SPLICE: SPARSE PHYSICS LIBRARY IDENTIFICATION AND COMPOSITIONAL EXPANSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operator splitting composes a library of neural operators, one per physical mechanism, into zero-shot simulators of new equations. Existing approaches, however, require these mechanisms and their labels to be specified in advance. We ask whether a pretrained library can instead be adapted and grown from unlabeled trajectories in which multiple mechanisms act simultaneously. We introduce SPLICE, a framework that discovers operators without mechanism labels on mixture data. The contributions of the candidate operators to each trajectory are inferred by solving a convex optimization problem. A calibrated test then assigns each discovered operator the mechanism it implements. Failure cases are also detectable, when the library lacks the required physics or when mechanisms are insufficiently separable. Across one and two dimensional systems, the discovered operators recover mechanisms from unlabeled mixtures and compose into zero-shot simulators of unseen equations; on Navier-Stokes they compose about as accurately as their supervised counterparts on average.

1 INTRODUCTION

Numerical simulation of physical systems underpins science and engineering, yet each new system typically requires its own solver or its own trained model (Raissi et al., 2019; Kochkov et al., 2021; Pfaff et al., 2020). Physical systems, however, share structure: many are driven by a small set of basic effects, such as transport and spreading, acting together. Neural operator splitting (NOS) (Serrano et al., 2026) exploits this by learning one neural operator per effect and combining them at test time, so that a single reusable library simulates equations never seen in training, instead of one model per system (McCabe et al., 2023; Herde et al., 2024).

Such libraries must be specified in advance. Each operator is trained on data in which its effect acts alone and is labeled, so adding an effect requires new labeled data. Real trajectories rarely come this way: instead, several unlabeled effects act at once. We therefore ask: *can an operator library learn new operators from unlabeled trajectories in which several effects act simultaneously, and can we tell what each learned operator does?*

We introduce SPLICE to answer both questions (Figure 1). It starts from a library pretrained on a few labeled effects, adapts its operators on unlabeled data, and grows the library when the data reveal dynamics it cannot yet express; neither step uses effect labels. A final identification step then tests which physical effect each learned operator implements, by comparing its action with reference solvers that are never used for training.

In one and two dimensional systems, SPLICE recovers the hidden effects of unlabeled trajectories, including fluid flows in which no effect is ever observed alone, and the learned operators combine to simulate equations unseen during training. On Navier-Stokes, the discovered libraries compose about as accurately as their supervised counterparts, and can re-grow a missing library entry. The experiments also mark where the approach stops: effects with near-identical actions end up sharing one operator, an operator can be correctly identified yet fail to improve a particular combination (§5.5), and growth re-grows effects that the pretrained library can represent, while learning an effect absent from pretraining remains open.

Contributions.

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.

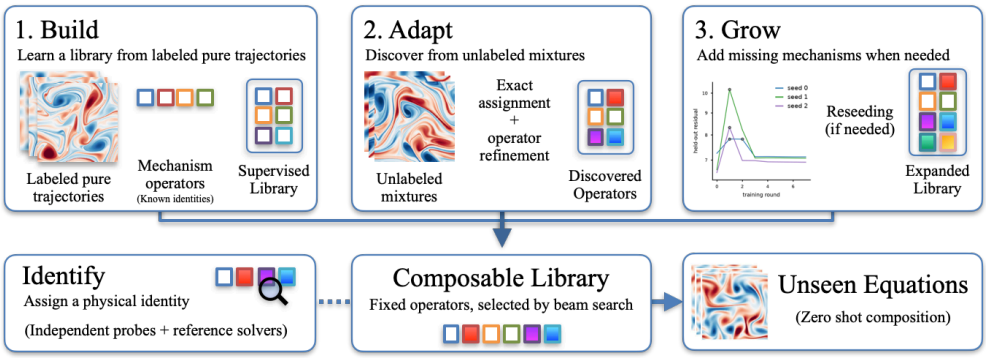

Figure 1: The SPLICE pipeline. *Build* trains one operator per labeled effect and composes them, following Serrano et al. (2026); *Adapt*, *Grow*, and *Identify* are ours. *Adapt* learns operators from unlabeled trajectories, and *Grow* adds an operator when the data reveal dynamics the library cannot express. *Identify* tests each learned operator against reference solvers used only for evaluation, never for training. At test time, a beam search selects which operators of the library to compose for each unseen equation.

- **A label-free method to adapt and grow operator libraries.** We introduce SPLICE, which adapts a pretrained operator library on unlabeled trajectories in which several effects act at once, and adds an operator when the data reveal dynamics the library cannot express.
- **A protocol to identify what each operator learned.** We propose an evaluation protocol, kept separate from training, that assigns each learned operator a physical effect by comparison with reference solvers, or reports that none can be assigned, so that a failing simulation points to the operator responsible.
- **Evidence that label-free libraries match supervised ones.** On 1D and 2D systems, including Navier–Stokes flows in which no effect ever acts alone, the adapted libraries recover the hidden effects and compose Navier–Stokes as accurately as supervised libraries.

2 Related work

Neural operators such as the Fourier neural operator (FNO) (Li et al., 2020) learn maps between function spaces (Lu et al., 2021; Kovachki et al., 2023; Koren and Lanthaler, 2025). Breadth is increasingly pursued through scale and conditioning: multi-physics pretraining (McCabe et al., 2023; Herde et al., 2024; Hao et al., 2024; Alkin et al., 2024), in-context and adaptive conditioning (Serrano et al., 2024; Kassaï Koupaï et al., 2024; Yang et al., 2023), PDE-conditional transformers (Zhou et al., 2025), and continually grown component libraries (Tripura and Chakraborty, 2023; Mendez and EATON, 2021), with labeled per-physics data in the physical case. All reach coverage through scale, conditioning, or adaptation rather than through a library of independently identifiable mechanism operators.

Classical operator splitting (Strang, 1968) offers a different route: compose solvers for the individual terms of the governing equation. NOS (Serrano et al., 2026) makes these solvers learned and searches over their compositions at test time; we build on it and ask how such a library can be discovered and grown when mechanism identities are unavailable. Concurrent work, HyCOP (Zhao et al., 2026), also builds a dictionary of composable PDE primitives, but from supervised residuals with a learned composition schedule, where we discover operators without mechanism labels under the fixed beam-searched Strang composition of NOS.

SINDy (Brunton et al., 2021), PDE-FIND (Rudy et al., 2017), and their weak-form and ensemble variants (Fasel et al., 2022) recover governing equations by sparse regression over candidate terms, and neural variants learn the terms themselves (Long et al., 2018; Both et al., 2021; Champion et al., 2019) or distill them from trained models (Cranmer et al., 2020). DMD and Koopman methods fit global evolution operators (Schmid, 2010; Brunton et al., 2021), operator inference assumes the governing structure (Peherstorfer and Willcox, 2016), and switching state-space models find regimes without naming them (Linderman et al., 2017). These methods recover equations or components

within a specified representation; few yield reusable per-mechanism solvers whose physical identity is established by an external test. PDE-FIND also serves as a baseline in §5.3.

LEADS (Yin et al., 2021), CoDA (Kirchmeyer et al., 2022), and DyAd (Wang et al., 2022) decompose unlabeled dynamics into shared or environment-specific components, but the components are not assigned a physical identity, and independent-mechanism learning discovers competing modules without naming them (Parascandolo et al., 2018). Unsupervised identifiability is known to require external structure (Locatello et al., 2019); our setting supplies it after the fact: a discovered operator is graded against independently specified reference dynamics, which makes discovery a falsifiable identification problem rather than only a decomposition.

Closest to our evaluation design, Garrido et al. (2023) grade self-supervised representations against transformation parameters known only to the evaluator, as discovered operators are graded here against a mechanism basis withheld from training. The Well (Ohana et al., 2024) supplies our external transfer data (§C.5); its coupled multi-mechanism systems suit this setting more directly than largely single-equation suites (Takamoto et al., 2022).

3 Approach

3.1 Background: neural solver operators and their composition

A trajectory is a sequence of fields $u_{0:T}$, with $u_t \in \mathbb{R}^{C\times N}$ in 1D and $u_t \in \mathbb{R}^{C\times N\times N}$ in 2D, where C is the number of physical channels (e.g. velocity components) and N the number of grid points per dimension; Table 4 (appendix) collects all symbols. We assume the dynamics are a sum of a few effects, $\partial_t u = \sum_m c_m A_m(u)$, where each A_m is a primitive mechanism (one of the basic effects of §1), such as advection (transport) or diffusion (spreading), and $c_m \geq 0$ is its strength.

Following Serrano et al. (2026), a single neural network represents every operator. An encoder E reads a context window of L frames and outputs a latent code $\theta = E(u_{t-L:t}) \in \mathbb{R}^d$. A hypernetwork maps θ to the weights of a vector field f_θ, and one step integrates this field with the classical fourth-order Runge–Kutta scheme (RK4), a neural ODE step (Chen et al., 2018):

$$\Phi_\theta(u_t) = \mathrm{RK4}[f_\theta](u_t) \approx u_{t+1}. \tag{1}$$

Trained on labeled single-mechanism trajectories, the code space organizes by mechanism, so each region of θ space acts as one operator. The library $\mathcal{L} = \{\Phi_{\theta_1}, \ldots, \Phi_{\theta_M}\}$ holds one operator per mechanism, M being the number of mechanisms in the labeled data (Figure 2).

To simulate an equation with several active mechanisms, operators are applied in sequence within each time step (Strang splitting: $\Phi_A^{1/2} \circ \Phi_B \circ \Phi_A^{1/2}$, local error $O(\Delta t^3)$). Which operators to combine is chosen at test time by beam search, a pruned search that keeps the few best partial compositions as it extends them and scores each by rollout error. We use this machinery as in Serrano et al. (2026): an unseen equation can be simulated when its mechanisms are represented in the library.

3.2 Build: learning a library of mechanism operators

PRIM-1D extends the two and three family settings of NOS to eight primitive families with dense coefficient coverage. We study corpus design and solver fidelity primarily in 1D, where exact spectral solvers make labeled data inexpensive to generate and multi seed studies practical. Before an operator is used in library, we require stable rollout on trajectories generated by its own mechanism.

3.3 Adapt: discovering operators from unlabeled mixtures

We now observe trajectories whose dynamics are a sum of mechanisms as in Section 3.1 with one or two active mechanisms and random coefficients, but with no label saying which mechanisms are active. The goal is to learn a set of *discovered* operators $\{\hat{\Phi}_1, \ldots, \hat{\Phi}_K\}$, each of the form in Eq. (1) with its own learnable code fed to the frozen pretrained hypernetwork, together with per trajectory assignments indicating which discovered operators contribute to each trajectory. Throughout, hatted symbols denote discovered quantities, k indexes discovered operators, and m is reserved for physical mechanisms; K need not equal the number of supervised families M and is chosen without labels

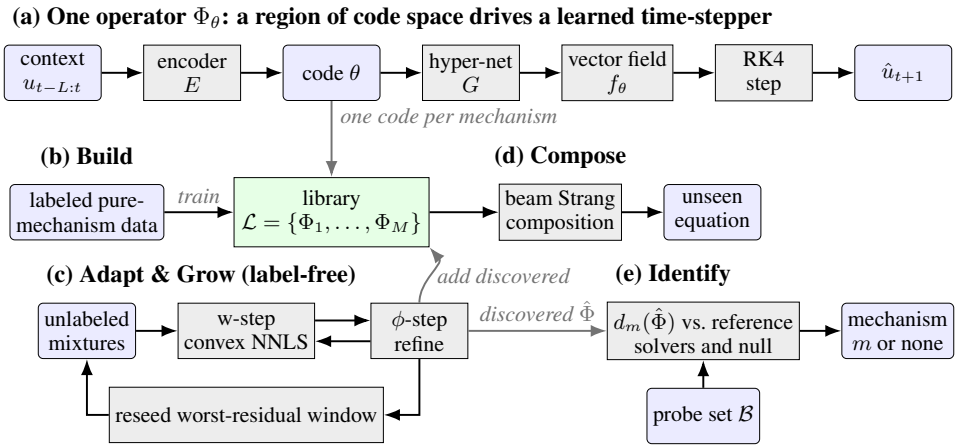

Figure 2: The shared architecture and where each stage acts. **(a)** Every library entry is one learned time-stepper, in which an encoder maps a context window to a code θ, a hypernetwork turns θ into the weights of a vector field f_θ, and RK4 advances one step, so one region of code space is one operator. **(b)** *Build* trains the library on labeled pure-mechanism trajectories. **(c)** *Adapt and Grow* alternate a convex assignment (w-step) with operator refinement (ϕ-step) on unlabeled mixtures, and reseed an operator from the worst-reconstructed window. **(d)** *Compose* beam-searches Strang compositions of library operators. **(e)** *Identify* grades each discovered operator against reference solvers on a fixed probe set. Blue nodes are data; grey nodes are learned modules.

(predictive K, below). Both discovery losses are defined only from observed future frames. The physical mechanism basis is withheld from discovery and used only for post hoc identification in §3.5. We denote the resulting discovered library by $\mathcal{L}^{\mathrm{SSL}}$.

Exact assignment by convex optimization. Because the active mechanisms are unobserved, discovery alternates between inferring which discovered operators contribute to each trajectory and refining them given those assignments. For fixed discovered operators, the inference step requires no learned component: because the dynamics are additive, the nonnegative mixture weights are obtained by a least squares problem solved to optimality for each trajectory. The joint problem over discovered operators and assignments remains nonconvex. Learned alternatives to this assignment step recover less and are compared in Appendix C.1. Let z_t denote the frame u_t standardized by its window mean and scale, and $\Delta z_t = z_{t+1} - z_t$ the resulting one step increment, a finite difference estimate of $\partial_t u$ at time t. The alternating procedure is

$$w\textbf{-step (exact, convex):} \quad w^\star = \arg\min_{w \geq 0} \left\| \sum_{k=1}^{K} w_k \hat{f}_k(z_t) - \Delta z_t \right\|_2^2 + \lambda \big(\textstyle\sum_k w_k \big)^2, \tag{2}$$

$$\phi\textbf{-step (gradient):} \quad \min_{\hat{\phi}} \quad \sum_{h \in \{1,4\}} \left\| \hat{\Phi}^{(h)}_{\hat{\phi}, w^\star}(u_t) - u_{t+h} \right\|_2^2. \tag{3}$$

Here $\| \cdot \|_2$ is the Euclidean norm over grid points, $w \in \mathbb{R}^K_{\geq 0}$ are per trajectory mixture weights, $\hat{\phi}$ collects the codes of the discovered operators, and $\hat{\Phi}^{(h)}_{\hat{\phi}, w}$ rolls out the weight mixed field for h steps from frame u_t. The w-step is analogous to sparse regression in SINDy (Brunton et al., 2016), with learned operators replacing symbolic terms. The multi horizon ϕ-step refines the discovered operators with the assignment fixed and exposes slower mechanisms that may contribute little to one step prediction. We use $\lambda = 10^{-3}$ throughout; the w-step stays an exact nonnegative least-squares problem, whose nonnegativity keeps w sparse. The weights w are assignment weights rather than estimates of the physical coefficients c_m: the discovered operators' actions enter the fit unnormalized, and w_m tracks the true coefficient magnitude only for advection (Appendix C.3).

Label free model selection. Merging duplicate discovered operators pooled across runs and choosing among runs are based on held out prediction on unlabeled data (Arlot and Celisse, 2010). Structural clustering heuristics fail in this setting (Appendix C.3), while the physical identification metric would use withheld mechanism information and is therefore not used for model selection.

3.4 GROW: EXPANDING THE LIBRARY WHEN RESIDUALS SUGGEST MISSING STRUCTURE

Adaptation cannot introduce a mechanism absent from the initial library: it moves operator codes within the fixed pretrained network, and only for operators that some trajectory already selects. We thus monitor the held out reconstruction residual of Eq. (2): a persistently high residual signals dynamics the current library cannot express. When an operator's assignment weight stays near zero, it is *reseeded*: among the R unlabeled windows ranked worst by held out residual, each candidate window is encoded to initialize a new operator, the operator is refined by the same self supervised objective (Eqs. (2)–(3)), and the restart with the lowest held out residual is kept (best of R). Growth therefore uses only quantities available without labels, and mechanism identities remain unknown until identification. The procedure is dead operator replacement from dictionary learning (Aharon et al., 2006; Mairal et al., 2010) transposed to dynamical operators; the reseeding details and the leave-one-mechanism-out study are in Appendix C.2.

3.5 IDENTIFYING DISCOVERED OPERATORS

A discovered operator is evaluated against an external mechanism only after discovery. The question is if its action on independent probe fields agrees more closely with one reference mechanism than with competent alternatives. We use a fixed probe set $\mathcal{B}$ of synthetic fields, generated once independently of training, and compare each discovered operator's displacement with the one produced by each reference solver, where $\mathcal{S}_m$ denotes one step of the exact reference solver for mechanism m:

$$d_m(\hat{\Phi}) = \underset{p\in\mathcal{B}}{\text{mean}} \frac{\left\| (\hat{\Phi}(p) - p) - (\mathcal{S}_m(p) - p) \right\|_2}{\|\mathcal{S}_m(p) - p\|_2}. \tag{4}$$

Here $d_m(\hat{\Phi})$ measures disagreement between the discovered operator and mechanism m on the probe set, normalized by the magnitude of the reference displacement. Small d_m therefore indicates agreement with that mechanism's action.

We evaluate discovered libraries along three dimensions. *(i) Mechanism identification:* the closest reference mechanism must be preferred by a predefined margin, and the discovered operator's distance to that mechanism must fall below a null threshold set from the distances of operators that do not implement it (Appendix B). *(ii) Selection:* on held out unlabeled mixtures, the discovered library must recover active mechanisms above permutation chance. *(iii) Utility:* the error of rollouts composed from the discovered operators on held-out trajectories, measured in the same simulation harness as the supervised library (lower is better). Mechanism identification and simulation utility are therefore evaluated separately: agreement with a reference mechanism does not by itself imply that the discovered operator is sufficiently accurate for long rollouts.

The probe set, thresholds, and decision criteria are fixed before the corresponding evaluation. We report the resulting identification outcomes together with the selection and utility measurements. This separation keeps the mechanism basis out of discovery while making the physical interpretation of a discovered operator testable against independently specified reference dynamics.

4 EXPERIMENTAL SETUP

Table 5 defines every dataset and library name. Supervised libraries ($\mathcal{L}^{\mathrm{SL}}$) are trained on labeled pure mechanism trajectories. Discovered libraries ($\mathcal{L}^{\mathrm{SSL}}$) are learned from unlabeled mixtures and warm start from a supervised library trained on labeled synthetic primitives, where labels are free by construction (§3.3). We denote by *label-free* the fact that no mechanism labels enter training or model selection, rather the mechanism basis is used only by the post-hoc identification protocol (§3.5). EVAL-9 is a test-only suite of nine held-out named equations, each a two-to-four-term composition of PRIM-1D families with jittered coefficients (the equations and exact forms are in Appendix A.2). 2D experiments and ablations run at 128^2, and the main Navier–Stokes results are repeated at 256^2, the resolution NOS reports.

We use two architectures: the NOS architecture (Serrano et al., 2026) and a second, independently implemented *control architecture*, which tests implementation independence (§5.1). A *supervised operator* is one library entry: the operator of Eq. (1) with its code held in the code-space region

Table 1: Zero-shot composition on EVAL-9 (mean NRMSE over nine equations; one shared beam harness; all models supervised). $\mathcal{L}_8^{\mathrm{SL}}$ rows are mean±sd over 3 seeds; the NOS rows are measured at 3 seeds (Appendix D) and are indistinguishable from the control architecture within run-to-run variance. The monolithic rows roll out one network trained on the same corpus. $\mathcal{L}_2^{\mathrm{SL}}$ and $\mathcal{L}_8^{\mathrm{SL}}$ are supervised libraries trained on two (advection, diffusion) or eight mechanism families. The main observation is that generalizing to new unseen equations is made possible thanks to the library breadth (per-equation results in Table 8; architectures defined in §4).

Architecture	Library	Mean
control architecture	$\mathcal{L}_2^{\mathrm{SL}}$	0.190
control architecture	$\mathcal{L}_8^{\mathrm{SL}}$	0.096 ± 0.001
NOS operator	$\mathcal{L}_8^{\mathrm{SL}}$	0.096 ± 0.005
full NOS method	$\mathcal{L}_8^{\mathrm{SL}}$	0.107 ± 0.009
monolithic FNO, no composition (2.2M)	trained on PRIM-1D	0.117 ± 0.001
monolithic FNO, parameter-matched (16.4M)	trained on PRIM-1D	0.113 ± 0.002

associated with one mechanism. All operators in a library come from a single jointly trained model, one encoder and generator whose code space is carved into per-mechanism regions; no separate network is trained per mechanism (Figure 2). Library instances are per-dimension (1D on PRIM-1D, 2D on NS-2D) and never share weights. We use two versions of the NOS architecture: the *NOS operator*, our reproduction without the vector-quantized codebook and in-context conditioning, which keeps each operator's identity fixed, and the *full NOS method*, with both (Appendix D).

Rollout quality is per-step-mean relative L_2 error (NRMSE) over an autoregressive rollout: 16 steps for EVAL-9 and ABL-2D, and 10 steps for the NS-2D test set, the horizon of NOS's released test trajectories.

5 RESULTS

We first establish the breadth and solver fidelity of the supervised library that discovery builds on (§5.1, §5.2). We then test what unlabeled discovery recovers (§5.3, §5.4), and where its main limitations are (§5.5); transfer to external benchmark data is in Appendix C.5.

5.1 COMPOSITION OVER A BROAD CORPUS ALLOWS GENERALIZATION TO NINE UNSEEN EQUATIONS

Before evaluating discovery, we first establish that the supervised library is broad enough to support composition on equations not seen during training. Table 1 evaluates zero-shot composition on EVAL-9 through one shared beam harness. Growing the corpus from two to eight primitive families divides the error of the control architecture by about two across the nine unseen equations. The NOS recipe libraries reach the same range (0.096 ± 0.005 for the NOS operator and 0.107 ± 0.009 for the full NOS method) with no noticeable differences beyond run variance. Here, corpus breadth is the main contributor to result quality, while the choice of operator architecture has a smaller effect.

A monolithic FNO Li et al. (2020) trained on the same corpus reaches 0.117 ± 0.001, while a parameter-matched monolith (16.4M parameters) reaches 0.113 ± 0.002. Thus, the improvement is not explained by parameter count alone. Rather, composition allows the library to reuse mechanism specific operators across equations. Figure 4 (appendix) shows one composed example, mechanism by mechanism.

5.2 SOLVER FIDELITY DOMINATES LONG HORIZON ACCURACY

Before testing discovery, we check that the operators it starts from are accurate enough. The training data matter most. Replacing the low fidelity ABL-2D corpus with the high fidelity NS-2D corpus reduces one step error from 0.155 to 0.0005, and at matched volume and resolution the generating solver alone accounts for about $200\times$ of this reduction, far more than resolution or training volume (Table 9, Appendix A.5).

Table 2: Discovery on MIX-1D, five hidden mechanisms. *Recovery*: mechanisms identified out of five (§3.5). *Held-out selection*: accuracy of the active mechanisms predicted on unseen mixtures (permutation chance in parentheses). *Utility*: composed rollout error on held-out mixtures (lower is better). — marks a quantity the method does not produce. The full method uses the NOS architecture with its codebook; Table 10 (appendix) reports both architectures, both gradings, and oracle utilities.

Method	Recovery	Held-out selection	Utility
amortized assignment (best of 4 formulations, Appendix C.1)	0/5	—	0.377
SPLICE, single run (Eqs. 2–3)	4/5	57.9% ± 7.7 (51%)	0.147
SPLICE, full (ensemble pooling, predictive K)	**5/5**	**95.0%** (46%)	0.154
PDE-FIND (true symbolic terms given) (Rudy et al., 2017)	5/5	—	0.056
LEADS-style (environment ids given) (Yin et al., 2021)	4/5	—	—

The training curriculum matters less. In this curriculum the operator first learns to predict one step ahead, then gradually longer rollouts of its own predictions, up to 16 steps. It stabilizes 16 step rollouts on ABL-2D (0.309, against 0.965 with one step training and 0.93 for persistence, which copies the last frame forward) but is unnecessary on NS-2D, where one step training is already stable (0.0225, against 0.038 with the curriculum).

With high fidelity data, the 23M parameter NOS operator composes Navier–Stokes zero-shot at parity with the released NOS checkpoint at 128^2 (0.0036 against 0.0045) and reaches 0.0031 ± 0.0004 at 256^2 over three seeds (Table 3). Together, these results show that the supervised operators are accurate once trained on high fidelity data, so the errors reported in the following sections can be attributed to discovery rather than to the operators it starts from.

5.3 DISCOVERY IN 1D

We first test whether unlabeled mixtures are sufficient to recover reusable mechanism operators. Table 2 reports recovery, selection and utility for our procedure and its alternatives. Discovery observes only mixtures and uses the physical mechanism basis only for *post-hoc* identification.

Exact assignment is important for recovery. The exact w step recovers four out of five families in a single run and all five after pooling runs, whereas amortized assignment recovers at most two families per run (Appendix C.1). Operator representation also matters: the same discovery procedure identifies five out of five families with the codebook architecture, compared with three of five with the control architecture. The codebook model recovers the active mechanisms of held out mixtures at 95% accuracy against 46% permutation chance, and identification is unchanged with an independently implemented reference solver (Appendix B.1).

These choices remain label-free during discovery. Cluster count and run selection are based on held-out prediction on unlabeled data, while the physical identification metric is used only for *post-hoc* evaluation (Appendix C.3). The discovered libraries can also be composed into simulators: their composed error stays within about twice that of the exact operators composed the same way, and most of the remaining gap comes from the nonlinear mechanisms (Figure 5, appendix).

Baselines. PDE FIND (Rudy et al., 2017), given the true symbolic terms, approaches oracle performance on clean data, while our operators require no symbolic prior. Under 10% noise, the two approaches reach similar simulation utility (Appendix C.4). LEADS (Yin et al., 2021) identifies four of five primitives even when environment identities are provided, compared with five of five for our exact fit procedure. Its shared component mixes advection and diffusion and misses dispersion, whereas our procedure produces separate, identifiable operators.

5.4 DISCOVERY IN 2D: DECOMPOSING COUPLED NAVIER–STOKES DYNAMICS

We next test whether discovery transfers to 2D dynamics and, more importantly, whether it can separate mechanisms that are never observed in isolation. We use the NOS operator of §5.2 and evaluate identification against independently specified per-family null hypotheses; Table 3 summarizes the results.

Table 3: Zero-shot Navier–Stokes composition (NRMSE on the released test trajectories, 10 steps, lower is better; persistence scores 0.012 at 128^2 and 0.017 at 256^2). *Supervised*: trained on labeled pure-mechanism trajectories. *Label-free*: discovered from the same trajectories with mechanism labels hidden, or from coupled flows where no pure trajectory exists; labels enter only post-hoc identification. Mean $\pm$ sd over seeds, median in parentheses. NOS is used composed, like our libraries; its direct rollout reaches 0.021 at 128^2 and 0.0016 at 256^2 (Appendix D). Full comparison in Table 12.

Training data	Released NOS	Supervised (ours)	Label-free (ours)	Identified
pure mechanisms, 128^2	0.0045	0.0036 ± 0.0006	0.0038 ± 0.0001	all seeds
pure mechanisms, 256^2	0.0056	0.0031 ± 0.0004 (0.0033)	0.0028 ± 0.0025 (0.0016)	10/10
coupled flows, 256^2	n/a	n/a (no labeled data)	0.0116 ± 0.0014 (0.0118)	10/10

On a mixture of 128 Euler and 128 diffusion trajectories with identities hidden, the discovered library composes Navier–Stokes at essentially the same accuracy as the supervised library, 0.0038 versus 0.0036. At native 256^2, the full discovery loop identifies both mechanisms on $10/10$ seeds and composes at 0.0028 ± 0.0025 (median 0.0016), against 0.0031 ± 0.0004 (median 0.0033) for the supervised library.

The more demanding test uses coupled Navier–Stokes trajectories, where Euler and diffusion act simultaneously at every step and no pure mechanism trajectories are available. Discovery recovers both mechanisms on all three seeds. At native 256^2, discovery on coupled flows, warm started from the supervised library, identifies both mechanisms on all ten seeds and composes the coupled dynamics at 0.0116 ± 0.0014 (median 0.0118).

The same adaptation applies to real data. On `shear_flow` from The Well (Ohana et al., 2024), adapting the library on 32 unlabeled realizations, disjoint from the evaluated ones, makes its operators identify as Euler on three of three seeds, where the unadapted library missed, and lowers the composed transfer error (VRMSE) from 0.868 to 0.818 on the same base model, more than twice the seed spread, though short of the pre-registered target (Appendix C.5).

Figure 3 shows one composed rollout. The beam selects only the advection operators, although the mixture assignment attributes activity to both mechanisms: an operator can implement an identifiable mechanism without improving a particular composition (§5.5).

The same experiments show that discovery extends a pretrained library rather than reliably creating one from scratch. Random initialization collapses onto diffusion, while reseeding a low-weight operator recovers the missing Euler family on all three tested 2D seeds. We treat this as supporting evidence for library growth; detailed reseeding and leave-one-mechanism-out experiments are reported in Appendix C.

Taken together, removing the labels loses nothing relative to the supervised NOS operators. The discovered library identifies Euler and diffusion, composes Navier–Stokes as accurately as its supervised counterpart and more accurately than the released NOS checkpoint, and recovers the same mechanisms from coupled flows, where no labeled data can exist.

5.5 Limits

The experiments also reveal four challenges for the approach.

Mechanism separability. Identification depends on different mechanisms producing sufficiently distinct actions. In 2D, diffusion, hyperdiffusion, and drag act almost identically, and drag aliases to diffusion in coupled discovery on all three tested seeds, whereas Euler, whose action is distinct, is recovered from coupled flows. Similar mechanisms therefore share one library slot (Appendix B).

Composability. Physical identity does not guarantee that an operator improves a particular composition. On the flow of Figure 3, forcing the identified diffusion operator into the composition increases the rollout error from 0.24 to 0.68; the effect depends on how the operator enters the composition, not on its accuracy as diffusion (Appendix C.3).

Operator representation. The operators standardize each input window by its mean and scale, so mechanism identity is recovered while absolute coefficient magnitude is not, which makes amplitude dependent mechanisms less natural for the current representation (Appendix C.3).

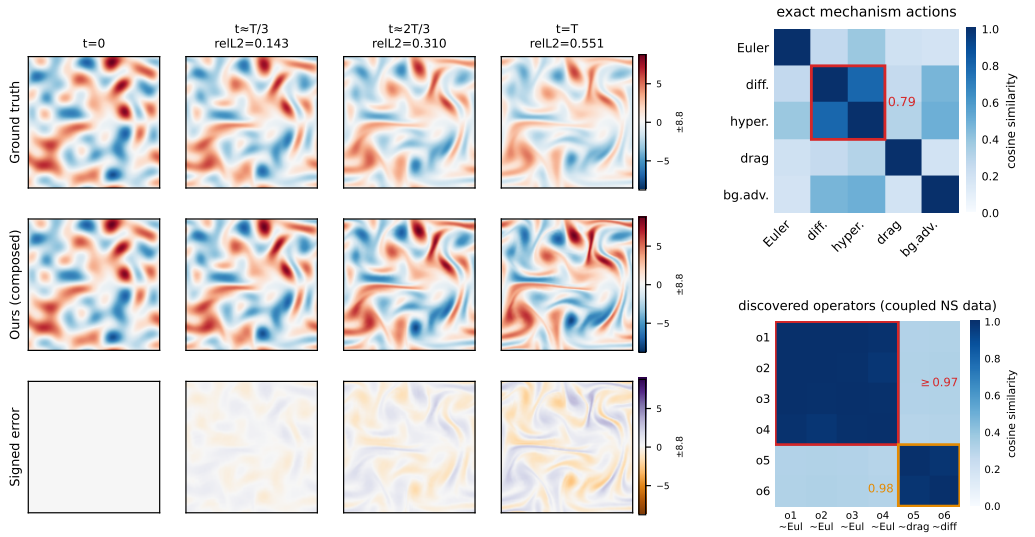

Figure 3: Zero-shot composition with a label-free library, and why Euler is identifiable. Left: exact solution (top), rollout composed from discovered operators (middle) and their difference (bottom), at four times of a 60-step rollout on a coupled flow never seen in training (128^2). Reading each column from top to bottom, the positions and shapes of the vortices match throughout, and the error grows mainly in intensity. The true flow fades because viscosity and drag dissipate its energy; the composed rollout does not, because the beam selects only advection operators and each operator normalizes the amplitude of its input window, so this dissipation is not represented (Appendix C.3). **Right:** similarity between mechanism actions, for exact (top) and discovered (bottom) operators. Euler is nearly orthogonal to the overlapping smoothing mechanisms, hence identifiable (§5.5).

Splitting regime. Finally, composition assumes that mechanisms can be applied one after the other with small error. In developed turbulence they interact too strongly for this: on turbulent flows from The Well, the beam finds no useful composition, and the rollout stays sharper than the true flow, which smooths out over time (Appendix C.5). A learned operator for the interaction between mechanisms is a natural extension. Without fine-tuning, the 2D library composes `shear_flow` stably where the beam selects diffusion, diverges where it selects Euler, and declines to compose `rayleigh_benard`, whose buoyancy and walls lie outside the library (Appendix C.5).

6 Discussion and Conclusion

We found that the operator splitting of physics can be learned from unlabeled trajectories and checked without labels. SPLICE adapts a library pretrained on synthetic single-mechanism data, where labels are free, to unlabeled data, and names the mechanism each operator implements, or reports that it cannot. Hidden mechanisms of 1D mixtures are recovered, the Euler and diffusion operators of Navier–Stokes are identified from coupled flows in which neither ever acts alone, and a missing mechanism grows back from its own residuals. Learning without labels costs no accuracy and often improves it: at 256^2, the adapted library composes more accurately than the supervised library it starts from on seven of ten seeds, halving its error on the median run, and more accurately than the released NOS model.

Because every component is identified, a failing simulation points to the operator responsible, and the remaining limits, such as mechanisms with nearly identical actions sharing one operator, can be measured directly (§5.5).

This suggests a way to build physics simulators: pretrain a library where labels are free, adapt it where they are not, and compose from identified parts. Next steps include adapting the network during discovery, composition-aware training, and other modalities, where labels are even scarcer. After pretraining, mechanism labels can simply be removed, and reference solvers suffice to tell what each operator learned.

Reproducibility statement

Appendix B specifies the identification protocol in full; the hyperparameter tables and reproducibility notes are in the appendix, and every printed number traces to a logged artifact shipped with the supplementary material, alongside a pre-registration bundle of six records whose manifest states, for each, the criterion written before grading and whether it was met. Code at `https://anonymous.4open.science/r/SPLICE_ICLR-EBC8`

AI use statement

Large language model assistants were used for drafting and editing text, writing experiment and analysis code, and orchestrating experiment runs. All quantitative claims trace to code; the authors checked every reported number against it and reviewed and take responsibility for all content.

References

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11): 4311–4322, 2006.

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024.

Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. 2010.

Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. Deepmod: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, 2021.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Steven L Brunton, Marko Budišić, Eurika Kaiser, and J Nathan Kutz. Modern koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.

Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf`.

Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, page 17429–17442. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/c9f2f917078bd2db12f23c3b413d9cba-Paper.pdf`.

Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings. Mathematical, Physical, and Engineering Sciences*, 478(2260):20210904, 2022.

Quentin Garrido, Laurent Najman, and Yann Lecun. Self-supervised learning of split invariant equivariant representations. *arXiv preprint arXiv:2302.10283*, 2023.

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. *arXiv preprint arXiv:2403.03542*, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Armand Kassaï Koupaï, Jorge Mifsut Benet, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Boosting generalization in parametric pde neural solvers through adaptive conditioning. *Advances in Neural Information Processing Systems*, 37:70659–70692, 2024.

Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. In *International conference on machine learning*, pages 11283–11301. PMLR, 2022.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Science*, 118(21):e2101784118, May 2021. doi: 10.1073/pnas.2101784118.

Nodens Koren and Samuel Lanthaler. Merging memory and space: A state space neural operator. *arXiv preprint arXiv:2507.23428*, 2025.

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. URL http://jmlr.org/papers/v24/21-1524.html.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 914–922. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/linderman17a.html.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4114–4124. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR, 2018.

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

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of machine learning research*, 11(1), 2010.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

Jorge A Mendez and ERIC EATON. Lifelong learning of compositional structures. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ADWd4TJO13G.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B Dalziel, Drummond B Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4036–4044. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/parascandolo18a.html.

Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning Mesh-Based Simulation with Graph Networks. *arXiv e-prints*, art. arXiv:2010.03409, October 2020. doi: 10.48550/arXiv.2010.03409.

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. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3(4):e1602614, 2017.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Louis Serrano, Armand Kassaï Koupaï, Thomas X Wang, Pierre Erbacher, and Patrick Gallinari. Zebra: In-context generative pretraining for solving parametric pdes. *arXiv preprint arXiv:2410.03437*, 2024.

Louis Serrano, Jiequn Han, Edouard Oyallon, Shirley Ho, and Rudy Morel. Test-time generalization for physics through neural operator splitting. *arXiv preprint arXiv:2602.00884*, 2026.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

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, editors, *Advances in Neural Information Processing Systems*, volume 35, page 1596–1611. Curran Associates, Inc., 2022. doi: 10.52202/068431-0117. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0a9747136d411fb83f0cf81820d44afb-Paper-Datasets_and_Benchmarks.pdf.

Tapas Tripura and Souvik Chakraborty. A foundational neural operator that continuously learns without forgetting. *arXiv e-prints*, art. arXiv:2310.18885, October 2023. doi: 10.48550/arXiv.2310.18885.

Rui Wang, Robin Walters, and Rose Yu. Meta-learning dynamics forecasting using task inference. *Advances in Neural Information Processing Systems*, 35:21640–21653, 2022.

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), Sept 2023. ISSN 1091-6490. doi: 10.1073/pnas.2310142120. URL http://dx.doi.org/10.1073/pnas.2310142120.

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

Jinpai Zhao, Nishant Panda, Yen Ting Lin, Eirik Valseth, Diane Oyen, and Clint Dawson. Hycop: Hybrid composition operators for interpretable learning of pdes, 2026. URL https://arxiv.org/abs/2605.00820.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-conditional transformers towards universal neural PDE solvers. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 79061–79088. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25r.html.

A Experimental Details

A.1 Notation

Table 4 collects the symbols used in the paper; dataset and library names are in Table 5.

Table 4: Symbols (top) and architectures with their instances (bottom).

Symbol	Meaning
$u_t \in \mathbb{R}^{C\times N}$ (1D), $\mathbb{R}^{C\times N\times N}$ (2D)	discretized field at step t; C channels, N grid points/axis
$u_{0:T}$	a trajectory; $u_{t-L:t}$ a length-L context window
A_m, $c_m \geq 0$	mechanism m (diffusion, advection, …) and its coefficient
E	encoder: context window to latent code
$\theta = E(u_{t-L:t}) \in \mathbb{R}^d$	latent code (selects mechanism and coefficient); d its dimension
f_θ	conditioned vector field; continuous dynamics $\mathrm{d}v/\mathrm{d}s = f_\theta(v)$
$\Phi_\theta(u_t) = \mathrm{RK4}[f_\theta](u_t)$	one solver-operator step (Eq. 1)
$\Phi_A^{1/2} \circ \Phi_B \circ \Phi_A^{1/2}$	one Strang-composed step; local error $O(\Delta t^3)$
$\mathcal{L} = \{\Phi_{\theta_1}, \ldots, \Phi_{\theta_M}\}$	operator library (M mechanisms)
$\hat{\Phi}_k$, $\hat{f}_k$	discovered operator k and its vector field; $\hat{\phi}$ their codes
$w \in \mathbb{R}^K_{\geq 0}$	per-trajectory mixture weights over discovered operators (w-step, Eq. 2)
z_t; $\Delta z_t = z_{t+1} - z_t$	standardized frame at time t; finite-difference estimate of $\partial_t u$
$\hat{\Phi}^{(h)}_{\hat{\phi},w}$	h-step rollout of the weight-mixed discovered field (ϕ-step, Eq. 3)
λ	weight of the total-weight penalty in the w-step
K	number of discovered operators kept (cluster count for merging pooled runs)
$\mathcal{B}$; $\mathcal{S}_m$	probe set of fixed synthetic fields; exact reference solver step for mechanism m (App. B)
NRMSE	per-step-mean relative L_2 error over a 16-step rollout
Architecture / instance	**Meaning**
NOS architecture	Serrano et al. (2026) architecture (instantiated 1D and 2D)
control architecture	our independent second architecture (instantiated 1D and 2D)
NOS operator	NOS architecture without codebook and in-context conditioning (our reproduction)
full NOS method	NOS architecture with codebook, in-context, and substep ramp
2D instance	trained on NS-2D; source of all NS numbers
1D instance	trained on PRIM-1D; source of all breadth numbers

A.2 Datasets and mechanisms

Each PRIM-1D mechanism is one differential operator acting on the field u: *advection* $-c\,\partial_x u$ transports the field at speed c; *diffusion* $\nu\,\partial_{xx} u$ smooths it; *dispersion* $-\partial_{xxx} u$ makes phase speed depend on wavenumber; *nonlinear self-advection* $-u\,\partial_x u$ and *cubic advection* $-u^2\,\partial_x u$ steepen the field toward shocks; *decay* $-\gamma u$ damps it uniformly; *hyperdiffusion* $-\nu_4\,\partial_{xxxx} u$ damps high wavenumbers more sharply than diffusion; and *reaction* $\rho\,u(1-u)$ (bistable form $\rho\,u(1-u^2)$) grows or saturates the field pointwise. In 2D the *Euler* operator advects vorticity by its own induced velocity. Table 6 gives the exact form of each EVAL-9 equation.

A.3 Training configurations and architecture

Table 7 lists the generation and training settings; the shared architecture is shown in Figure 2.

Table 5: Datasets (top) and operator libraries (bottom). "Labels"=mechanism identity per trajectory.

Name	Content	Role	Labels
PRIM-1D	1D, 8 mechanisms × dense coefficients (spectral solver): advection, diffusion, dispersion, nonlinear advection, decay, hyperdiffusion, cubic advection, bistable reaction	SL training	yes
MIX-1D	1D, hidden random 1–2-mechanism mixtures of 5 PRIM-1D mechanisms (advection, diffusion, dispersion, nonlinear advection, decay)	SSL training	hidden (grading only)
EVAL-9	1D, nine named PDEs (Table 6), each a 2- to 4-term composition of PRIM-1D mechanisms with jittered coefficients	zero-shot test	—
NS-2D	2D vorticity: pure Euler and pure diffusion training trajectories (2048 traj.) + NS test set (3072 traj.); public release of Serrano et al., 2026	2D SL training / test	yes / —
ABL-2D	2D vorticity, self-generated, lower-fidelity solver (256 trajectories)	*data ablation only* (§5.2)	yes
$\mathcal{L}_2^{\mathrm{SL}}$	supervised library, 2 mechanisms (NOS scale)	baseline	yes
$\mathcal{L}_8^{\mathrm{SL}}$	supervised library, 8 mechanisms (trained on PRIM-1D)	ours	yes
$\mathcal{L}^{\mathrm{SSL}}$	discovered library (trained on MIX-1D or unlabeled NS-2D mixtures)	ours	none

Table 6: The nine EVAL-9 equations, each a composition of PRIM-1D mechanisms. Free coefficients are jittered within the ranges of Table 7; the dispersion coefficient is fixed to unity. Signs and the exact reaction terms follow the data generators.

Equation	PDE	Active mechanisms
Burgers	$\partial_t u = -u\,\partial_x u + \nu\,\partial_{xx} u$	nonlinear advection, diffusion
KdV	$\partial_t u = -u\,\partial_x u - \partial_{xxx} u$	nonlinear advection, dispersion
mKdV	$\partial_t u = -u^2\,\partial_x u - \partial_{xxx} u$	cubic advection, dispersion
Allen–Cahn	$\partial_t u = \nu\,\partial_{xx} u + \rho\,u(1-u^2)$	diffusion, bistable reaction
advection+reaction	$\partial_t u = -c\,\partial_x u + \rho\,u(1-u)$	advection, reaction
damped KdV	$\partial_t u = -u\,\partial_x u - \partial_{xxx} u - \gamma u$	nonlinear advection, dispersion, decay
hyperviscous KdV	$\partial_t u = -u\,\partial_x u - \partial_{xxx} u - \nu_4\,\partial_{xxxx} u$	nonlinear advection, dispersion, hyperdiffusion
KdV–Burgers–decay	$\partial_t u = -u\,\partial_x u + \nu\,\partial_{xx} u - \partial_{xxx} u - \gamma u$	nonlin. advection, diffusion, dispersion, decay
adv.–diff.–decay–hyperdiff.	$\partial_t u = -c\,\partial_x u + \nu\,\partial_{xx} u - \gamma u - \nu_4\,\partial_{xxxx} u$	advection, diffusion, decay, hyperdiffusion

Table 7: Data generation and training configurations. 1D: spectral solver, five discovery mechanisms shown (the full PRIM-1D corpus has eight). 2D: pseudo-spectral vorticity solver; data generated at 256^2 and trained at 128^2. Parameter counts measured from checkpoints.

	Setting	Value
1D	grid, steps per trajectory	n_x=256, 100 steps (L=16, T=4, Δt=0.04)
	coefficient ranges	advection $[0.1, 2.0]$, diffusion $[0.01, 0.4]$, dispersion $[0.1, 1.0]$, nonlinear advection $[0.1, 0.5]$, decay $[0.1, 1.0]$
	training	10^5 iterations, batch 32, lr 3×10^{-4} (cosine), weight decay 10^{-4}
	model	16.30M parameters
2D	grid, snapshots	128^2 (generated at 256^2), 50 snapshots (T=4, Δt=4/250)
	coefficient ranges	diffusion $\nu \in [10^{-3}, 10^{-2}]$, hyperdiffusion $\nu_4 \in [10^{-4}, 10^{-3}]$, drag $[0.1, 0.6]$, background advection $U \in [0.5, 1.5]$, Euler (no free coeff.)
	training	4×10^4 iterations, batch 8, lr 3×10^{-4} (warmup 10^3), 4 integration substeps, rollout-horizon ramp $1 \to 16$ (parity); $1 \to 8$ (breadth-suite operator)
	corpus	1024 Euler + 1024 diffusion trajectories; released NS test set (native 256^2)
	model	23.61M parameters

A.4 EVALUATION SETTINGS AND PER-EQUATION RESULTS

Table 8 breaks the EVAL-9 mean of Table 1 down per equation for the NOS-operator $\mathcal{L}_8^{\mathrm{SL}}$ library (three seeds, re-measured; Appendix D). The error is spread across equations (0.048 to 0.123), and the four-term systems (0.107, 0.067) are no harder than the two-term ones. Figure 4 shows one composed example and Figure 5 three held-out discovery rollouts.

Table 8: Per-equation composed NRMSE on EVAL-9 (NOS operator, $\mathcal{L}_8^{\mathrm{SL}}$; mean over 3 seeds, re-measured 2026-09, Appendix D).

Equation	NRMSE	Equation	NRMSE
Burgers	0.091	damped KdV	0.112
KdV	0.123	hyperviscous KdV	0.123
mKdV	0.048	KdV-Burgers-decay (4-term)	0.107
Allen-Cahn	0.074	adv.-diff.-decay-hyperdiff. (4-term)	0.067
advection+reaction	0.115	**mean**	**0.096**

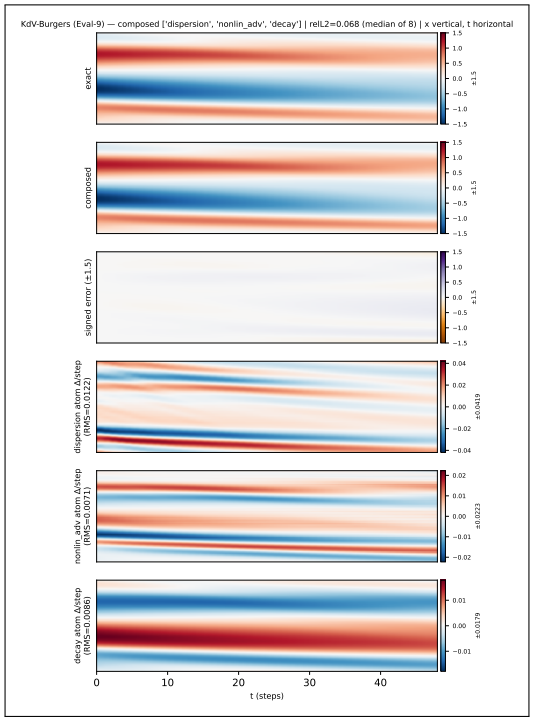

Figure 4: Zero-shot composition of the four-term EVAL-9 equation KdV-Burgers with decay (nonlinear advection, dispersion, diffusion, decay) by a five-family supervised library (space-time; the upper-median of 8 test trajectories by composed rollout error, relative L_2 0.068, set median 0.065). Rows: exact solution; composed rollout; signed error; per-operator increments on their own scales. Compositions are capped at three operators, so one of the four terms must be dropped; the beam keeps dispersion, nonlinear advection, and decay and drops diffusion, a dissipative-for-dissipative substitution consistent with the separability limits of §5.5. The nonlinear operator carries the steepening.

A.5 SUPERVISED OPERATOR FIDELITY ABLATIONS

Table 9 collects the Navier–Stokes ablations behind §5.2. As a reading aid for rollout NRMSE: below 0.1 is usable, above 0.3 is not. A monolithic FNO trained with the curriculum on the same 2048 pure-mechanism trajectories does not beat persistence on the released NS-2D slice (0.031 over 16 steps, best of a sweep) and ties our operator on the dynamic ABL-2D slice (0.31 each); we therefore make no 2D claim of composition over monoliths. In the one-step against curriculum

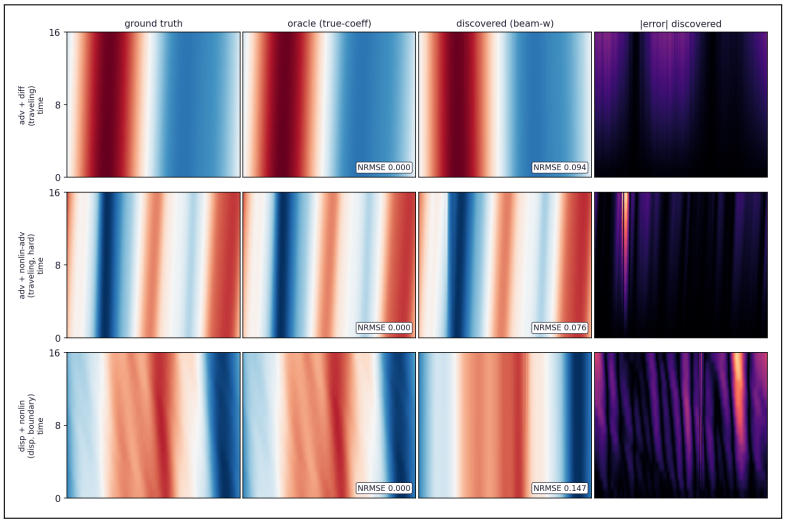

Figure 5: Rollouts on three held-out 1D mixtures (§5.3): ground truth; oracle operators; discovered library $\mathcal{L}^{\mathrm{SSL}}$; absolute error (shared scale). Top: advection+diffusion, cleanly reproduced (0.094). Middle: a traveling advection+nonlinear case, where the discovered library reproduces the transport (0.076). Bottom: an advection+dispersion case at the dispersion-limited boundary (0.147): the discovered rollout cannot travel because the residual transport there is dispersion-driven and dispersion is the rarest mechanism in the discovered pool (Appendix C.3).

comparison, batch sizes are not perfectly matched, because the 16-frame curriculum does not fit in memory at 256^2.

Table 9: Navier–Stokes quality ablations (per-step-mean NRMSE, 128^2). ABL-2D rows: 16-step rollouts, persistence 0.93. NS-2D data-control rows: released test trajectories over the 10-step released horizon (persistence 0.012); the NS-2D pushforward rows in the objective control are 16-step rollouts on the same trajectories. Left: training-objective control on ABL-2D and, below the rule, on NS-2D (both objectives from scratch with the pushforward trick). Right: data control and, below the rule, its split into volume, resolution, and solver at a shorter matched budget (hence the 0.0031 anchor in that block). All models supervised.

Objective control	rollout	Data control	rollout	one-step
ABL-2D: persistence (floor)	0.93	small self-generated corpus (ABL-2D)	0.31	0.155
ABL-2D: one-step training	0.965 (diverges)	NOS corpus (NS-2D)	**0.0034**	0.0005
ABL-2D: + best post-hoc stabilizer	0.267	NOS model, same protocol	0.021	—
ABL-2D: rollout curriculum	0.309 (stable)	*disentanglement (matched budget):*		
NS-2D: one-step training (pushforward)	0.0225 (stable)	NS-2D, full volume, 128^2	—	0.0031
NS-2D: rollout curriculum (pushforward)	0.038	NS-2D, 256 traj. (volume matched), 128^2	—	0.0007
		NS-2D, full volume, 64^2 (resolution)	—	0.0071
		ABL-2D solver, 256 traj., 128^2 (solver)	—	0.155

Stabilizers and corpus levers. On ABL-2D, the best post-hoc stabilizer of one-step-trained operators reaches 0.267 (other variants 0.341 to 0.434), none reaching the usability bar. At fixed architecture and recipe, the generating solver moves one-step error about 200×, volume contributes nothing measurable down to 256 trajectories, and resolution about 2× per octave (Table 9).

1D head-to-head. The EVAL-9 comparison of Table 1 uses one shared beam harness with an identical search budget for both architectures, with a cap on per-context search so that neither is advantaged by longer test-time compute. Across seeds, the identification distance of exact-fit discovered operators varies about 50× less than that of amortized-selection discovered operators.

B PHYSICAL IDENTITY EVALUATION PROTOCOL

This appendix details criterion (i) of §3.5: the probe set, the identification distance, its margin, the per-mechanism null threshold, and the absolute cap.

B.1 PROBE SET, DISTANCE, AND THRESHOLDS

Probe set. All identification uses one fixed set of synthetic input fields, the probe set $\mathcal{B}$, built once, deterministically, and independent of any training data, trajectory, or label: (i) pure Fourier modes $\cos(kx), \sin(kx)$ over a fixed range of wavenumbers; (ii) localized Gaussian bumps at fixed widths and positions; (iii) random band-limited fields from a fixed seed.

Identification distance. A discovered operator $\hat{\Phi}$ is graded by Eq. (4): its one-step displacement $\hat{\Phi}(p) - p$ on each probe, against the displacement of each exact solver under the same integrator and time scale, with the probe interval pinned to each operator's θ-implied step. Normalizing by the displacement rather than the full response removes the input state, which otherwise dominates the distance for small-step operators. A discovered operator is assigned to A_m only if m is its closest mechanism by a margin (the ratio of second-closest to closest distance; recovery is unchanged for ratios 1.0 to 2.0) and $d_m(\hat{\Phi})$ falls below that mechanism's null threshold.

2D grading. In 2D, each discovery seed is graded on its own probe set of three random band-limited vorticity fields (25 Fourier modes) drawn from a seed-specific generator. The distance compares four-frame rollouts from each probe with those of the exact solver, normalized by the root-mean-square of the reference rollout, and takes the minimum over eight candidate time scales (and, for diffusion, over a grid of viscosities). The null is the 5th percentile of the same distance over 24 random codes, drawn at the scale used to initialize discovered codes and passed through the seed's own trained network, so both sides use the same estimator and the same network.

Null calibration. Thresholds are set per mechanism from a null of d_m values produced by trained discovered operators identified with the other mechanisms, so the null represents a trained operator of the wrong kind, not noise; a discovered operator identifies as A_m only below the 5th percentile of that null. Per-mechanism thresholds are needed because mechanisms differ widely in natural displacement size. An absolute cap ($d_m < 1.5$) also rejects diverged discovered operators that would otherwise identify by being the closest of uniformly poor fits.

Held-out selection. For each held-out trajectory the w-step (Eq. 2) yields a support pattern over discovered operators, compared to the true active set. Chance comes from a permutation null (shuffled trajectory-label pairings, 10^3 permutations); Table 2 reports the permutation mean in parentheses.

Pre-registration and seeds. Every model-selection criterion was committed in writing before its result was read; §5.3 reports both the criteria that held and those that failed (eigengap, silhouette). Headline recovery claims need two of three seeds.

Independent solver check. Repeating identification with an independently implemented finite-difference reference solver in place of the spectral one leaves operator identity, identification outcomes, and codebook recovery unchanged ($5/5$ in each case), with a maximum change in self-distance of 0.18 against a null distance near 1.0.

B.2 ORIGINAL AND DISCRIMINATIVE GRADING; TWO ARCHITECTURES

An earlier version of the distance normalized by the full response $\|\mathcal{S}_m(p)\|_2$ instead of the displacement. For small-step operators the probe p then dominates, so near-identity operators score about 0.01 against every mechanism and cannot be told apart. We therefore grade by the displacement, which isolates what an operator does rather than the state it receives; this discriminative distance (Eq. (4)) is used for all results. Because this correction followed the first results, the "orig." column of Table 10 keeps the original grading. The correction does not favor our method by construction: it

lowers the control architecture from $5/5$ to $3/5$, held-out selection, which does not use the distance, agrees with it (95% against 46% chance), and every later pre-registered experiment uses it.

The two gradings order the two architectures differently (Table 10). Under the original grading, pooled discovery identifies all five mechanisms with the control architecture and only decay with the NOS codebook architecture, whose small per-step displacements fall inside the null. Under the discriminative grading the codebook architecture identifies $5/5$ and the control architecture $3/5$, and held-out selection agrees (95.0% against 46% chance, and 72.5% against 61%). The control architecture's two misses are systematic (discovered diffusion operators assigned to decay, nonlinear-advection operators to dispersion) and persist for margin ratios 1.0 to 2.0; a stricter margin lowers both counts by one (codebook $4/5$, control $2/5$) without changing the ordering. Accuracy is comparable across the two: per-mechanism one-step error has median 0.010 for both 1D NOS-architecture operators, and the retrained full NOS composes NS at 0.0053 (Table 12).

B.3 Identifiability and separability

Drop-one identifiability. Before asking a selection rule to recover a mechanism, we ask whether the data can reveal it: we roll the exact solver with one true mechanism removed and report the NRMSE of that rollout against the full equation. A mechanism is identifiable at a horizon if this change exceeds three times the identified operator's rollout error. On a five-equation 2D suite (Euler advection, diffusion, hyperdiffusion, drag, and advection by a fixed background field; identifiable if the change is ≥ 0.15), every advective mechanism is identifiable (drop-one change 0.71 to 1.15), while diffusion is identifiable in three of the equations only after raising its viscosity (0.07 to 0.14 before, 0.19 to 0.27 after). On coupled NS, dropping Euler changes the 16-step rollout by 0.88. On Eval-9, 18 of 21 active mechanisms are identifiable at 16 steps (hyperdiffusion and reaction are not audited); the three below threshold are weak short-horizon terms (diffusion in KdV and mKdV, decay in damped KdV), so the $9/9$ mechanism-set result of Table 1 rests on the dominant terms.

Composing the 2D suite. None of the five suite equations composes to the NRMSE < 0.1 bar at 128^2 (0.32 to 0.54, against persistence 0.57 to 1.14). Standalone, supervised diffusion (0.061) and hyperdiffusion (0.092) pass, drag is borderline (0.118), and Euler (0.62; 0.43 when trained on NS-2D) and background advection (0.25) fail: pure advective transport compounds phase error over 16 steps at this scale. A pre-registered dissipative-only system (diffusion + hyperdiffusion + drag) also fails (0.358, persistence 0.381), with diffusion never selected.

Aliasing. With Euler, diffusion, and drag all active in coupled discovery, Euler and diffusion identify on $3/3$ seeds and drag on $0/3$, its discovered operators graded as diffusion. Horizon-resolved identification at $H \in \{4, 8, 16, 32\}$ changes no supervised operator's mechanism: hyperdiffusion and drag are graded diffusion throughout. Across a viscosity sweep of the coupled pair ($\nu \in \{2, 5, 9\} \times 10^{-3}$), Euler and diffusion identify at every ν (margins $+0.043$ and $+0.015$). A fresh joint retraining (three seeds) replicates the aliasing. The dissipative side offers one identifiable slot, and the advective side identifies only through coupling.

Scope of the architectures. Both architectures standardize each input window by its mean and standard deviation before the vector field acts, so a fixed discovered operator represents an operator only up to the window's mean and scale. Shift- and scale-equivariant differential operators (advection, diffusion, hyperdiffusion, dispersion, Euler) are represented exactly, and linear decay is exact on zero-mean fields. Amplitude-dependent reaction terms such as $\rho\, u(1-u)$ are represented only through the per-window code, which is why coefficient magnitude is not recovered (Appendix C.3).

C Discovery and Growth Ablations

C.1 Design of the discovery procedure

Exact versus amortized assignment. We compared the exact w-step of Eq. (2) with four amortized alternatives. Three are selection heads over a frozen dictionary (softmax, Gumbel-softmax, sparsemax); their recovery equals the dictionary's by construction, and the best head reaches 40.3% held-out selection against 44.5% (deduplicated) and 39.5% (single run) for the exact w-step, at a

permutation chance of 20.9%. The fourth, an end-to-end encoder that predicts mixture weights and learns the discovered operators jointly, plateaus at two of five recovered mechanisms: selector error and operator error compensate each other, the loss decreases, and no discovered operator becomes sharp under the identification distance. Replacing amortized inference with the exact w-step lifts recovery to four of five. (The joint-encoder checkpoints predate the discriminative distance, so Table 10 reports its recovery under the original grading only.)

Table 10: Complete discovery table on MIX-1D (5 hidden families), expanding Table 2: recovery under both the original (orig.) and the discriminative grading (Appendix B.2), held-out selection accuracy on unseen mixtures (permutation chance in parentheses), and utility through the shared composition harness against each architecture's oracle. The LEADS rows use a four-environment variant built for LEADS (identities given to LEADS only); — marks cells undefined for a method's output and n.m. cells not measured, and the amortized row's selection is omitted as incomparable (fixed top-2 versus variable-cardinality sets).

Method	Architecture	Recovery orig.	Recovery discr.	Held-out selection	Utility	Oracle	Labels
amortized (best of 4 formulations, Appendix C.1)	control	2/5	0/5	—	0.377	0.063	none
exact-fit (Eqs. 2–3), single run	control	4/5	4/5	57.9% ± 7.7 (51%)	0.147	0.063	none
+ ensemble pooling + predictive K	control	5/5	3/5	72.5% (61%)	0.124	0.063	none
+ ensemble pooling + predictive K	NOS + codebook	1/5	**5/5**	**95.0%** (46%)	0.154	0.077	none
PDE-FIND (true terms in candidate basis)	—	5/5	5/5	—	0.056	—	symbolic prior
LEADS-style shared field + per-environment residual (Yin et al., 2021)	control	—	4/5	—	—	—	environment ids
exact-fit, same four-environment variant	control	n.m.	5/5	n.m.	n.m.	n.m.	none
Where each construction can run				*supervised*	*label-free*		
NS composed zero-shot, 128^2				0.0036 ± 0.0006	0.0038 ± 0.0001 (parity)		
NS composed, native 256^2				0.0031 ± 0.0004 (median 0.0033)	0.0028 ± 0.0025 (median 0.0016), $10/10$ identified		
operators from coupled flows (§5.4)				n/a: no labeled data exists	$10/10$ identified; 0.0116 (native 256^2)		
regrowing a missing mechanism (§5.4)				n/a	$3/3$ (supporting)		
real data, `shear_flow` adaptation (§C.5)				n/a	Euler identified $3/3$; $0.868 \to 0.818$		

Warm start versus random initialization. On the 2D mixture of §5.4 (128 pure-Euler and 128 pure-diffusion trajectories, identities hidden), warm-started discovery recovers both mechanisms on seeds 0 and 1 (Euler fit 0.012 to 0.028), while random initialization recovers $1/2$ and $0/2$: its operators collapse onto diffusion and Euler is never recovered.

Fixed codes versus per-window refitting. Per-window code refitting, the in-context alternative to composing with identified codes locked, does not help: 0.634 refitted against 0.616 locked on the dynamic slice.

C.2 LIBRARY GROWTH BY RESEEDING

A discovered operator is dead when its held-out weight stays near zero (a duplicate, or near the identity), and the ϕ-step never moves an operator that no trajectory selects. Reseeding re-initializes it by encoding a high-residual window, the slice of unlabeled data the current library reconstructs worst under Eq. (2), so the new operator is planted where the error lives. This is the empty-cluster restart of k-means and the dead-element replacement of dictionary learning (Aharon et al., 2006; Mairal et al., 2010), transposed to operators. Keeping the best of R candidates from the R highest-residual windows removes the dependence on a single window; in §5.4, a missing mechanism never emerges without reseeding and emerges on three of three seeds with best-of-R. Dispersion is the exception: a pre-registered variant that scores windows by the odd-derivative fraction of the residual fails on three of three seeds, as does the total-residual control, with fresh discovered dispersion operators diverging. The bottleneck is training a stable discovered dispersion operator, not selecting windows.

Leave-one-family-out regrowth (negative). We also tested regrowth when the backbone has never seen the target family: a backbone pretrained from scratch with the Euler family excluded, followed by the same pre-registered best-of-3 regrowth. On seed 0 of the three registered seeds, regrowth recovered diffusion only; no regrown operator was identified as Euler. The remaining two seeds each need a new backbone pretraining and are not reported. A pre-registered follow-up places the cause in the backbone rather than in discovery: on the same Euler-excluded network, a single code fitted directly on pure Euler windows, with labels and discovery's optimization budget, still identifies as diffusion ($d_{\mathrm{Euler}} = 0.050$ against an Euler null of 0.038; nearest mechanism diffusion at 0.033). Since discovery adapts operator codes within the pretrained network, it reaches only dynamics that network already represents; adapting the network itself is the natural next test for mechanisms absent from pretraining.

C.3 THE DISCOVERED OPERATORS IN 1D AND 2D

Distribution and coefficient recovery. Figure 6 shows the discovered five-mechanism MIX-1D library in operator-signature space (a t-SNE of each discovered operator's action on the probe set): the discovered operators cluster by mechanism around the exact operators. The pool is uneven: 14 diffusion and 11 advection operators against a single dispersion operator. Dispersion is never learned on the harder eight-mechanism basis (§5.5). Coefficients are recovered far less reliably than identity: the deployed assignment weight correlates with the true mixture coefficient only for advection (r=0.82) and is near zero elsewhere (pooled r=0.17). Three analysis-level corrections raise the pooled correlation only to 0.275, and a coefficient-equivariant training variant in the spirit of Garrido et al. (2023) does not help. The cause is not mixture identifiability: on held-out pure single-mechanism trajectories the weight again tracks the coefficient only for advection (r=0.93), because the operator standardizes each window by mean and scale (Appendix B.3).

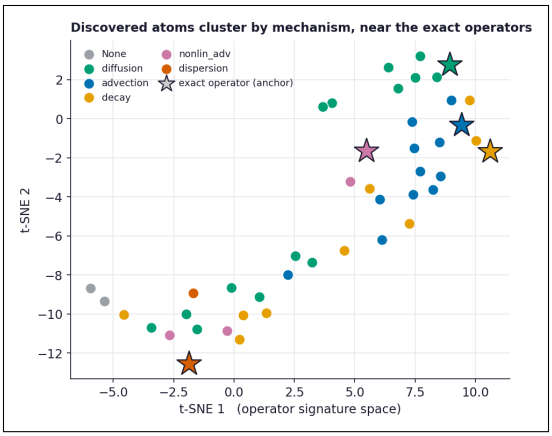

Figure 6: Discovered operators in operator-signature space (t-SNE) for the five-mechanism MIX-1D library, colored by graded mechanism; stars are the exact hand-coded operators. Discovered operators cluster near their mechanism's operator; all five mechanisms are recovered, with dispersion the rarest (a single discovered operator).

Cluster-count selection. Structure-based criteria for the cluster count K used to merge pooled duplicates (eigengap, silhouette; pre-registered) select K=4 and recover 2/5 under the original grading; held-out predictive utility selects K=15, inside the stable 5/5 region. Under discriminative grading the pooled library identifies 3/5 at every K, so K governs redundancy rather than coverage.

Identified is not composable. On the dynamic coupled flow of Figure 3, the composed library scores 0.243 (rollout mean, 60 steps), and dropping diffusion from the exact solver changes the rollout by 0.098, so diffusion matters. Yet forcing the identified diffusion operator into the composition degrades the rollout from 0.243 to 0.675, although its per-step increment behaves as diffusion (correlation +0.955 with $\nabla^2\omega$, 92% of its energy below wavenumber 8). The break depends on how the operator enters the composition: substituting it for an advection operator breaks the rollout (0.2431 $\rightarrow$ 0.6750), while appending it as a third substep is harmless (0.2413). Amplitude does not explain it: amplitude-aware operators degrade identically under substitution (0.2377 $\rightarrow$ 0.6712), and the correct and forced rollouts carry the same field-amplitude envelope. Every composition lacking the two Euler operators sits near 0.68.

Scaling discovery to native 256^2. Label-free adaptation from the 256^2 library composes at 0.0079 ± 0.0003 over three seeds against the supervised 0.0031, with both mechanisms identified on every seed. The full discovery loop improves on adaptation. At the registered budget (WIN= 8, BS= 6, $H \in \{1, 2\}$, gradient-checkpointed RK4 with verified gradient equivalence), ten seeds give median composed 0.00164 at the registered-primary selected round with 10/10 identified; a secondary rollout selector reaches 0.00093 at 7/10 identified. The means overlap (0.00279 ± 0.00254 label-free against 0.00317 ± 0.00034 supervised) and a per-seed sign test at n=10 (7/10) is not significant, so we carry the claim at the median with the spread stated. The two seeds that lose Euler

at the final round share one collapse mode: the Euler operators drift onto the diffusion action. Run on coupled flows at native 256^2, where no labeled data exists at any resolution, the loop identifies both mechanisms on ten of ten seeds and composes at 0.0116 ± 0.0014 (median 0.0118).

Selecting the adaptation length without labels. Across rounds $\{1, 2, 4, 8\}$, the held-out operator-fit residual on a disjoint unlabeled shard separates the identifying rounds (about 3.10) from those where over-adaptation collapses the Euler operator into diffusion (about 4.76), and selects round 1 without seeing identification. The collapsed round 4 composes best of all (0.0038) with only one identified mechanism, yet the rule still prefers the fully identified library.

C.4 Robustness and scope

Noise. Under additive observation noise, PDE-FIND's utility degrades by a factor of 2.4 by 10% noise while the learned library degrades by 1.2, so the gap narrows; PDE-FIND still leads on utility at every level (composed NRMSE $0.056 \to 0.134$ against $0.124 \to 0.150$ from no noise to 10%), and its selection of active terms stays more accurate (0.91 to 0.98 against 0.41 to 0.49).

A shared encoder across dimensions. The main-text libraries are per-dimension (§4). We also trained a single shared-trunk encoder on five 1D mechanisms and three 2D mechanisms jointly with every criterion pre-registered. On all three seeds the shared encoder identifies all eight mechanisms, and rollout utility stays within the pre-registered $2\times$ of per-dimension baselines (1D $1.28/1.25/1.08\times$; 2D $0.98/1.99/1.01\times$, the 1.99 marginal). Cross-dimension alignment fails, as pre-registered at 30%: codes of the same mechanism are $0/2$ mutual nearest neighbors across dimensions (permutation $p = 1.0$) on all three seeds.

C.5 External transfer to The Well

Zero-shot transfer. We apply the synthetic 2D library, without fine-tuning, to the Well's `shear_flow` vorticity channel (Reynolds 10^4, Schmidt 1) and `rayleigh_benard` buoyancy channel (Rayleigh 10^6, Prandtl 1), at native resolution, with the same context length, beam harness, and search budget as Table 12, scored by the Well's VRMSE (Table 11). The published baselines are the Well's own; its exact rollout length is not pinned in the released configuration, so we report time-averaged VRMSE at 10 and 50 steps. The diffusion-selected trajectories average 0.52 / 0.77 (10 / 50 steps) and smooth toward the mean, keeping far more high-wavenumber energy than the true flow ($53\times$ at mid-horizon); the all-trajectory 50-step mean (2.46) is dominated by the single Euler-selected divergence.

Table 11: Transfer to The Well with no fine-tuning. VRMSE (predicting the mean scores 1.0), time-averaged, 50-step horizon unless noted; baselines are trained on the dataset, our library never sees it.

Dataset	Beam selection	Ours (zero-shot)	Trained baselines
`shear_flow` (3/4 traj.)	diffusion	0.77	CNextU-Net 0.81; FNO 1.19; TFNO 1.47; U-Net 3.45
`shear_flow` (1/4 traj.)	Euler	7.5 (1.39 at 10 steps)	
`rayleigh_benard`	none engaged	declines to compose	

Zero-shot discovery on `shear_flow` identifies diffusion (0.0009 to 0.0015 against a null of 0.014) but not Euler.

Label-free adaptation on shear flow (pre-registered). Adaptation on 32 unlabeled shear-flow realizations disjoint from the graded trajectories recovers the advective mechanism: adapted discovered operators identify as Euler on three of three seeds, where the zero-shot discovered operators had missed the threshold by margins as small as 0.0005. Composed transfer improves on a matched base (0.868 to 0.818, more than twice the seed spread) but misses the pre-registered 0.72 bar, and the Euler-selected trajectory still diverges: adaptation identifies the mechanism but does not improve composed transfer to the registered level. The bar was set against a different base model than the adaptation harness used, so only the same-base change is quoted.

Developed turbulence. On developed turbulence from The Well (Ohana et al., 2024), the beam selects single-operator rollouts and never mixes operators within a step, so composition provides no

benefit over a monolithic rollout. We read this as a boundary of the splitting paradigm rather than of discovery: at developed-turbulence scales the mechanisms are not weakly coupled, and the local $O(\Delta t^3)$ splitting error of §3.1 ceases to be small.

D Reproducibility and NOS Comparison

D.1 NOS reference and our reproduction

NOS reference. We evaluated the NOS authors' released checkpoint with their released evaluation code, unmodified except for a checkpoint-loading shim. On the released NS-2D test data this gives a composed zero-shot NRMSE of 0.0045 and a direct rollout of 0.021 at 128^2 (Table 12). The headline NRMSE printed in the NOS paper (0.066) is not reproduced by the released code on the released data; candidate causes are an evaluation-split difference and the evaluation resolution (256^2 there, 128^2 here). Our comparisons are made against what the released artifacts produce under one shared protocol, never against the printed number. Reproducing the NOS numbers required the `nos` repository at commit `6adbdcd`, its vendored `torchdiffeq`, and `typeguard` <3.

Our reproduction. Our supervised operator uses the NOS architecture, its RelativeL2 loss, its NS configuration, and its rollout-horizon curriculum (horizon ramped from 1 to 16). It omits the VQ-codebook loss and in-context learning, and fixes the integration substeps at 4. To check that these omissions do not matter, we also retrained the full method (codebook, in-context, substep ramp; 23M parameters) on the full corpus of $14{,}746$ mechanism trajectories used for the released checkpoint. It composes NS at 0.0053, in the range of the released checkpoint (0.0045) and our NOS operator (0.0036 ± 0.0006). Within one codebase, corpus, and schedule ($20{,}000$ steps), switching the codebook and in-context learning off moves the composed NRMSE from 0.0053 to 0.0060. At native 256^2 (n=32, shared harness), the released checkpoint composes at 0.0056 and the retrained full method at 0.0043; NOS's published 256^2 direct rollout (0.0016) is a different evaluation mode. Training the 2D operator (23.61M parameters) takes about two GPU-hours on one consumer GPU (estimated).

Re-measured 1D battery. A September re-run of the 1D battery did not reproduce two earlier single-run NOS-recipe values (0.036 NOS operator, 0.060 full NOS method), which re-measure at 0.096 ± 0.005 and 0.107 ± 0.009 over three seeds, while the control row (0.096 ± 0.001) and the monolithic baseline (0.117 ± 0.001) reproduced exactly. We could not identify the cause. The printed tables use only the re-measured values, and every result file carries a provenance record (checkpoint path and modification time, dictionary, code date, library versions).

D.2 Resolution transfer without retraining

Every 128^2-trained library also runs at 256^2 with no retraining; the cost is roughly uniform across constructions: the supervised NOS operator moves from 0.0036 to 0.0090, the discovered library from pure-trajectory mixtures from 0.0038 to 0.013, from 0.0038 to 0.013 (released 10-step protocol, same harness as Table 12). Training or adapting at native resolution removes this cost (the 256^2 rows of Table 12). Direct, composition-free rollouts at 128^2 on the same protocol: released NOS checkpoint 0.021 (0.0016 at native 256^2, its published mode), our NOS operator 0.0038 ± 0.0006, the retrained full NOS method 0.052, the only entry above the persistence floor.

Table 12: Zero-shot Navier–Stokes on the released NS-2D test trajectories (n=32; per-step-mean NRMSE over the released 10-step horizon; one protocol for all models). The slice evolves slowly, so every row is read against the persistence floor. "Composed": operators trained on pure mechanisms only, combined by beam-searched Strang splitting. The 256^2 column holds models trained or adapted at native resolution. Rows are mean±sd where multi-seed; NOS is its single released checkpoint, the retrained full method a single run; the claim is parity. The 256^2-trained NOS-operator row (3 seeds) lands below both references; with single-run references the claim is parity, not superiority. [†]the full label-free loop of §3.3 at native 256^2, run at its registered budget: median composed over 10 seeds at the pre-registered label-free-selected round, both mechanisms identified on 10 of 10 seeds at that round; 7 of 10 seeds individually beat their supervised warm start. A secondary rollout-based selector reaches median 0.00093 but identifies both mechanisms on only 7 of 10. A lighter adaptation-only variant reaches 0.0079 ± 0.0003 (Appendix C.3).

Model	Composed, 128^2	Composed, 256^2
persistence (repeat last frame)	0.012	0.017
supervised (labeled pure-mechanism trajectories)		
released NOS checkpoint (Serrano et al., 2026)	0.0045	0.0056
NOS operator, this work (codebook, in-context omitted)	0.0036 ± 0.0006	—
NOS operator, trained at 256^2 (this work)	—	0.0031 ± 0.0004
full NOS method, retrained (this work)	0.0053	0.0043
discovered, label-free (ours: the method of §3.3, from unlabeled mixtures)		
$\mathcal{L}^{\mathrm{SSL}}$ from pure-trajectory mixtures (§5.4)	0.0038 ± 0.0001	—
$\mathcal{L}^{\mathrm{SSL}}$ from coupled NS (§5.4)	0.056	0.0116 ± 0.0014 (10 seeds)
$\mathcal{L}^{\mathrm{SSL}}$ full discovery at 256^2, registered budget, 10 seeds	—	0.0028 ± 0.0025, median 0.0016 (10/10 identified)[†]