explicit scalar/vector/tensor routing. The channel-blind and typed variants use the same representation pretraining signals, encoder initialization and tuning policy, adapters, FNO, labeled subset, optimization schedule, and downstream seeds. Downstream physics and transformation losses are zero for both rows, leaving the representation interface as the only changed component.

Table 21: Matched typed-interface ablation on Active Matter. Channel-blind retains the full raw state and the same pretraining signals; only explicit scalar/vector/tensor routing is removed.

Metric ↓	Channel-blind	Typed UNIJEPA
State	$0.1420^{\pm 0.0019}$	$\mathbf{0.1215}^{\pm 0.0020}$
Raw state	$0.1066^{\pm 0.0010}$	$\mathbf{0.0907}^{\pm 0.0009}$
Orientation	$0.0809^{\pm 0.0013}$	$\mathbf{0.0685}^{\pm 0.0020}$
Tensor C_4	$0.1335^{\pm 0.0007}$	$\mathbf{0.1069}^{\pm 0.0033}$
Rotated OOD	$0.1131^{\pm 0.0006}$	$\mathbf{0.0927}^{\pm 0.0004}$
Strain consistency	$0.2201^{\pm 0.0253}$	$\mathbf{0.1812}^{\pm 0.0197}$

Observations. Typed UNIJEPA improves all displayed state- and structure-sensitive metrics under the matched protocol. Because raw channel access, pretraining supervision, downstream optimization, and evaluation are held fixed, the gap isolates the contribution of explicit field typing at the representation interface.

F.5 DOWNSTREAM C_4-REGULARIZED SCRATCH CONTROL

The archive includes a supplementary Scratch FNO trained with a downstream C_4 consistency loss. This control answers a different question from the main tables: how far can explicit downstream symmetry regularization go without predictive pretraining? It uses the same 8,416,967-parameter width-64 FNO as Ordinary Scratch.

The regularization weight was selected without test access from $\{0.01, 0.05, 0.10\}$. Mean validation selection scores over seeds 0–2 were $0.1103^{\pm 0.0053}$, $0.1035^{\pm 0.0043}$, and $\mathbf{0.0988}^{\pm 0.0039}$, respectively, so $\lambda_{C_4} = 0.10$ was promoted. The selected weight is retained for the updated evaluation over five seeds (0–4), reported below as mean ± sample standard deviation.

Table 22: Updated five-seed downstream C_4-regularized Scratch FNO with $\lambda_{C_4} = 0.10$. This is not a zero-auxiliary-loss baseline.

Metric ↓	Mean ± sample SD
State and field prediction	
Normalized full-state VRMSE	$0.1312^{\pm 0.0140}$
Raw full-state rel-L2	$0.0979^{\pm 0.0102}$
Velocity rel-L2	$0.0830^{\pm 0.0089}$
Orientation tensor rel-L2	$0.0751^{\pm 0.0084}$
Strain tensor rel-L2	$0.1763^{\pm 0.0179}$
Rotation consistency and generalization	
Tensor C_4 error	$0.1122^{\pm 0.0074}$
Full-field C_4 error	$0.0852^{\pm 0.0055}$
Rotated-OOD full-state rel-L2	$0.0995^{\pm 0.0098}$
Physical consistency	
Strain consistency rel-L2	$0.2156^{\pm 0.0737}$
Long-horizon prediction	
Far-horizon raw full-state rel-L2	$0.9703^{\pm 0.0348}$

Observations. The updated control reaches a similar mean full-state VRMSE to UNIJEPA in the five-seed comparison (0.1312 versus 0.1301), confirming that explicit downstream geometric regularization is effective.

G SELECTED CROSS-BACKBONE RESULTS

Table 23 reports selected backbone-specific comparisons on Active Matter with 16 labeled trajectories and three downstream seeds. The augmented model combines UNIJEPA with downstream physical constraints. These are selected metrics, not the complete evaluation suite; the Transolver selection includes orientation and H10 errors. Each entry is a mean with sample standard deviation in superscript. Relative reductions are taken from the experiment reports and were calculated before rounding.

Table 23: Selected cross-backbone results on Active Matter. Base denotes the supervised backbone; +UNIJEPA includes downstream physical constraints. All errors are lower-is-better. Blue marks the augmented model and orange the relative error reduction.

Metric ↓	Base	+UNIJEPA	**Reduction ↑**
MWT			
Full-state VRMSE	$0.132035^{\pm 0.003490}$	$\mathbf{0.123703}^{\pm 0.002866}$	**6.31%**
Velocity rel-L2	$0.085818^{\pm 0.006369}$	$\mathbf{0.075123}^{\pm 0.004095}$	**12.46%**
Full-field C_4 error	$0.095722^{\pm 0.000928}$	$\mathbf{0.086187}^{\pm 0.001584}$	**9.96%**
Strain consistency rel-L2	$0.420759^{\pm 0.028931}$	$\mathbf{0.151720}^{\pm 0.030673}$	**63.94%**
CNO			
Full-state VRMSE	$0.126519^{\pm 0.003324}$	$\mathbf{0.121463}^{\pm 0.003953}$	**4.00%**
Velocity rel-L2	$0.101882^{\pm 0.002017}$	$\mathbf{0.091992}^{\pm 0.003119}$	**9.71%**
Full-field C_4 error	$0.095055^{\pm 0.001569}$	$\mathbf{0.088147}^{\pm 0.000475}$	**7.27%**
Strain consistency rel-L2	$0.705112^{\pm 0.056455}$	$\mathbf{0.487819}^{\pm 0.020889}$	**30.82%**
PDE-Transformer			
Full-state VRMSE	$0.116606^{\pm 0.004048}$	$\mathbf{0.113525}^{\pm 0.002364}$	**2.64%**
Velocity rel-L2	$0.090688^{\pm 0.001361}$	$\mathbf{0.086830}^{\pm 0.000766}$	**4.25%**
Full-field C_4 error	$0.069798^{\pm 0.001668}$	$\mathbf{0.067101}^{\pm 0.001863}$	**3.86%**
Strain consistency rel-L2	$0.391818^{\pm 0.054897}$	$\mathbf{0.238089}^{\pm 0.020981}$	**39.23%**
Transolver			
Full-state VRMSE	$0.149423^{\pm 0.018009}$	$\mathbf{0.136080}^{\pm 0.001381}$	**8.93%**
Orientation tensor rel-L2	$0.078312^{\pm 0.012139}$	$\mathbf{0.070464}^{\pm 0.003297}$	**10.02%**
H10 full-state rel-L2	$1.056566^{\pm 0.136799}$	$\mathbf{0.931115}^{\pm 0.035890}$	**11.87%**
Strain consistency rel-L2	$0.309115^{\pm 0.043857}$	$\mathbf{0.253130}^{\pm 0.049797}$	**18.11%**

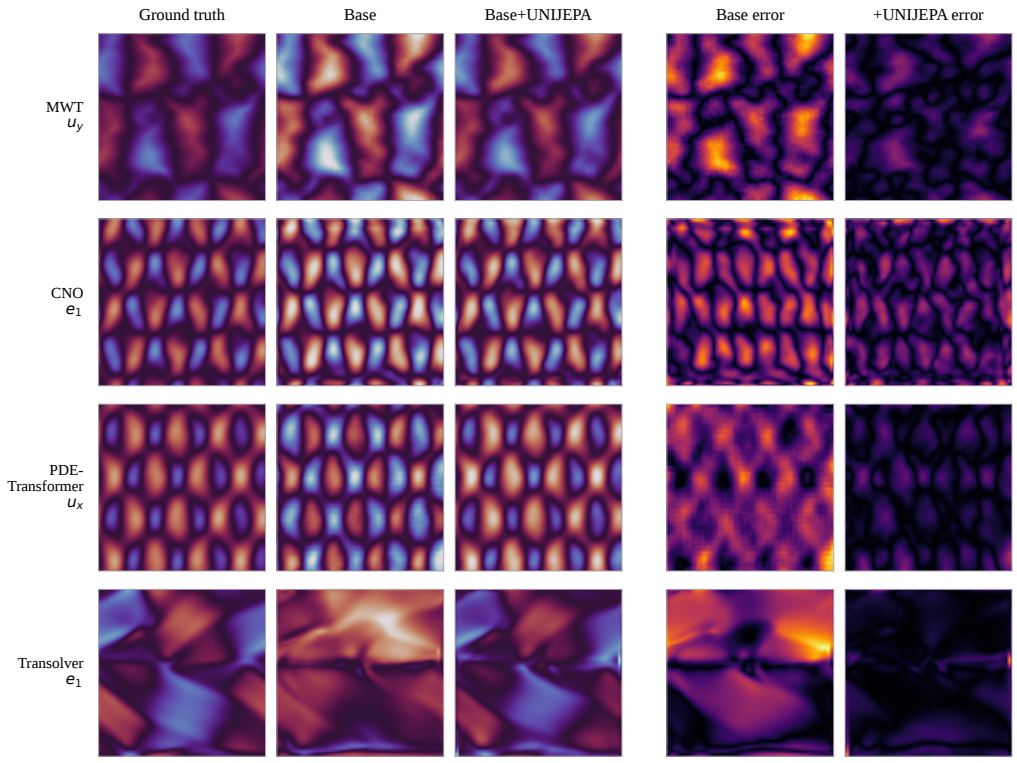

Figure 8: **Qualitative cross-backbone prediction on Active Matter.** Selected seed-0 validation examples of one-step prediction, chosen for clear error reductions. Each row compares ground truth, Base, Base+UNIJEPA, and their pointwise absolute errors. Within each row, the three fields share a color scale covering their full value range; the two error maps share a separate scale, with brighter colors indicating larger errors. Base+UNIJEPA includes downstream physical constraints.

H Code Availability and Reproducibility

H.1 Supplementary code

The anonymous supplementary ZIP contains the scalar, CFD2D, and Active Matter implementations, including representation pretraining, physical objectives, downstream prediction, frozen-latent probes, ablations, and evaluation. The accompanying documentation describes the data preparation steps and the entry points for each experiment. Archived configurations are provided for unified pretraining and two Active Matter seed-0 runs. Dataset arrays and pretrained checkpoints are not included in the code package.

H.2 Running the experiments

The experiments used Python 3.10.20 and PyTorch 2.5.1+cu121. Dependencies and example commands are provided in `requirements.txt` and `docs/RUNNING.md`; `docs/CODE_MAP.md` maps the experiments to their scripts. The command-line entry points and supplied synthetic tests have been checked in the original H100 environment.

Data preprocessing and split construction follow Appendix C.1, and model and optimization settings are given in Appendix C.2. Each comparison uses its specified labeled budget and seed set. The labeled budget counts complete trajectories, with the same subset used across methods within each comparison. Checkpoints are selected using validation data before evaluation on the test split.

H.3 Reporting results

Except for deterministic ridge probes, tables report the arithmetic mean and sample standard deviation of the per-seed test metrics. Table 9 lists the seed sets and evaluation units for each study. Three-seed comparisons and five-seed extensions are reported separately. For paired comparisons, confidence intervals are computed from the differences between methods at corresponding seeds. Normalized full-state error, raw relative error, and physical-structure metrics are reported separately using the definitions given above.

I Limitations

Scope of physical structure. Field types and applicable physical anchors are supplied from known variable semantics. The present transformation study is limited to C_4 on a square periodic grid, and the released Active Matter state does not contain the fourth-moment closure required for the full kinetic-theory residual. Extending the same principle to unknown channel semantics, unstructured meshes, three-dimensional rotations, or partially observed physics remains future work.

Long-horizon behavior. The method improves one-step prediction and most structure-sensitive diagnostics, but it does not improve the 20-step Active Matter endpoint. The present transfer protocol is optimized for short-horizon representation quality; extending it with rollout-aware training remains future work.