LOCO: Local Orbit Consistency for Symmetry-Robust Neural Operators

Anonymous authors
Paper under double-blind review

Abstract

Known Lie symmetries specify how a partial differential equation (PDE) solution operator should transform under group actions, but finite supervised training data constrain this relationship only at observed input–output pairs. To enforce these symmetries without architectural rigidity, we introduce **LOCO** (**L**ocal **O**rbit **Co**nsistency), a lightweight and flexible training objective for neural operators. LOCO samples a small Lie-group transformation during training and explicitly penalizes the mismatch between the operator's prediction on the transformed input and the transformed prediction of the original input. By leveraging known input and output actions, LOCO leaves the underlying neural-operator architecture entirely unchanged and introduces zero computational overhead at inference time. We evaluate our method on a challenging periodic 64×64 two-dimensional Navier–Stokes Galilean benchmark. At a highly constrained data regime of 2% labels, combining data augmentation with local orbit consistency ($\lambda_{\text{orb}} = 0.30$) reduces the orbit out-of-distribution (OOD) relative L_2 error by 35.3% and the equivariance defect by 46.4% compared to a compute- and time-matched supervised augmentation baseline. Furthermore, a fixed lower-weight setting, $\lambda_{\text{orb}} = 0.10$, yields consistent performance gains across varying label fractions and OOD boost radii. Finally, solver-level closure residuals remain negligible, and action-mismatch ablations confirm that these performance gains depend strictly on the alignment with the intended physical symmetry constraint.

1 Introduction

Many physical learning problems approximate physical laws that are invariant to certain known transformations. A fluid equation posed on a periodic domain, for example, may transform predictably under translations or Galilean boosts; a force field should rotate its vector forces when the coordinates are rotated. Such structure is useful because it restricts the class of functions a model should learn. In machine learning this idea is usually described as an *inductive bias*: instead of learning the same rule independently at every transformed version of an input, the model or training objective can encode how predictions should transform along the corresponding symmetry orbit (Bronstein et al., 2021; Cohen & Welling, 2016). The relevant property is *equivariance*: if an input is transformed by a group element, the output should transform by the corresponding output action. A learned model that fits the training distribution can still violate this relation, and the violation becomes visible when test inputs differ from training inputs mainly by a known symmetry transformation.

For differential equations, many useful symmetries form Lie groups: continuous transformation families such as translations, rotations, scalings, and Galilean boosts (i.e., changing velocity field by a constant). Lie symmetry analysis provides a classical way to characterize transformations that map solutions of a differential equation to other valid solutions (Olver, 1993). Recent machine learning work has used such symmetries to augment PDE data, impose architectural equivariance, or canonicalize inputs before prediction (Brandstetter et al., 2022; Helwig et al., 2023; Shumaylov et al., 2025; Xu et al., 2026). These methods show that symmetry can improve data efficiency and robustness, but they typically require either transformed labels, specialized architectures, or an altered inference pipeline.

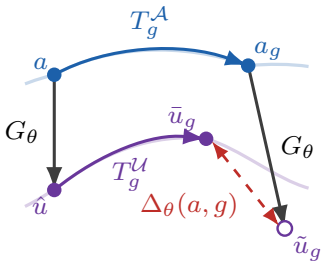

Figure 1: LOCO intuition: a small Lie-group step moves an input field from a to $T_g^{\mathcal{A}} a$ along its input orbit. A symmetry-consistent neural operator should map this local input displacement to the corresponding output-orbit displacement, so that $G_\theta(T_g^{\mathcal{A}} a) \approx T_g^{\mathcal{U}} G_\theta(a)$. LOCO penalizes the equivariance defect $\Delta_\theta(a, g)$ (equation 2), thus mitigating larger symmetry-induced shifts through the accumulated orbit-defect decomposition.

Neural operators learn maps between function spaces and are widely used as surrogate solution operators for families of partial differential equations (PDEs) (Kovachki et al., 2023; Li et al., 2021; Lu et al., 2021). In this setting, labeled examples are often generated by numerical solvers, so reducing the number of required labeled trajectories is practically important. Suppose a PDE family induces a solution operator $S : \mathcal{A} \to \mathcal{U}$, and suppose the full problem family is closed under a known group action, i.e., $S(T_g^{\mathcal{A}} a) = T_g^{\mathcal{U}} S(a)$. Then a neural operator G_θ should satisfy the same relation. Standard supervised training does not enforce this relation away from the sampled labels, and supervised data augmentation enforces it only through additional transformed labeled pairs.

We introduce **LOCO** (**Lo**cal **O**rbit **Co**nsistency), a training-time regularizer for neural operators under known Lie-group actions. LOCO samples a small transform g and penalizes the finite-orbit defect $\Delta_\theta(a, g) = G_\theta(T_g^{\mathcal{A}} a) - T_g^{\mathcal{U}} G_\theta(a)$ (see Figure 1). The loss compares two model predictions, so it does not require an additional transformed label. It can be added to an existing neural operator backbone, combined with supervised Lie augmentation, and evaluated on unlabeled inputs. It also leaves inference unchanged: after training, prediction is still the single base call $a \mapsto G_\theta(a)$, with no learned frame estimator, canonicalizer, decoder, or test-time refinement. Contributions:

- We introduce LOCO as a training-only regularizer for neural operators under known input and output Lie-group actions. The objective compares the prediction on a transformed input with the transformed prediction on the original input, requires no transformed label for the orbit term, and leaves the inference map unchanged. We then analyze LOCO through an infinitesimal tangent-defect limit and a local-to-global orbit-defect decomposition. These results connect the sampled local loss in equation 3 to larger symmetry-induced shifts and motivate the equivariance defect as an orbit-OOD diagnostic.
- We evaluate LOCO on a one-dimensional Burgers equation setting with exact symmetry transformations and on a periodic 64×64 two-dimensional Navier–Stokes setting with Galilean symmetry, both in low-label regimes. In the latter, augmentation plus orbit consistency improves orbit-OOD error and equivariance defect over compute-matched supervised augmentation with the same FNO2d inference architecture, with gains from 1% through 10% labels. Action-mismatch ablations, OOD-severity sweeps, canonicalization comparisons, and defect/OOD correlation analyses support the conclusion that the improvements come from the intended Galilean input–output action.

2 Related Work

Symmetry priors. Symmetry can be imposed through architectures, data transformations, or losses. Group-equivariant convolutional networks share weights across symmetry orbits (Cohen & Welling, 2016); steerable CNNs impose representation constraints on convolution kernels (Weiler & Cesa, 2019); and gauge-equivariant CNNs extend equivariance to local frame changes on manifolds (Cohen et al., 2019). For operator

learning, Group-Equivariant FNOs encode translation, rotation, and reflection symmetries in Fourier layers (Helwig et al., 2023). A complementary line derives transformations from the PDE itself. Lie Point Symmetry Data Augmentation constructs solution-preserving transformations from a PDE's Lie point symmetries (Brandstetter et al., 2022). Lie-symmetry losses for PINNs enforce preservation of solution families under infinitesimal generators (Akhound-Sadegh et al., 2023). Generalized Lie symmetries for PINOs use evolutionary representatives to obtain training signals that may be absent for point symmetries alone (Wang et al., 2025). The closest model-agnostic precedents are tangent propagation and recent canonicalization methods. Tangent propagation penalizes directional derivatives of a network along known transformation directions (Simard et al., 1991). Lie Algebra Canonicalization (LieLAC) uses infinitesimal generators to canonicalize inputs for pretrained models and neural operators under arbitrary Lie groups, including non-compact groups common in PDEs (Shumaylov et al., 2025). Mialon et al. (2023) use Lie symmetries as augmentations for joint-embedding self-supervised representation learning on PDE trajectories. LOCO is complementary: it regularizes an operator map for equivariance in output space rather than learning an invariant latent representation. PACE-FNO estimates a Lie-algebraic input frame, predicts in a canonical frame, and restores the output frame for symmetry-induced shifts (Xu et al., 2026). With our method, we instead keep the inference map identical to the base neural operator and apply a finite-orbit consistency penalty during training.

Neural operators. Neural operators model solution maps rather than single finite-dimensional outputs. DeepONet introduced a branch–trunk construction for learning nonlinear operators from sampled input functions and output coordinates (Lu et al., 2021). FNOs parameterize integral kernels in Fourier space and provide an efficient backbone for PDE families on regular grids (Li et al., 2021). The general neural-operator framework formalizes these models as discretization-invariant compositions of integral operators and nonlinearities, with textbook examples including Burgers' equation, Darcy flow, and Navier–Stokes (Kovachki et al., 2023). Physics-informed neural networks (PINNs) enforce PDE residuals for solution functions (Raissi et al., 2019); Physics-Informed Neural Operators (PINOs) add residual constraints to operator learning (Li et al., 2024). Extensions to non-rectangular or geometric domains include Geo-FNO, GINO, and NORM (Chen et al., 2024; Li et al., 2023a;b). PDEBench and The Well provide reusable scientific machine-learning benchmarks, while discontinuous or advection-dominated regimes expose approximation and reconstruction issues for operator learners (Lanthaler et al., 2023; Ohana et al., 2024; Takamoto et al., 2022).

3 Method

We first formalize the problem setting and LOCO objective, then state the theoretical properties used to interpret the experiments. We also specify the governing equations and symmetry actions for the PDE benchmarks.

3.1 Preliminaries

Lie groups, Lie algebras, and actions. We briefly recall the notation for Lie groups and their actions; see, e.g., Olver (1993) for their use in differential equations. A Lie group $\mathcal{G}$ is a group that is also a smooth manifold, with smooth multiplication and inversion. Its Lie algebra $\mathfrak{g} = T_e\mathcal{G}$ is the tangent space at the identity element e; elements $X \in \mathfrak{g}$ are infinitesimal generators of one-parameter transformations. The exponential map $\exp : \mathfrak{g} \to \mathcal{G}$ maps an infinitesimal generator to a finite group element along the corresponding one-parameter flow. Thus $\exp(\epsilon X)$ denotes a local transform of size ϵ in the direction X. We use $g, h \in \mathcal{G}$ for finite transformations and $X \in \mathfrak{g}$ for infinitesimal directions. The exponential map is used only locally in our training objective; it need not parameterize the entire group globally. An action of $\mathcal{G}$ on an input field space $\mathcal{A}$ is a map $T^{\mathcal{A}} : \mathcal{G} \times \mathcal{A} \to \mathcal{A}$ such that $T^{\mathcal{A}}_e = \mathrm{Id}$ and $T^{\mathcal{A}}_{hg} = T^{\mathcal{A}}_h T^{\mathcal{A}}_g$. We use an analogous output action $T^{\mathcal{U}}$ on $\mathcal{U}$. The input orbit of $a \in \mathcal{A}$ is $\mathrm{Orb}_{\mathcal{G}}(a) = \{T^{\mathcal{A}}_g a : g \in \mathcal{G}\}$. LOCO samples a generator $X_i \in \mathfrak{g}$, a small scalar step ϵ, forms the finite transform $g = \exp(\epsilon X_i)$, and applies the known input and output actions $T^{\mathcal{A}}_g$ and $T^{\mathcal{U}}_g$.

Neural operators and solution operators. Neural operators approximate maps between function spaces rather than single finite-dimensional vectors (Kovachki et al., 2023; Li et al., 2021; Lu et al., 2021). A PDE problem family induces an exact solution operator $S : \mathcal{A} \to \mathcal{U}$, where $a \in \mathcal{A}$ denotes the problem input,

such as an initial condition, coefficient field, forcing field, boundary data, or auxiliary frame variables, and $u = S(a) \in \mathcal{U}$ denotes the corresponding solution field or observation. A neural operator is a parameterized approximation $G_\theta : \mathcal{A} \to \mathcal{U}$ trained from solver-generated pairs (a_j, u_j), with $u_j \approx S(a_j)$. In the analysis we write these maps at the continuum operator level; in the experiments, a and u are represented by finite-dimensional arrays on numerical grids. Throughout the paper, S denotes the exact solution operator, G_θ denotes the learned neural operator, and $\mathcal{G}$ denotes the Lie group.

3.2 Problem Setting

Let $\mathcal{A}$ and $\mathcal{U}$ denote Banach spaces of input and output fields. A PDE family induces a solution operator $S : \mathcal{A} \to \mathcal{U}$. Let $\mathcal{G}$ be a Lie group acting on the full problem family through transformations $T_g^{\mathcal{A}}$ and $T_g^{\mathcal{U}}$. Assume closure[1] of the full problem class under $\mathcal{G}$, i.e., assume that the equation, coefficients, forcing terms, boundary or initial conditions, domain class, and observation map transform consistently. Then, the exact solution operator is *equivariant*, i.e.:

$$S(T_g^{\mathcal{A}} a) = T_g^{\mathcal{U}} S(a). \tag{1}$$

For affine output actions, write $T_g^{\mathcal{U}} u = L_g^{\mathcal{U}} u + b_g^{\mathcal{U}}$, where $L_g^{\mathcal{U}} \in \mathcal{L}(\mathcal{U},\mathcal{U})$ is the induced action on output differences. Thus $T_g^{\mathcal{U}} u - T_g^{\mathcal{U}} v = L_g^{\mathcal{U}}(u - v)$. For a composed group element $h = g_k \cdots g_1$, $L_h^{\mathcal{U}}$ denotes the linear part of $T_h^{\mathcal{U}}$. The group-action property implies $L_{hg}^{\mathcal{U}} = L_h^{\mathcal{U}} L_g^{\mathcal{U}}$ and $b_{hg}^{\mathcal{U}} = L_h^{\mathcal{U}} b_g^{\mathcal{U}} + b_h^{\mathcal{U}}$. For linear actions, $b_g^{\mathcal{U}} = 0$ and $L_g^{\mathcal{U}} = T_g^{\mathcal{U}}$. We define the operator-level *equivariance defect* as

$$\Delta_\theta(a, g) := G_\theta(T_g^{\mathcal{A}} a) - T_g^{\mathcal{U}} G_\theta(a). \tag{2}$$

Our regularizer samples $g = \exp(\epsilon X_i)$ and penalizes the corresponding equivariance defect by means of the loss term

$$\mathcal{L}_{\text{orb}}(\theta) = \mathbb{E}_{a,i,\epsilon}\left[\frac{\|\Delta_\theta(a, \exp(\epsilon X_i))\|_{\mathcal{U}}^2}{\epsilon^2 + \eta}\right]. \tag{3}$$

The objective is label-free because both terms in equation 2 are model predictions transformed by known group actions. The next results characterize the objective.

Lemma 1 (Infinitesimal tangent defect)**.** *Fix a Lie-algebra element X. Assume that the orbit curves $\epsilon \mapsto T^{\mathcal{A}}_{\exp(\epsilon X)} a$, $\epsilon \mapsto T^{\mathcal{U}}_{\exp(\epsilon X)} u$ are differentiable at $\epsilon = 0$, and assume that $G_\theta : \mathcal{A} \to \mathcal{U}$ is Fréchet differentiable at a. Define the infinitesimal action fields*

$$X_{\mathcal{A}} a := \frac{d}{d\epsilon} T^{\mathcal{A}}_{\exp(\epsilon X)} a \bigg|_{\epsilon=0}, \qquad X_{\mathcal{U}} u := \frac{d}{d\epsilon} T^{\mathcal{U}}_{\exp(\epsilon X)} u \bigg|_{\epsilon=0}.$$

Then, for $\epsilon \approx 0$,

$$\Delta_\theta(a, \exp(\epsilon X)) = \epsilon\Big(DG_\theta(a)[X_{\mathcal{A}} a] - X_{\mathcal{U}} G_\theta(a)\Big) + o(|\epsilon|).$$

Consequently, when $\eta = 0$,

$$\frac{\|\Delta_\theta(a, \exp(\epsilon X))\|_{\mathcal{U}}^2}{\epsilon^2} \to \big\|DG_\theta(a)[X_{\mathcal{A}} a] - X_{\mathcal{U}} G_\theta(a)\big\|_{\mathcal{U}}^2.$$

Proof. Let $\gamma_{\mathcal{A}}(\epsilon) = T^{\mathcal{A}}_{\exp(\epsilon X)} a$. By differentiability of the input action curve, $\gamma_{\mathcal{A}}(\epsilon) = a + \epsilon X_{\mathcal{A}} a + o(|\epsilon|)$. Since G_θ is Fréchet differentiable at a,

$$G_\theta(\gamma_{\mathcal{A}}(\epsilon)) = G_\theta(a) + DG_\theta(a)[\gamma_{\mathcal{A}}(\epsilon) - a] + o(\|\gamma_{\mathcal{A}}(\epsilon) - a\|_{\mathcal{A}}),$$

[1] Closure is problem-dependent. Periodic domains are compatible with translations and Galilean shifts because shifted fields remain periodic. Bounded domains generally permit only transformations that preserve the domain and boundary data; for example, a square with symmetric boundary conditions may admit rotations or reflections, whereas fixed nonhomogeneous Dirichlet data usually break translations and rotations. External forces can also break symmetries: a fixed gravity vector breaks full rotational symmetry unless the gravity vector is included in the input and transformed as part of the problem. Similarly, a forcing field fixed in a laboratory frame can break Galilean invariance unless it is transformed consistently. If the input representation omits variables needed to define the transformed problem, the action is not closed on the represented dataset. This is why the Galilean Navier–Stokes benchmark below includes auxiliary ambient-velocity channels; vorticity alone does not determine the Galilean frame.

and hence

$$G_\theta(T^{\mathcal{A}}_{\exp(\epsilon X)}a) = G_\theta(a) + \epsilon DG_\theta(a)[X_{\mathcal{A}}a] + o(|\epsilon|).$$

Similarly, differentiability of the output action curve at $u = G_\theta(a)$ gives

$$T^{\mathcal{U}}_{\exp(\epsilon X)}G_\theta(a) = G_\theta(a) + \epsilon X_{\mathcal{U}}G_\theta(a) + o(|\epsilon|).$$

Subtracting the two expansions proves the first claim. The normalized squared-norm limit follows by continuity of the norm. □

Lemma 1 identifies what the finite-orbit penalty is measuring near the identity. Define the Lie-algebraic equivariance defect $\tau_\theta(a; X) := DG_\theta(a)[X_{\mathcal{A}}a] - X_{\mathcal{U}}G_\theta(a)$. This is the infinitesimal failure of G_θ to commute with the generator X: the first term differentiates the learned operator after moving the input infinitesimally, while the second term moves the output infinitesimally after prediction. For sampled steps satisfying $\epsilon^2 \gg \eta$,

$$\frac{\|\Delta_\theta(a, \exp(\epsilon X))\|^2_{\mathcal{U}}}{\epsilon^2 + \eta} \approx \|\tau_\theta(a; X)\|^2_{\mathcal{U}}.$$

Thus LOCO behaves like a stochastic tangent-propagation penalty on the squared Lie-algebraic defect.

The finite-transform form is useful because it avoids explicit Jacobian-vector products $DG_\theta(a)[X_{\mathcal{A}}a]$ and avoids differentiating through the action generators. It only requires two ordinary model evaluations and known finite transformations of the represented fields. This is also closer to the finite orbit shifts used at evaluation time. The stabilizer η prevents near-zero sampled steps from producing numerically unstable weights. With fixed $\eta > 0$, steps with $\epsilon^2 \ll \eta$ contribute approximately $\frac{\epsilon^2}{\eta}\|\tau_\theta(a; X)\|^2_{\mathcal{U}}$.

Lemma 2 (Orbit-shift error decomposition)**.** *Assume that the exact solution operator is equivariant, $S(T^{\mathcal{A}}_g a) = T^{\mathcal{U}}_g S(a)$, and that the output action is affine, $T^{\mathcal{U}}_g u = L^{\mathcal{U}}_g u + b^{\mathcal{U}}_g$, with $L^{\mathcal{U}}_g \in \mathcal{L}(\mathcal{U},\mathcal{U})$. Then, for any $a \in \mathcal{A}$ and $g \in \mathcal{G}$,*

$$G_\theta(T^{\mathcal{A}}_g a) - S(T^{\mathcal{A}}_g a) = \Delta_\theta(a, g) + L^{\mathcal{U}}_g\big(G_\theta(a) - S(a)\big).$$

Consequently,

$$\|G_\theta(T^{\mathcal{A}}_g a) - S(T^{\mathcal{A}}_g a)\|_{\mathcal{U}} \le \|\Delta_\theta(a, g)\|_{\mathcal{U}} + \|L^{\mathcal{U}}_g\|\,\|G_\theta(a) - S(a)\|_{\mathcal{U}}. \tag{4}$$

If the output action is isometric on differences, then

$$\|G_\theta(T^{\mathcal{A}}_g a) - S(T^{\mathcal{A}}_g a)\|_{\mathcal{U}} \le \|\Delta_\theta(a, g)\|_{\mathcal{U}} + \|G_\theta(a) - S(a)\|_{\mathcal{U}}.$$

Proof. Using equivariance of S, we write $G_\theta(T^{\mathcal{A}}_g a) - S(T^{\mathcal{A}}_g a) = G_\theta(T^{\mathcal{A}}_g a) - T^{\mathcal{U}}_g S(a)$. Adding and subtracting $T^{\mathcal{U}}_g G_\theta(a)$ gives $G_\theta(T^{\mathcal{A}}_g a) - T^{\mathcal{U}}_g S(a) = \big(G_\theta(T^{\mathcal{A}}_g a) - T^{\mathcal{U}}_g G_\theta(a)\big) + \big(T^{\mathcal{U}}_g G_\theta(a) - T^{\mathcal{U}}_g S(a)\big)$. The first term is $\Delta_\theta(a, g)$. Since the output action is affine, the second term equals $L^{\mathcal{U}}_g\big(G_\theta(a) - S(a)\big)$. The norm bounds follow from the triangle inequality and the operator norm bound for $L^{\mathcal{U}}_g$. □

Lemma 2 separates orbit-shift prediction error into two components: the learned equivariance defect and the base prediction error transported by the output action. This is the one-step error relation behind the use of equivariance defect as an OOD diagnostic. The next result explains how defects for larger transforms decompose into local defects along the orbit.

Local-to-global control along composed transforms. Lemma 2 is a one-step statement for a fixed group element. The objective in equation 3 samples local transformations, whereas OOD evaluation may use larger transformations. The next proposition shows how equivariance defects for a composed transform decompose into local defects along the orbit.

Proposition 1 (Telescoping orbit defect)**.** *Let $g_1, \ldots, g_K \in \mathcal{G}$, define $h_j = g_j \cdots g_1$, $h_0 = e$, and $a_j = T^{\mathcal{A}}_{h_j} a$. Then*

$$\Delta_\theta(a, h_K) = \sum_{j=1}^{K} L^{\mathcal{U}}_{g_K \cdots g_{j+1}} \Delta_\theta(a_{j-1}, g_j), \tag{5}$$

where the empty product $g_K \cdots g_{K+1}$ is interpreted as the identity. Hence

$$\|\Delta_\theta(a, h_K)\|_{\mathcal{U}} \leq \sum_{j=1}^{K} \|L^{\mathcal{U}}_{g_K \cdots g_{j+1}}\| \, \|\Delta_\theta(a_{j-1}, g_j)\|_{\mathcal{U}}. \tag{6}$$

If the output action is isometric on differences, then

$$\|\Delta_\theta(a, h_K)\|_{\mathcal{U}} \leq \sum_{j=1}^{K} \|\Delta_\theta(a_{j-1}, g_j)\|_{\mathcal{U}}.$$

Proof. Let $\hat{u}_j = G_\theta(a_j)$, where $a_j = T^{\mathcal{A}}_{h_j} a$. Since $h_K = g_K \cdots g_1$, it holds that $\Delta_\theta(a, h_K) = \hat{u}_K - T^{\mathcal{U}}_{h_K} \hat{u}_0$. Insert the intermediate transported predictions:

$$\hat{u}_K - T^{\mathcal{U}}_{h_K} \hat{u}_0 = \sum_{j=1}^{K} \left(T^{\mathcal{U}}_{g_K \cdots g_{j+1}} \hat{u}_j - T^{\mathcal{U}}_{g_K \cdots g_{j+1}} T^{\mathcal{U}}_{g_j} \hat{u}_{j-1} \right),$$

where the empty product $g_K \cdots g_{K+1}$ is the identity. Since the output action is affine, i.e., $T^{\mathcal{U}}_p u - T^{\mathcal{U}}_p v = L^{\mathcal{U}}_p (u - v)$ for every composed element p. Therefore each summand equals

$$L^{\mathcal{U}}_{g_K \cdots g_{j+1}} \left(\hat{u}_j - T^{\mathcal{U}}_{g_j} \hat{u}_{j-1} \right) = L^{\mathcal{U}}_{g_K \cdots g_{j+1}} \Delta_\theta(a_{j-1}, g_j).$$

This proves the relationship given by equation 5. The norm bounds follow from the triangle inequality and the operator norm bound for $L^{\mathcal{U}}_{g_K \cdots g_{j+1}}$. If the output action is isometric on differences, these operator norms are one. □

Proposition 1 illustrates how a local loss can affect larger orbit shifts: a large transform h_K can be written as a product of small transforms, and its equivariance defect is controlled by the accumulated local defects along the intermediate states $a_0, \ldots, a_{K-1}$. This statement holds locally; for non-compact groups, such as Galilean boosts, no uniform global guarantee follows unless the local defects remain controlled along the whole path and the output actions $L^{\mathcal{U}}_h$ remain bounded on the evaluated radius. We present the corresponding evaluations in Figure 2 and Table 6.

3.3 Governing Equations and Symmetry Actions

In this section we state the governing equations for the PDE benchmarks used in the symmetry checks. The equations are included together with the input/output actions used to construct transformed examples and orbit-consistency losses. All spatial domains are periodic tori denoted $\mathbb{T}$.

One-dimensional Burgers' equation. The one-dimensional viscous Burgers equation is

$$\begin{aligned} \partial_t u + \frac{1}{2} \partial_x (u^2) &= \nu \, \partial_{xx} u, \quad (x, t) \in \mathbb{T} \times (0, T), \\ u(x, 0) &= u_0(x), \quad x \in \mathbb{T}. \end{aligned} \tag{7}$$

Equivalently, $\partial_t u + u \, \partial_x u = \nu \partial_{xx} u$. For a constant Galilean drift $c \in \mathbb{R}$, if u solves equation 7, then

$$u_c(x, t) = u(x - ct, t) + c$$

also solves equation 7. Thus the induced action on the operator $u_0 \mapsto u_T$ is

$$T^{\mathcal{A}}_c u_0(x) = u_0(x) + c, \qquad T^{\mathcal{U}}_c u_T(x) = u_T(x - cT) + c. \tag{8}$$

Spatial translations can be composed with the following action:

$$T^{\mathcal{A}}_s u_0(x) = u_0(x - s), \qquad T^{\mathcal{U}}_s u_T(x) = u_T(x - s), \qquad s \in \mathbb{T}. \tag{9}$$

Two-dimensional Navier–Stokes Galilean benchmark. The main benchmark which we use in the experiments is a periodic two-dimensional incompressible Navier–Stokes problem in vorticity form with an explicit ambient velocity channel. Let ω be vorticity, let ψ be the stream function, and let $\bar{\mathbf{u}} \in \mathbb{R}^2$ denote the constant ambient velocity/frame variable carried by the input. The total advecting velocity is

$$\mathbf{v}(\mathbf{x}, t) = \bar{\mathbf{u}} + \nabla^{\perp}\psi(\mathbf{x}, t), \qquad \nabla^{\perp}\psi = (\partial_y \psi, -\partial_x \psi).$$

The vorticity equation is

$$\begin{aligned} \partial_t \omega + \mathbf{v} \cdot \nabla \omega &= \nu \nabla^2 \omega, \quad (\mathbf{x}, t) \in \mathbb{T}^2 \times (0, T), \\ \nabla^2 \psi &= -\omega, \qquad (\mathbf{x}, t) \in \mathbb{T}^2 \times (0, T), \\ \omega(\mathbf{x}, 0) &= \omega_0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{T}^2. \end{aligned} \tag{10}$$

The Poisson equation is understood with the usual periodic zero-mean convention for ψ. The learned operator takes $a = (\omega_0, \bar{\mathbf{u}})$ as input and predicts ω_T. For a Galilean boost $c \in \mathbb{R}^2$, the input action updates the ambient frame variable,

$$T_c^{\mathcal{A}}(\omega_0, \bar{\mathbf{u}}) = (\omega_0, \bar{\mathbf{u}} + c), \tag{11}$$

and the output action shifts the terminal vorticity field by the corresponding displacement,

$$T_c^{\mathcal{U}} \omega_T(\mathbf{x}) = \omega_T(\mathbf{x} - cT). \tag{12}$$

A spatial translation $s \in \mathbb{T}^2$ acts by $T_s^{\mathcal{A}}(\omega_0, \bar{\mathbf{u}})(\mathbf{x}) = (\omega_0(\mathbf{x} - s), \bar{\mathbf{u}})$, $T_s^{\mathcal{U}} \omega_T(\mathbf{x}) = \omega_T(\mathbf{x} - s)$. In the reported Galilean experiments, we use equations 11–12; the solver-level closure check verifies that the coded data generator and these actions agree through the largest tested boost radius (Table 1).

4 Empirical Design

We test whether local orbit consistency reduces symmetry-induced OOD error and equivariance defect without changing the neural-operator backbone. With one sampled transform per minibatch, orbit training uses one additional model evaluation and a known output transformation. Inference remains $G_\theta(a)$, so deployment latency and memory match the base operator up to ordinary measurement noise. Throughout the empirical section, we use *N64 Galilean benchmark* as shorthand for the two-dimensional Navier–Stokes Galilean benchmark represented on a periodic 64×64 spatial grid. Analogously, N128 Burgers denotes the one-dimensional Burgers benchmark represented on 128 spatial grid points.

Solver-level symmetry validation. The N64 Galilean setting uses the closed representation in equations 10–12: The input contains the initial vorticity and the auxiliary ambient-velocity channels needed to define the Galilean action. This representation is required because vorticity alone does not determine the Galilean frame. Before evaluating learned models, we validate the coded symmetry at the solver level. For cached test inputs a and exact targets $S(a)$, we solve the transformed problem $S(T_c^{\mathcal{A}} a)$ and compare it to the transformed cached target $T_c^{\mathcal{U}} S(a)$:

$$R_{\text{closure}}(c) = \frac{\|S(T_c^{\mathcal{A}} a) - T_c^{\mathcal{U}} S(a)\|_2}{\|T_c^{\mathcal{U}} S(a)\|_2 + \delta}. \tag{13}$$

Table 1 reports this residual at the training boost radius 0.25 and at the OOD radii 0.35 and 0.50 used in Figure 2 and Table 6. The mean residual remains below 5×10^{-7}, and the largest observed residual is 6.22×10^{-7}, thus we conclude that the N64 Galilean discrepancy measured below is a learned-operator defect.

Training objectives. For labeled pairs (a, u), the supervised operator-regression loss is

$$\mathcal{L}_{\text{sup}}(\theta) = \mathbb{E}_{(a,u)} \left[\frac{\|G_\theta(a) - u\|_{\mathcal{U}}^2}{\|u\|_{\mathcal{U}}^2 + \delta} \right]. \tag{14}$$

Table 1: Solver-level Galilean closure for the periodic N64 2D Navier–Stokes and N128 1D Burgers datasets. The residual is defined in equation 13. Navier–Stokes rows use the training boost radius 0.25 and OOD radii $0.35, 0.50$; Burgers rows use the training boost radius 0.35 and OOD radii $0.525, 0.70, 1.05$.

Problem	Samples	Max boost	Mean residual	P95	Max
NS N64	32	0.25	3.63×10^{-7}	4.48×10^{-7}	4.77×10^{-7}
NS N64	32	0.35	3.98×10^{-7}	5.18×10^{-7}	5.83×10^{-7}
NS N64	32	0.50	4.62×10^{-7}	5.94×10^{-7}	6.22×10^{-7}
Burgers N128	32	0.35	3.16×10^{-7}	5.64×10^{-7}	6.38×10^{-7}
Burgers N128	32	0.525	3.02×10^{-7}	5.70×10^{-7}	7.51×10^{-7}
Burgers N128	32	0.70	2.52×10^{-7}	4.14×10^{-7}	7.42×10^{-7}
Burgers N128	32	1.05	3.16×10^{-7}	6.92×10^{-7}	6.93×10^{-7}

Supervised Lie augmentation uses transformed labeled pairs,

$$\mathcal{L}_{\text{aug}}(\theta) = \mathbb{E}_{(a,u),g}\left[\frac{\|G_\theta(T_g^{\mathcal{A}}a) - T_g^{\mathcal{U}}u\|_{\mathcal{U}}^2}{\|T_g^{\mathcal{U}}u\|_{\mathcal{U}}^2 + \delta}\right]. \tag{15}$$

For partial or approximate symmetries, the output norm in $\mathcal{L}_{\text{orb}}$ can be replaced by a masked or weighted norm that excludes boundary regions, invalid transformed coefficients, or unobserved output components; we use this variant only in the boundary diagnostic in Appendix D. The four training modes used in the experiments are

$$\mathcal{L}_{\text{FNO}} = \mathcal{L}_{\text{sup}}, \tag{16}$$

$$\mathcal{L}_{\text{orbit}} = \mathcal{L}_{\text{sup}} + \lambda_{\text{orb}}\mathcal{L}_{\text{orb}}, \tag{17}$$

$$\mathcal{L}_{\text{aug}}^{\text{train}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{aug}}, \tag{18}$$

$$\mathcal{L}_{\text{aug+orbit}} = \mathcal{L}_{\text{aug}}^{\text{train}} + \lambda_{\text{orb}}\mathcal{L}_{\text{orb}}. \tag{19}$$

The augmentation term uses transformed labels; the orbit term uses transformed model predictions and can therefore also be evaluated on unlabeled inputs.

Optional unlabeled adaptation. Because $\mathcal{L}_{\text{orb}}$ is label-free once the actions are known, it can also be evaluated on target inputs. We use this only in Appendix A, where the orbit term is anchored to a pretrained model because equivariance alone does not identify the solution operator.

Evaluation metrics. Let $\mathcal{C}_r$ denote the Galilean test-boost distribution with maximum boost r. We report in-distribution relative error,

$$E_{\text{ID}}(\theta) = \mathbb{E}_{(a,u)}\left[\frac{\|G_\theta(a) - u\|_2}{\|u\|_2 + \delta}\right], \tag{20}$$

orbit OOD relative error at severity r,

$$E_{\text{OOD}}(\theta; r) = \mathbb{E}_{(a,u),\, c\sim\mathcal{C}_r}\left[\frac{\|G_\theta(T_c^{\mathcal{A}}a) - T_c^{\mathcal{U}}u\|_2}{\|T_c^{\mathcal{U}}u\|_2 + \delta}\right], \tag{21}$$

and relative equivariance defect,

$$D_{\text{eq}}(\theta; r) = \mathbb{E}_{a,\, c\sim\mathcal{C}_r}\left[\frac{\|G_\theta(T_c^{\mathcal{A}}a) - T_c^{\mathcal{U}}G_\theta(a)\|_2}{\|T_c^{\mathcal{U}}G_\theta(a)\|_2 + \delta}\right]. \tag{22}$$

5 Results

Training protocol and compute. We compare supervised FNO, supervised Lie augmentation, supervised orbit consistency, and augmentation plus orbit consistency, primarily with 2% labeled data and average

Table 2: N64 Galilean at 2% labels over five seeds. All rows use the same FNO2d architecture with 2,668,609 parameters. Lower is better for relative L^2, orbit OOD relative L^2, and equivariance defect. The $\lambda_{\text{orb}} = 0.10$ row is used for the fixed-weight analyses; $\lambda_{\text{orb}} = 0.30$ is stronger in the orbit-weight sweep. Training and inference costs are reported in Table 14.

Method	Updates/epoch	Relative L^2	Orbit OOD L^2	Equiv. defect
FNO	4	0.9287 ± 0.0228	0.9568 ± 0.0223	0.6988 ± 0.0173
FNO + orbit	4	0.4646 ± 0.0394	0.4952 ± 0.0348	0.3359 ± 0.0170
FNO + aug.	6	0.5009 ± 0.0590	0.5276 ± 0.0525	0.3816 ± 0.0322
FNO + aug. + orbit, $\lambda_{\text{orb}} = 0.10$	4	0.3676 ± 0.0360	0.3950 ± 0.0334	0.2677 ± 0.0174
FNO + aug. + orbit, $\lambda_{\text{orb}} = 0.30$	4	$\mathbf{0.3149 \pm 0.0298}$	$\mathbf{0.3416 \pm 0.0289}$	$\mathbf{0.2044 \pm 0.0160}$

the results over five random seeds. The orbit weight is $\lambda_{\text{orb}} = 0.10$ unless stated otherwise. In the primary comparison, supervised augmentation uses six update steps per epoch, while augmentation plus orbit consistency uses four update steps per epoch. Table 14 reports the resulting training budget.

Main N64 result. Table 2 reports the 2%-label result. The $\lambda_{\text{orb}} = 0.10$ setting is used in the label-efficiency, OOD-severity, action-mismatch ablations, and diagnostic analyses below. With $\lambda_{\text{orb}} = 0.30$, orbit OOD relative L^2 decreases from 0.5276 ± 0.0525 for compute-matched augmentation to 0.3416 ± 0.0289, a 35.3% reduction. Equivariance defect decreases from 0.3816 ± 0.0322 to 0.2044 ± 0.0160, a 46.4% reduction. Relative to plain supervised FNO, the reductions are 64.3% for orbit OOD error and 70.7% for equivariance defect. Since all rows use the same FNO2d architecture, the inference path and parameter count are unchanged. The corresponding seed-matched paired deltas are also separated from zero: $-0.1860\,[-0.2375, -0.1345]$ for orbit OOD error and $-0.1772\,[-0.2169, -0.1374]$ for equivariance defect; see Table 4.

Label efficiency. For each $\rho \in \{1\%, 2\%, 5\%, 10\%\}$, all objectives draw minibatch inputs from the same labeled subset $\mathcal{D}_\rho^\ell \subset \mathcal{D}_{\text{train}}$. The supervised and augmentation losses use labels; the orbit loss uses only the inputs from the same subset, so Table 3 measures low-label regularization rather than semi-supervised learning. The separate semi-supervised experiment in Appendix A is the only setting where orbit updates use the full unlabeled training split.

Table 3 reports the completed 1%, 2%, 5%, and 10% labeled-subset settings using the fixed orbit weight $\lambda_{\text{orb}} = 0.10$. At each label fraction, supervised augmentation and augmentation plus orbit consistency use the same labeled subset and the same FNO2d inference architecture; the orbit term is label-free but is not given additional unlabeled inputs in this table. Augmentation plus orbit consistency lowers orbit OOD error and equivariance defect at every label fraction. The OOD reductions over compute-matched augmentation are 13.7%, 25.1%, 22.3%, and 11.8% at 1%, 2%, 5%, and 10% labels, respectively. The corresponding defect reductions are 21.3%, 29.8%, 33.2%, and 23.5%. The largest gains occur in the low-label regime, but the 10% row shows that the effect persists after the supervised augmentation baseline becomes substantially stronger. The tuned 2% result in Table 2 is stronger than the fixed-weight 2% row shown here; we keep Table 3 at $\lambda_{\text{orb}} = 0.10$ to isolate label fraction rather than retuning λ_{orb} at each data budget.

Paired seed differences. Because seeds are matched across the central augmentation comparisons, Table 4 reports paired deltas, defined as augmentation plus orbit minus augmentation; negative values indicate improvement. The main 2%, $\lambda_{\text{orb}} = 0.30$ comparison is separated from zero on ID error, orbit OOD error, and equivariance defect. The fixed $\lambda_{\text{orb}} = 0.10$ comparison shows the same sign pattern and is used for the label-efficiency, OOD-severity, action-mismatch, and diagnostic analyses. The 10% comparison remains separated from zero despite using 3 seeds.

Action-mismatch ablations. Table 5 tests whether the improvement can be explained by generic two-view consistency. The identity-output-action ablation keeps the input transform and sample pairing but replaces the physical orbit target $T_c^{\mathcal{U}} G_\theta(a)$ by $G_\theta(a)$. The shuffled-target ablation keeps the physical output action but pairs each transformed prediction with an incorrect orbit target from the minibatch. Both

Table 3: N64 Galilean label efficiency metrics at $\lambda_{\text{orb}} = 0.10$. The 2% row uses five seeds; others three seeds.

Labels	Method	ID L^2	Orbit OOD L^2	Equiv. defect	Latency ms/sample
1%	FNO + aug.	0.7341 ± 0.0848	0.7421 ± 0.0838	0.5080 ± 0.0456	12.76 ± 0.12
1%	FNO + aug. + orbit	$\mathbf{0.6285 \pm 0.0510}$	$\mathbf{0.6406 \pm 0.0508}$	$\mathbf{0.3999 \pm 0.0197}$	16.98 ± 1.96
2%	FNO + aug.	0.5009 ± 0.0590	0.5276 ± 0.0525	0.3816 ± 0.0322	18.76 ± 7.34
2%	FNO + aug. + orbit	$\mathbf{0.3676 \pm 0.0360}$	$\mathbf{0.3950 \pm 0.0334}$	$\mathbf{0.2677 \pm 0.0174}$	16.87 ± 3.19
5%	FNO + aug.	0.2950 ± 0.0061	0.3237 ± 0.0013	0.2427 ± 0.0014	11.46 ± 0.50
5%	FNO + aug. + orbit	$\mathbf{0.2260 \pm 0.0119}$	$\mathbf{0.2515 \pm 0.0112}$	$\mathbf{0.1621 \pm 0.0078}$	12.54 ± 0.68
10%	FNO + aug.	0.1694 ± 0.0121	0.1996 ± 0.0091	0.1596 ± 0.0076	14.37 ± 3.71
10%	FNO + aug. + orbit	$\mathbf{0.1510 \pm 0.0049}$	$\mathbf{0.1760 \pm 0.0027}$	$\mathbf{0.1221 \pm 0.0004}$	14.61 ± 1.54

Table 4: Paired seed differences for central comparisons. Delta is augmentation plus orbit minus augmentation, so negative values indicate improvement. The $\lambda_{\text{orb}} = 0.30$ rows correspond to the main 2% setting; the $\lambda_{\text{orb}} = 0.10$ rows correspond to the fixed setting used in the fixed-weight analysis suite. N64 Galilean 2% is averaged over 5 seeds, others over 3 seeds.

Setting	Metric	Aug. mean	Aug. + orbit mean	Paired delta [95% CI]
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.30$	ID L^2	0.5009	0.3149	$-0.1860\,[-0.2393, -0.1327]$
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.30$	OOD L^2	0.5276	0.3416	$-0.1860\,[-0.2375, -0.1345]$
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.30$	Equiv. defect	0.3816	0.2044	$-0.1772\,[-0.2169, -0.1374]$
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.10$	ID L^2	0.5009	0.3676	$-0.1332\,[-0.1731, -0.0933]$
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.10$	OOD L^2	0.5276	0.3950	$-0.1326\,[-0.1700, -0.0953]$
N64 Galilean 2%, $\lambda_{\text{orb}} = 0.10$	Equiv. defect	0.3816	0.2677	$-0.1139\,[-0.1397, -0.0881]$
N64 Galilean 10%, $\lambda_{\text{orb}} = 0.10$	ID L^2	0.1694	0.1510	$-0.0185\,[-0.0364, -0.0005]$
N64 Galilean 10%, $\lambda_{\text{orb}} = 0.10$	OOD L^2	0.1996	0.1760	$-0.0235\,[-0.0393, -0.0078]$
N64 Galilean 10%, $\lambda_{\text{orb}} = 0.10$	Equiv. defect	0.1596	0.1221	$-0.0375\,[-0.0555, -0.0196]$
1D Burgers	ID L^2	2.51×10^{-3}	2.65×10^{-3}	$1.43 \times 10^{-4}\,[-2.13 \times 10^{-4}, 4.99 \times 10^{-4}]$
1D Burgers	OOD L^2	4.05×10^{-3}	3.81×10^{-3}	$-2.42 \times 10^{-4}\,[-7.01 \times 10^{-4}, 2.16 \times 10^{-4}]$
1D Burgers	Equiv. defect	2.65×10^{-3}	2.28×10^{-3}	$-3.70 \times 10^{-4}\,[-7.02 \times 10^{-4}, -3.84 \times 10^{-5}]$

ablations use the same supervised augmentation setting, optimizer budget, FNO2d backbone, and $\lambda_{\text{orb}} = 0.10$ as the physical Aug+Orbit row. The physical Aug+Orbit loss improves all metrics, while both action-mismatch ablations are worse than supervised augmentation, supporting the conclusion that the gain depends on the intended Galilean input–output action rather than generic two-view consistency.

OOD severity. Figure 2 and Table 6 evaluate boosts from below the training radius to twice the training radius. The solver-level closure check in Table 1 validates the same coded Galilean transform at the training radius 0.25 and at the OOD radii 0.35 and 0.50. The severity curve is therefore testing learned robustness along a numerically closed symmetry orbit. This experiment is the empirical counterpart of Proposition 1: a larger boost can be decomposed into smaller boosts, and equation 6 controls the large-radius equivariance defect by accumulated local defects along the orbit. Augmentation plus orbit consistency remains better than compute-matched augmentation at every tested severity. The relative OOD reductions are 26.4%, 25.8%, 25.3%, 23.6%, and 20.3% at maximum boosts 0.10, 0.20, 0.25, 0.35, and 0.50, respectively. Defect reductions follow the same trend. The reduction decreases at larger boosts, as expected from the accumulation in equation 6, but it does not reverse.

Equivariance defect as an OOD diagnostic. The orbit-error decomposition in equation 4 motivates equivariance defect as a diagnostic for orbit OOD error. We test this across the performed N64 Galilean runs on the total of 90 runs spanning seeds, label fractions, orbit weights, and boost severities. We report Pearson correlation equal 0.851, and Spearman equal 0.838, between equivariance defect and orbit OOD relative L^2. The residualized correlation remains high at 0.779 after controlling for source, method, label fraction, maximum boost, and λ_{orb}.

Table 5: Action-mismatch ablations at 2% labels over matched seeds. The identity-output-action row trains $G_\theta(T_c^{\mathcal{A}} a) \approx G_\theta(a)$. The shuffled-target row pairs each transformed prediction with an incorrect orbit target from the minibatch. The results are averaged over three seeds. Lower is better.

Method	ID L^2	OOD L^2	Equiv. defect
FNO + aug.	0.5181 ± 0.0377	0.5415 ± 0.0370	0.3867 ± 0.0237
FNO + aug. + orbit	$\mathbf{0.3674 \pm 0.0136}$	$\mathbf{0.3915 \pm 0.0151}$	$\mathbf{0.2630 \pm 0.0087}$
FNO + aug. + identity output action	0.8827 ± 0.0194	0.8994 ± 0.0191	0.6236 ± 0.0238
FNO + aug. + shuffled target	0.9654 ± 0.0034	0.9677 ± 0.0036	0.5631 ± 0.0318

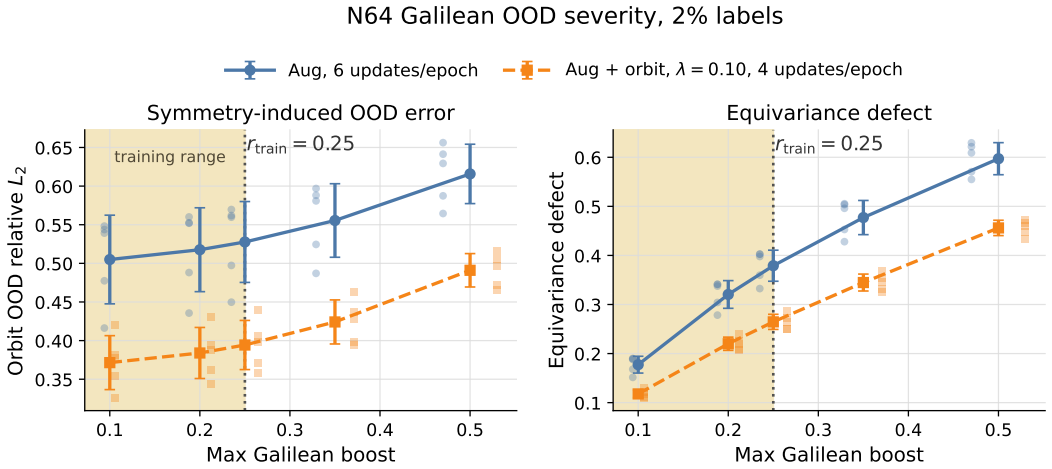

Figure 2: OOD severity at 2% labels with $\lambda_{\text{orb}} = 0.10$. The dashed line marks the training boost radius. Augmentation plus orbit consistency remains below compute-matched augmentation from below the training radius through twice the training radius. The error bars are standard deviations across seeds. Augmentation is compute-matched using 6 optimizer updates per epoch versus 4 for orbit models. Lower is better.

Canonicalization baselines. The N64 Galilean input contains observable frame channels $c = (c_x, c_y)$, so analytic canonicalization is available as a test-time baseline. We therefore separate two quantities: paired-source relative-boost calibration uses the sampled evaluation boost and the paired source input. The observable analytic canonicalization baseline reads the frame coordinate from the input, maps the input to the zero-boost frame, evaluates the same FNO once, and maps the prediction back. This baseline is deployable when the frame channels are reliable, but it changes the inference map by adding explicit pre- and postprocessing. LOCO instead leaves the deployed model equal to the base neural-operator call. Table 7 reports the direct, paired-source, and observable-canonical errors.

Orbit-weight ablation and additional evidence. Table 8 reports the orbit-weight sweep at 2% labels. We use $\lambda_{\text{orb}} = 0.10$ for the fixed-weight analyses: label efficiency, OOD severity, action-mismatch ablations, and defect/OOD diagnostics. Increasing the weight to $\lambda_{\text{orb}} = 0.30$ gives the strongest 2%-label result in the sweep, reducing ID error, orbit-OOD error, and equivariance defect by 14.3%, 13.5%, and 23.6%, respectively, relative to $\lambda_{\text{orb}} = 0.10$.

Additional results. Appendix A reports auxiliary PDE experiments. Post-hoc unlabeled target adaptation gives small stable OOD reductions with negligible ID drift; semi-supervised orbit training strongly reduces equivariance defect but is weaker than supervised augmentation plus LOCO in predictive error; and 1D Burgers provides an exact-symmetry check where augmentation is near-saturated, leaving small OOD gains but consistent defect reductions.

Table 6: Relative reductions of augmentation plus orbit consistency over compute-matched augmentation across Galilean OOD severities at 2% labels using the fixed $\lambda_{\text{orb}} = 0.10$ LOCO setting.

Max boost	OOD reduction	Defect reduction
0.10	26.4%	33.6%
0.20	25.8%	31.3%
0.25	25.3%	30.2%
0.35	23.6%	27.8%
0.50	20.3%	23.6%

Table 7: Canonicalization comparisons at 2% labels and maximum boost 0.50. Paired-source relative-boost calibration uses the paired source input and sampled relative boost. Observable analytic canonicalization reads the frame coordinate from the input and changes inference by pre/postprocessing.

Method	Direct OOD L^2	Paired-source relative-boost L^2	Observable canonical L^2
FNO + aug.	0.6158 ± 0.0384	0.5009 ± 0.0590	0.4658 ± 0.0784
FNO + aug. + orbit, $\lambda_{\text{orb}} = 0.10$	$\mathbf{0.4910 \pm 0.0216}$	$\mathbf{0.3676 \pm 0.0360}$	$\mathbf{0.3253 \pm 0.0564}$

Appendix A.2 reports an out-of-domain rMD17 atomistic force check with the $E(3)$-equivariant force action $F \mapsto FQ^\top$. Augmentation plus LOCO improves force MAE and substantially reduces force equivariance defect, whereas action-mismatch ablations do not reproduce the reduction.

Appendix D reports controls for tangent propagation, η/ϵ sensitivity, discrete D_4 symmetries, and non-periodic Dirichlet boundaries. These tests show that finite-difference orbit consistency is competitive with tangent propagation, robust across the tested step/stabilizer range, transferable beyond Galilean symmetry, and localizable when boundaries break global symmetry.

6 Final remarks

Summary. We proposed LOCO, a training-time regularizer that penalizes the equivariance defect of a neural operator along known local group orbits. For a sampled transform g, LOCO compares the two paths $G_\theta(T_g^{\mathcal{A}} a)$ and $T_g^{\mathcal{U}} G_\theta(a)$, thereby encouraging consistency between the learned operator and the prescribed input–output symmetry action. The method is architecture-agnostic, uses deterministic group actions on the represented fields, and adds no inference-time module: after training, deployment remains the single map $a \mapsto G_\theta(a)$. In the grid-based settings studied here, the required actions are implemented directly on the discretized fields, e.g., by circular shifts for integer periodic translations, Fourier phase shifts for non-integer periodic translations, and array rotations or reflections for finite square-grid symmetries.

Limitations. LOCO assumes that the relevant input and output group actions are known, numerically reliable, and compatible with the data-generating problem family. In the N64 Galilean benchmark, we validate solver-level closure at the training and OOD boost radii, but this check is benchmark-specific; each new PDE, boundary condition, forcing convention, coefficient field, and observation map requires its own validation. If the dataset is not closed under the proposed action, if forcing or boundary conditions remain fixed in a frame that the action changes, or if the input representation omits variables needed to define the transformed problem, LOCO can enforce an incorrect constraint.

The orbit loss constrains equivariance, and hence it must be paired with supervised labels, a pretrained-model anchor, or another accuracy-preserving objective when used for adaptation. Its finite-difference interpretation depends on the step distribution and stabilizer η: with fixed η, very small sampled steps are damped rather than treated as exact tangent estimates. Finally, our main experiments use an FNO backbone and periodic grids. Additional evidence is needed for non-periodic boundaries, irregular meshes, strongly forced systems, complex observation maps, and three-dimensional flows. In the N64 Galilean dataset the frame coordinate is observable, so analytic canonicalization is available as a separate test-time procedure. LOCO does not

Table 8: Orbit-weight ablation at 2% labels. $\lambda_{\text{orb}} = 0.10$ is the fixed setting used for label-efficiency, OOD-severity, action-mismatch, and diagnostic analyses; $\lambda_{\text{orb}} = 0.30$ is strongest in this sweep. Results are averaged over five seeds.

λ_{orb}	ID L^2	Orbit OOD L^2	Eq. defect
0.10	0.3676 ± 0.0360	0.3950 ± 0.0334	0.2677 ± 0.0174
0.20	0.3291 ± 0.0318	0.3564 ± 0.0304	0.2268 ± 0.0165
0.30	$\mathbf{0.3149 \pm 0.0298}$	$\mathbf{0.3416 \pm 0.0289}$	$\mathbf{0.2044 \pm 0.0160}$

replace such canonicalization; it improves the unchanged neural-operator map and can be composed with canonicalization when reliable frame variables are available.

Future work. A natural next direction is to extend LOCO beyond exactly specified global symmetries. This includes learning approximate action maps or validity masks from data, adapting the loss to local or partial symmetries, and testing the method on non-periodic domains and irregular geometries where group actions are only approximate at the discretized level. Another direction is to combine LOCO with canonicalization or equivariant architectures, using orbit consistency as a training signal for parts of the problem where exact architectural equivariance is inconvenient or unavailable. Appendix A.2 reports a preliminary rMD17 ethanol force-prediction mechanism check in this direction; broader atomistic benchmarks and comparisons with strong $E(3)$-equivariant baselines are needed.

References

Tara Akhound-Sadegh, Laurence Perreault-Levasseur, Johannes Brandstetter, Max Welling, and Siamak Ravanbakhsh. Lie point symmetry and physics-informed networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8493c860bec41705f7743d5764301b94-Abstract-Conference.html.

Johannes Brandstetter, Max Welling, and Daniel E. Worrall. Lie point symmetry data augmentation for neural PDE solvers. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2241–2256. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/brandstetter22a.html.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Gengxiang Chen, Xu Liu, Qinglu Meng, Lu Chen, Changqing Liu, and Yingguang Li. Learning neural operators on riemannian manifolds. *National Science Open*, 3(6), 2024. doi: 10.1360/nso/20240001. URL https://doi.org/10.1360/nso/20240001.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999. PMLR, 20–22 Jun 2016. URL https://proceedings.mlr.press/v48/cohenc16.html.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2019. URL https://proceedings.mlr.press/v97/cohen19d.html.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant Fourier neural operators for partial differential equations. In *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. URL https://proceedings.mlr.press/v202/helwig23a.html.

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 https://www.jmlr.org/papers/v24/21-1524.html.

Samuel Lanthaler, Roberto Molinaro, Patrik Hadorn, and Siddhartha Mishra. Nonlinear reconstruction for operator learning of PDEs with discontinuities. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CrfhZAsJDsZ.

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. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388): 1–26, 2023a. URL https://www.jmlr.org/papers/v24/23-0064.html.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/70518ea42831f02afc3a2828993935ad-Abstract-Conference.html.

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. URL https://doi.org/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. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL https://doi.org/10.1038/s42256-021-00302-5.

Grégoire Mialon, Quentin Garrido, Hannah Lawrence, Danyal Rehman, Yann LeCun, and Bobak Kiani. Self-supervised learning with lie symmetries for partial differential equations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 28973–29004. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5c46ae130105fa012da0446126c01d1d-Paper-Conference.pdf.

Ruben Ohana, Michael McCabe, Lucas Thibaut Meyer, Rudy Morel, Fruzsina Julia Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond Buschman Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich Kerswell, Suryanarayana Maddu, Jonah M. Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, et al. The well: A large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=00Sx577BT3.

Peter J. Olver. *Applications of Lie Groups to Differential Equations*, volume 107 of *Graduate Texts in Mathematics*. Springer, New York, 2 edition, 1993. doi: 10.1007/978-1-4612-4350-2. URL https://doi.org/10.1007/978-1-4612-4350-2.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045. URL https://doi.org/10.1016/j.jcp.2018.10.045.

Zakhar Shumaylov, Peter Zaika, James Rowbottom, Ferdia Sherry, Melanie Weber, and Carola-Bibiane Schönlieb. Lie algebra canonicalization: Equivariant neural operators under arbitrary lie groups. In

International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=7PLpiVdnUC.

Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. Tangent prop - a formalism for specifying selected invariances in an adaptive network. In J. Moody, S. Hanson, and R.P. Lippmann (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991. URL https://proceedings.neurips.cc/paper_files/paper/1991/file/65658fde58ab3c2b6e5132a39fae7cb9-Paper.pdf.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0a9747136d411fb83f0cf81820d44afb-Abstract-Datasets_and_Benchmarks.html.

Amy Xiang Wang, Zakhar Shumaylov, Peter Zaika, Ferdia Sherry, and Carola-Bibiane Schönlieb. Generalized Lie symmetries in Physics-Informed neural operators. *Journal of Scientific Computing and Machine Learning*, 1:43–51, 2025. URL https://scml.jp/2025/paper/7.pdf.

Maurice Weiler and Gabriele Cesa. General $E(2)$-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/9580-general-e2-equivariant-steerable-cnns.

Jiaxiao Xu, Changhong Mou, Yeyu Zhang, and Fengxiang He. Physics-aligned canonical equivariant fourier neural operator under symmetry-induced shifts. *arXiv*, 2026. URL https://arxiv.org/abs/2605.18606.

Appendix Contents

A Additional Experiments **15**

A.1 Adaptation on unlabeled targets 15

A.2 Atomistic Vector-output Mechanism Study 16

B Algorithm and Complexity **17**

C PACE-style Observable-frame Canonicalization **18**

D Additional Controls and Robustness Experiments **19**

E Training Budget **20**

Appendix structure. Appendix A reports additional N64, Burgers, and target-adaptation experiments. Appendix B gives the training algorithm and complexity accounting. Appendix C gives the observable-frame canonicalization comparison. Appendix D reports auxiliary controls and robustness checks. Appendix E reports training-budget details.

A Additional Experiments

A.1 Adaptation on unlabeled targets

Unlabeled target-input adaptation. We evaluate post-hoc adaptation on unlabeled target inputs at maximum boosts 0.35 and 0.50. Starting from trained 2%-label checkpoints, we update only the projection head for five epochs using the target orbit loss with prediction preservation. A random-orbit-preservation

control uses the same number of optimizer steps. The effect is modest: OOD error decreases by 0.2% to 0.6%, with negligible in-distribution drift. Table 9 reports the post-adaptation deltas.

Table 9: Unlabeled target-input adaptation at 2% labels. Negative OOD delta means lower post-adaptation OOD error; ID drift is the signed change in in-distribution relative L^2.

Source checkpoint	Target boost	Pre OOD L^2	Post OOD L^2	OOD delta	ID drift
FNO + aug.	0.35	0.5688	0.5664	−0.43%	−0.0013
FNO + aug.	0.50	0.6281	0.6246	−0.56%	−0.0013
FNO + aug. + orbit	0.35	0.4200	0.4191	−0.21%	0.0004
FNO + aug. + orbit	0.50	0.4870	0.4847	−0.47%	0.0004

Semi-supervised joint orbit training. We also test a semi-supervised joint objective in which the supervised 2% labeled split is augmented with orbit-consistency updates on the full unlabeled training split. This variant reduces equivariance defect from 0.3816 ± 0.0322 for augmentation to 0.2463 ± 0.0217, a 35.4% reduction, and slightly improves orbit OOD error from 0.5276 ± 0.0525 to 0.5086 ± 0.0490. It is weaker in predictive error than supervised augmentation plus orbit consistency, so we treat it as secondary evidence: unlabeled orbit data regularizes symmetry, but it is not the strongest training mode in this benchmark.

1D Burgers exact-symmetry sanity check. The 1D Burgers experiments provide a compact second-setting check. Supervised augmentation nearly solves the Galilean shift in this setting, so the incremental OOD gain from orbit consistency is small. Augmentation plus orbit consistency lowers orbit OOD relative L^2 from 0.00405 to 0.00381 and equivariance defect from 0.00265 to 0.00228. The paired OOD interval crosses zero, while the paired defect interval remains below zero, as reported in Table 4.

A.2 Atomistic Vector-output Mechanism Study

We include a lightweight rMD17 ethanol force-prediction experiment to test whether the orbit-consistency mechanism transfers beyond grid-based PDE outputs. With this experiment we intend to test whether a physically correct input-output group action improves force prediction and force equivariance in a non-equivariant atomistic model. We note that molecular modeling has strong exactly $E(3)$-equivariant architectures; LOCO is complementary to such models and is most relevant when exact equivariance is absent or approximate.

The molecule which we focus on is ethanol, with 9 atoms per conformation. The predictor is a fixed-molecule MLP with 57,755 parameters. It flattens per-atom inputs $(x, y, z, Z/10)$, uses hidden width 128 and depth 4, and outputs three Cartesian force components per atom. As such, the model has no built-in $E(3)$-equivariance.

Let $X \in \mathbb{R}^{N\times 3}$ denote atom coordinates and $F_\theta(X) \in \mathbb{R}^{N\times 3}$ predicted forces. For row-vector coordinates, we sample a rigid motion $(Q, t) \in SO(3) \times \mathbb{R}^3$, where the rotation axis is sampled from a normalized three-dimensional Gaussian direction, the angle is sampled uniformly from $[-\pi/2, \pi/2]$, and each component of t is sampled uniformly from $[-1, 1]$. Before applying rigid motions, we center each conformation as $\widetilde{X} = C(X) = X - \mathbf{1}\bar{x}^\top$. The rigid-motion action used in the experiment is $T^{\mathcal{A}}_{Q,t}\widetilde{X} = \widetilde{X}Q^\top + \mathbf{1}t^\top$, $T^{\mathcal{U}}_{Q,t}F = FQ^\top$. The force-equivariance defect is

$$D^F_{\mathrm{eq}}(\theta) = \mathbb{E}_{X,Q,t}\left[\frac{\|F_\theta(T^{\mathcal{A}}_{Q,t}X) - T^{\mathcal{U}}_{Q,t}F_\theta(X)\|_F}{\|T^{\mathcal{U}}_{Q,t}F_\theta(X)\|_F + \delta}\right]. \tag{23}$$

The identity-output-action ablation keeps paired transformed inputs but replaces the physical target $T^{\mathcal{U}}_{Q,t}F_\theta(X)$ by $F_\theta(X)$. The shuffled-target ablation uses the physical output action but pairs each transformed prediction with a different source example in the minibatch. Table 10 reports the 500-label rMD17 mechanism check.

Relative to supervised rigid-motion augmentation at 500 labels, MLP+aug.+LOCO reduces force MAE by 3.4%, orbit force MAE by 3.3%, ID relative L^2 by 3.6%, orbit relative L^2 by 3.5%, and force equivariance

Table 10: rMD17 ethanol force-prediction mechanism check at 500 labeled conformations. Entries are mean ± sample standard deviation over three matched seeds. Lower is better. The results are averaged over 3 seeds.

Method	Force MAE	Orbit force MAE	Eq. defect	ID L^2	Orbit L^2
MLP	20.2503 ± 0.0196	20.3762 ± 0.0138	0.7400 ± 0.0169	0.9972 ± 0.0011	1.0014 ± 0.0012
MLP + aug.	20.1483 ± 0.0071	20.2196 ± 0.0174	0.5394 ± 0.0031	0.9923 ± 0.0003	0.9934 ± 0.0004
MLP + aug. + id-output action	20.1034 ± 0.0056	20.1861 ± 0.0256	0.5877 ± 0.0090	0.9906 ± 0.0005	0.9921 ± 0.0007
MLP + aug. + shuffled target	20.1554 ± 0.0136	20.2252 ± 0.0185	0.6824 ± 0.0120	0.9938 ± 0.0008	0.9943 ± 0.0007
MLP + aug. + LOCO	$\mathbf{19.4633 \pm 0.1725}$	$\mathbf{19.5615 \pm 0.1646}$	$\mathbf{0.2357 \pm 0.0021}$	$\mathbf{0.9566 \pm 0.0089}$	$\mathbf{0.9586 \pm 0.0086}$

Algorithm 1 Stochastic local orbit consistency training

Require: Labeled data $\mathcal{D} = \{(a_j, u_j)\}$, neural operator G_θ, actions $T_g^{\mathcal{A}}, T_g^{\mathcal{U}}$, generators $\{X_i\}_{i=1}^m$, step samplers $\rho_{\text{aug}}, \rho_{\text{orb}}$, weight λ_{orb}, stabilizers δ, η, flags $s_{\text{aug}}, s_{\text{orb}}$

for each training step **do**
  Sample minibatch $\mathcal{B} = \{(a_b, u_b)\}_{b=1}^B$
  $\hat{u}_b \leftarrow G_\theta(a_b)$ for $b = 1, \ldots, B$ ▷ base prediction; reused below
  $\mathcal{L} \leftarrow \widehat{\mathcal{L}}_{\text{sup}}(\theta; \mathcal{B})$ ▷ minibatch estimate of equation 14
  if s_{aug} **then**
    Draw $g_b \sim \rho_{\text{aug}}$, form $a_b^g = T_{g_b}^{\mathcal{A}} a_b$, $u_b^g = T_{g_b}^{\mathcal{U}} u_b$
    $\hat{u}_b^g \leftarrow G_\theta(a_b^g)$ ▷ supervised transformed-label branch
    $\mathcal{L} \leftarrow \mathcal{L} + \widehat{\mathcal{L}}_{\text{aug}}(\theta; \mathcal{B}, \{g_b\})$ ▷ equation 15
  end if
  if s_{orb} **then**
    Draw local transforms $h_b = \exp(\epsilon_b X_{i_b}) \sim \rho_{\text{orb}}$ ▷ small Lie steps
    $\tilde{u}_b \leftarrow G_\theta(T_{h_b}^{\mathcal{A}} a_b)$, $\bar{u}_b \leftarrow T_{h_b}^{\mathcal{U}} \hat{u}_b$ ▷ two paths in equation 2
    $\widehat{\mathcal{L}}_{\text{orb}} \leftarrow B^{-1} \sum_{b=1}^B \|\tilde{u}_b - \bar{u}_b\|_{\mathcal{U}}^2 / (\epsilon_b^2 + \eta)$ ▷ equation 3
    $\mathcal{L} \leftarrow \mathcal{L} + \lambda_{\text{orb}} \widehat{\mathcal{L}}_{\text{orb}}$ ▷ orbit regularizer from equation 3
  end if
  Update θ using $\nabla_\theta \mathcal{L}$
end for

Ensure: Trained operator G_θ; inference uses only $a \mapsto G_\theta(a)$

defect by 56.3%. The paired LOCO-minus-augmentation deltas over the three matched seeds are -0.685 for force MAE with 95% CI $[-1.103, -0.267]$, -0.658 for orbit force MAE with 95% CI $[-1.067, -0.249]$, -0.0358 for ID relative L^2 with 95% CI $[-0.0576, -0.0139]$, -0.0348 for orbit relative L^2 with 95% CI $[-0.0555, -0.0141]$, and -0.304 for force equivariance defect with 95% CI $[-0.308, -0.299]$. The identity-output-action and shuffled-target ablations do not reproduce the defect reduction, and the physically correct LOCO row has lower defect than both controls on every matched seed.

A 1000-label sanity check gives the same qualitative conclusion with smaller predictive gains. With only augmentation and LOCO compared, force MAE is 18.4899 ± 0.0401 for augmentation and 18.4143 ± 0.0140 for LOCO; orbit force MAE is 18.6387 ± 0.0463 versus 18.4801 ± 0.0085; and force equivariance defect is 0.4556 ± 0.0074 versus 0.0983 ± 0.0032. The corresponding paired deltas are -0.0756 for force MAE with 95% CI $[-0.1406, -0.0106]$, -0.1586 for orbit force MAE with 95% CI $[-0.2527, -0.0646]$, and -0.3572 for defect with 95% CI $[-0.3677, -0.3467]$. This supports portability of the orbit-consistency target to atomistic vector outputs while keeping the main paper claim focused on PDE neural operators.

B Algorithm and Complexity

Algorithm 1 gives the stochastic training loop, and Table 11 summarizes the corresponding inference complexity.

Table 11: Inference complexity relative to the same base FNO. C_E denotes a learned frame estimator, C_A, C_U deterministic group actions, and R optional refinement iterations.

Method	Trainable params	Inference calls/modules	Inference cost
FNO	P_{NO}	one FNO	C_{NO}
FNO + aug.	P_{NO}	one FNO	C_{NO}
LOCO	P_{NO}	one FNO	C_{NO}
Analytic canonicalization	P_{NO}	transform + one FNO + transform	$C_A + C_{\mathrm{NO}} + C_U$
Canonicalization with estimator	$P_{\mathrm{NO}} + P_E$	estimator + transforms + one FNO	$C_E + C_A + C_{\mathrm{NO}} + C_U + RC_{\mathrm{refine}}$

Complexity. Let P_{NO} and $C_{\mathrm{NO}}(N)$ denote the parameter count and inference cost of the base neural operator G_θ on a grid with N spatial points. Let $C_A(N)$ and $C_U(N)$ denote the deterministic costs of applying the known input and output group actions. LOCO introduces no trainable inference-time modules, so

$$P_{\mathrm{LOCO}} = P_{\mathrm{NO}}, \qquad C_{\mathrm{inf}}^{\mathrm{LOCO}}(N) = C_{\mathrm{NO}}(N).$$

During training, with K sampled orbit transforms per minibatch, the orbit-consistency term costs

$$C_{\mathrm{orb}}^{\mathrm{train}}(N, K) = (K+1)C_{\mathrm{NO}}(N) + K\big(C_A(N) + C_U(N)\big) + C_{\mathrm{loss}}(N),$$

if the original prediction is not reused by another loss. When the supervised loss already computes $G_\theta(a)$, the incremental cost is

$$\Delta C_{\mathrm{orb}}^{\mathrm{train}}(N, K) = KC_{\mathrm{NO}}(N) + K\big(C_A(N) + C_U(N)\big) + C_{\mathrm{loss}}(N),$$

In the reported experiments $K = 1$, thus LOCO pays for one additional transformed operator evaluation during training and has no inference-time overhead relative to the base neural operator. We note that it is different for canonicalization pipelines: if $C_E(N)$ denotes a learned frame-estimator cost, $C_{\mathrm{can}}^{A}(N)$ and $C_{\mathrm{can}}^{U}(N)$ denote input and output canonicalization/restoration costs, and $RC_{\mathrm{refine}}(N)$ denotes optional test-time refinement, then a representative canonicalization pipeline has

$$C_{\mathrm{inf}}^{\mathrm{can}}(N) = C_E(N) + C_{\mathrm{can}}^{A}(N) + C_{\mathrm{NO}}(N) + C_{\mathrm{can}}^{U}(N) + RC_{\mathrm{refine}}(N).$$

LOCO removes the additive inference terms and deploys the unchanged map $a \mapsto G_\theta(a)$. Table 14 reports the measured training budget: the central FNO+aug. and FNO+aug.+orbit comparison is matched in batch-level model forward evaluations and differs by only 1.02× in wall-clock training time.

C PACE-style Observable-frame Canonicalization

PACE-FNO estimates a Lie-algebraic input frame, maps the input field to a canonical frame, applies an FNO in that frame, and restores the output prediction to the observation frame (Xu et al., 2026). To compare that approach with ours, we implement an observable-frame canonicalization baseline for our boosted N64 Galilean benchmark. For an input $a_c = (\omega_0, c_x, c_y)$, let $c = (c_x, c_y)$. The observable canonicalization baseline maps the input to the zero-boost frame, evaluates the trained FNO once, and restores the prediction:

$$\widehat{u}_{\mathrm{can}}(a_c) = T_c^{\mathcal{U}} G_\theta(T_{-c}^{\mathcal{A}} a_c). \tag{24}$$

Equivalently, it reads the explicit ambient-velocity channels, sets them to zero before the FNO call, and applies the analytic Galilean output shift by cT. This baseline is favorable to canonicalization because it uses the observed frame rather than a learned frame estimator. It also differs from PACE-FNO because it does not include joint estimator training, learned Lie-algebra coordinate prediction, or optional inference-time refinement. Table 12 reports the matched-seed observable-frame comparison across boost radii.

The canonicalized rows are constant across boost radii because analytic canonicalization removes the sampled Galilean frame before prediction and restores it after prediction; on the periodic grid, the output shift preserves the relative L^2 norm up to interpolation and numerical effects. Local orbit consistency remains

Table 12: Observable-frame canonicalization comparison on the N64 2D Navier–Stokes Galilean benchmark at 2% labels. Entries are orbit-OOD relative L^2, averaged over five matched seeds. Direct rows use the deployed FNO map unchanged. Observable-canonical rows add explicit inference-time pre/postprocessing using the observed boost channels. The LOCO rows use the fixed $\lambda_{\text{orb}} = 0.10$ checkpoints from the severity sweep. Lower is better.

Max boost	Aug. direct	Aug. + LOCO direct, $\lambda_{\text{orb}} = 0.10$	Aug. obs. canon.	Aug. + LOCO obs. canon., $\lambda_{\text{orb}} = 0.10$
0.10	0.5050 ± 0.0574	0.3715 ± 0.0349	0.4658 ± 0.0784	$\mathbf{0.3253 \pm 0.0564}$
0.20	0.5176 ± 0.0543	0.3840 ± 0.0331	0.4658 ± 0.0784	$\mathbf{0.3253 \pm 0.0564}$
0.25	0.5277 ± 0.0523	0.3943 ± 0.0318	0.4658 ± 0.0784	$\mathbf{0.3253 \pm 0.0564}$
0.35	0.5555 ± 0.0476	0.4242 ± 0.0285	0.4658 ± 0.0784	$\mathbf{0.3253 \pm 0.0564}$
0.50	0.6158 ± 0.0384	0.4910 ± 0.0216	0.4658 ± 0.0784	$\mathbf{0.3253 \pm 0.0564}$

the strongest unchanged-inference-map method: relative to compute-matched supervised augmentation, direct Aug.+LOCO reduces orbit-OOD error by 26.4%, 25.8%, 25.3%, 23.6%, and 20.3% as the maximum boost increases from 0.10 to 0.50. Observable canonicalization also helps, especially at larger shifts, but it changes the inference procedure by reading the frame and applying explicit pre/postprocessing. Direct Aug.+LOCO is better than Aug.+observable canonicalization through maximum boost 0.35, while observable canonicalization overtakes it at maximum boost 0.50. Applying observable canonicalization to the LOCO-trained checkpoint is best at every tested severity and reduces canonicalized OOD error by 30.2% relative to canonicalized augmentation. This supports the interpretation that LOCO and canonicalization are complementary: LOCO improves the learned operator itself, while frame canonicalization provides an additional inference-time correction when the symmetry frame is observable.

D Additional Controls and Robustness Experiments

In this section, we report additional experiments about the finite-difference interpretation of LOCO, the dependence on the stabilizer η and orbit step scale ϵ, and the generality of the method beyond periodic Galilean boosts. These experiments test whether the proposed finite-difference regularizer behaves as predicted, whether its numerical choices are robust, whether the effect persists for a distinct symmetry group, and how the method behaves when global symmetries are broken by non-periodic boundaries.

Tangent-propagation baseline. Lemma 1 suggests a close relationship between local orbit consistency and tangent propagation in the infinitesimal-step limit. To test whether the finite-difference form is empirically necessary, we trained tangent-propagation baselines on the Galilean N64 benchmark using the same low-label protocol. Pure tangent propagation achieved relative L^2 error 0.4902, orbit-OOD relative L^2 error 0.5349, and equivariance defect 0.3292. Combining supervised augmentation with tangent propagation improved these values to 0.4150, 0.4490, and 0.3157, respectively. In comparison, the finite-difference LOCO configuration at $\eta = 10^{-6}$ and $\epsilon = 0.25$ achieved relative L^2 error 0.3735, orbit-OOD relative L^2 error 0.3973, and equivariance defect 0.2614. Thus, although tangent propagation helps, the finite-difference objective gives a stronger control in this benchmark.

Sensitivity to η and ϵ. The finite-difference objective uses a stabilizer η in the normalized consistency loss. Since the intended interpretation assumes that the finite-difference scale dominates the stabilizer, we swept η at fixed orbit step scale $\epsilon = 0.25$. The results were essentially unchanged across $\eta \in \{10^{-8}, 10^{-6}, 10^{-4}\}$: relative L^2 error remained between 0.3735 and 0.3744, orbit-OOD relative L^2 error between 0.3973 and 0.3982, and equivariance defect between 0.2614 and 0.2623. This indicates that the reported behavior is not driven by a finely tuned stabilizer.

In contrast, the orbit step scale ϵ has a visible effect. At $\eta = 10^{-6}$, decreasing the step scale to $\epsilon = 0.125$ reduced the measured equivariance defect to 0.1999 but worsened relative and orbit-OOD error to 0.5149 and 0.5233. Increasing the step scale to $\epsilon = 0.5$ improved relative and orbit-OOD error to 0.2924 and 0.3480, while increasing the equivariance defect to 0.2753. This suggests that ϵ controls a bias–variance-like tradeoff:

Table 13: Architectural equivariance baseline on the D_4 vorticity benchmark. All values are mean $\pm$ standard deviation over five seeds. D4-GFNO enforces equivariance by construction, while LOCO keeps the standard FNO inference architecture unconstrained.

Method	Architecture	Rel. L^2	Orbit-OOD Rel. L^2	Equiv. Defect	Latency
Augmentation	FNO	0.06542 ± 0.00369	0.06538 ± 0.00366	0.04570 ± 0.00397	19.88 ± 5.97
LOCO + Augmentation	FNO	0.06462 ± 0.00331	0.06459 ± 0.00330	0.04499 ± 0.00383	21.93 ± 10.32
Architectural equivariance	D4-GFNO	0.09283 ± 0.00537	0.09283 ± 0.00537	$(7.88 \pm 0.32) \times 10^{-8}$	25.98 ± 1.19

smaller steps enforce more local equivariance, whereas larger steps provide stronger finite-orbit regularization for prediction and OOD generalization.

Discrete D_4 rotation/reflection benchmark. To test whether the method is specific to Galilean boosts, we added a two-dimensional benchmark with the discrete dihedral group D_4, consisting of rotations and reflections of the square grid. This is a non-Galilean symmetry and therefore probes a qualitatively different transformation family. Supervised augmentation alone achieved relative L^2 error 0.06450, orbit-OOD relative L^2 error 0.06448, and equivariance defect 0.04345. Adding local orbit consistency improved these values slightly to 0.06370, 0.06363, and 0.04269. The gain is modest, but consistent across the reported metrics, supporting the claim that LOCO is not tied to the Galilean setting.

Non-periodic boundary diagnostic. The main experiments use periodic domains, where the tested global symmetries act naturally. To probe a setting where global symmetry is broken, we added a one-dimensional heat-equation diagnostic with Dirichlet boundary conditions and a translation-like transformation. Without masking, applying orbit consistency across the full domain degraded performance relative to the supervised baseline: orbit-OOD relative L^2 error increased from 0.0325 to 0.0430, and equivariance defect remained comparable at 0.0241. When the consistency loss was restricted to the valid interior region using a spatial mask, orbit-OOD relative L^2 error improved to 0.0225 and equivariance defect dropped to 0.0109. This experiment shows both the expected failure mode near boundaries and a simple mitigation: local orbit consistency should be applied only where the assumed symmetry is locally valid.

Architectural equivariance baseline. We additionally compare LOCO against a hard architectural equivariance baseline on the D_4 vorticity benchmark. We implement a D_4 group-equivariant Fourier neural operator (D4-GFNO) using group Fourier layers over the regular representation of the dihedral group. Unlike LOCO, which leaves the inference architecture unchanged and encourages equivariance through a soft training regularizer, D4-GFNO enforces D_4 equivariance by construction. The architectural baseline indeed reduces the measured equivariance defect to numerical precision, but it obtains worse predictive and orbit-OOD error than the unconstrained FNO trained with augmentation or LOCO. This supports the intended design distinction: hard equivariance guarantees symmetry consistency, whereas LOCO preserves the standard FNO hypothesis class and can yield better accuracy in the low-label regime. This comparison is not parameter-matched; the D4-GFNO used here has fewer trainable parameters than the FNO baselines. Thus, the hard-equivariant architecture achieves the expected near-zero equivariance defect, but the soft-regularized FNO variants achieve substantially lower prediction error in this benchmark (Table 13).

E Training Budget

In Table 14 we include the training budget for the 2%-label runs.

Table 14: Training-budget for the 2%-label runs. Model forwards are batch-level forward evaluations. Sample columns count transformed training examples. Relative training time is normalized to FNO+aug. Latency is measured at inference.

Method	Opt. steps	Model fwds.	Bwd. passes	Aug. labeled samples	Orbit samples	Train min.	Rel. train time	Latency ms/sample
FNO	600	600	600	0	0	14.6 ± 0.4	$0.71\times$	16.53 ± 2.13
FNO + orbit	600	1200	600	0	9600	18.7 ± 0.6	$0.91\times$	16.11 ± 1.66
FNO + aug.	900	1800	900	14400	0	20.6 ± 2.5	$1.00\times$	18.76 ± 7.34
FNO + aug. + orbit	600	1800	600	9600	9600	21.0 ± 2.6	$1.02\times$	16.87 ± 3.19