FLOW EQUIVARIANT VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many visual sequences exhibit a geometric flow structure, where time-parameterized Lie-group actions describe motion and elements of the associated Lie algebra generate different flows. Standard self-attention does not inherently respect this structure, potentially limiting generalization across moving reference frames with geometric symmetries. In this work, we illustrate how this problem manifests in vision transformers, causing them to process motion-related sequences in an unrelated fashion. We then introduce a provably flow-equivariant vision transformer that generalizes to motion regimes absent from training. By connecting the geometry of continuous motion to patch-based visual representations, this framework provides an inductive bias for video understanding and spatiotemporal forecasting. Empirically, we demonstrate improved performance on both simple visual motion forecasting as well as PDE forecasting for physical systems with natural flow symmetries.

1 INTRODUCTION

We perceive the world as a continuous stream rather than a collection of isolated snapshots. As we walk, turn our heads, or passively observe a moving object, stimuli drift slowly across our field of view, and the entire world seems to be moving around us. Yet, we have no trouble recognizing that the world itself has not fundamentally changed – that the car that was just beside us a moment ago is the same car that is now receding into the distance ahead of us. What changes is our viewpoint, and as that viewpoint evolves continuously over time, it induces structured, smooth transformations, or *flows*, governed by an exact mathematical structure. Formally, these flows can be described by one-parameter Lie subgroups where different flows correspond to different elements of an underlying structured Lie algebra.

Deep neural network architectures that hard-code a relevant symmetry are called *equivariant*, and have been shown to generalize from less data and even scale more favorably than comparable unconstrained counterparts (Cohen and Welling, 2016; Kondor and Trivedi, 2018; Cohen et al., 2019; Bronstein et al., 2021; Ngo and Ravanbakhsh, 2026). Recently, Keller (2025) introduced flow equivariance as the natural extension of equivariant deep learning to such time-parameterized symmetries, and demonstrated that unconstrained temporal integration, such as standard recurrence, generally breaks flow equivariance.

In modern sequence modeling, however, Transformers have largely displaced recurrent networks as the dominant architecture. This raises the corresponding question for attention-based models: do Transformers similarly break the geometric structure necessary to accurately represent flow transformations? In other words, are Transformers generally flow equivariant? In this work, we show that they are not, and that this extends naturally to patch-based Vision Transformers (ViTs) (Dosovitskiy et al., 2021). However, as with the Flow Equivariant RNN (FERNN) (Keller, 2025), flow equivariance can be recovered in such architectures by making the relevant operations group-equivariant and operating on a flow-lifted hidden state. Our work therefore connects to prior work on equivariant attention and equivariant Transformers, including Lie-group self-attention and SE(3)/E(3)-equivariant Transformer architectures (Hutchinson et al., 2021; Fuchs et al., 2020; Liao and Smidt, 2023), while focusing specifically on symmetries whose action evolves continuously through time, rather than 'statically' in an identical manner over all timesteps.

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.

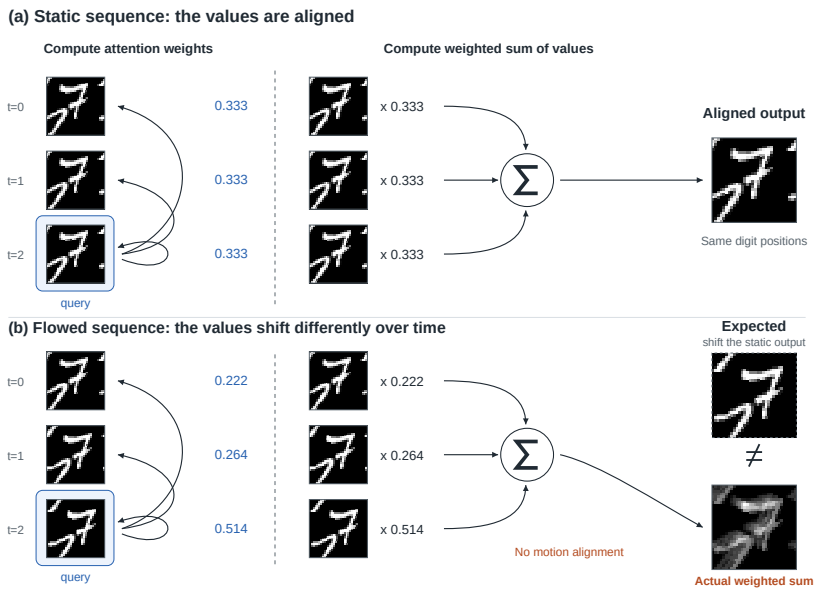


Figure 1: **Standard self-attention is not flow equivariant.** In order to satisfy equivariance, the layer must commute with the transformation of interest. In this figure, we demonstrate that self-attention does not commute with time-parameterized flow transformations such as visual motion: the output for a flowed input sequence (bottom) is not equivalent to flowing the output of a static sequence (top), i.e. $\mathrm{SA}(\psi(\nu) \cdot f)_t \neq \psi(\nu)_t \cdot \mathrm{SA}(f)_t$. Simply, attention weights change when the input is in motion compared to the stationary setting. We remedy this via lifting (Section 3).

In the following, we provide a brief review of 'static' group equivariance and distinguish it from flow equivariance, which relates more naturally to sequences and transformations that evolve over time – the new frontier for today's video and world models. We then demonstrate how standard self-attention in the Transformer architecture is not generically flow equivariant, and introduce the *Flow Equivariant Vision Transformer* which solves this by operating in a structured manner over a lifted flow dimension. We provide a general construction amenable to generic visual sequence modeling, and demonstrate the improved performance on both simple visual motion forecasting, as well as PDE forecasting for physical systems with natural flow symmetries.

2 Background

In the following, we introduce traditional spatial equivariance and flow equivariance before specializing to the periodic image grid used by our patch-based model.

Space-time signals and group actions. We consider discrete-time space-time signals $f \in \mathcal{F}_K(X, \mathbb{Z})$, where $\mathcal{F}_K(X, \mathbb{Z}) := \{f : X \times \mathbb{Z} \to \mathbb{R}^K\}$ denotes the space of K-channel signals defined over a 'spatial' domain X and discrete time $t \in \mathbb{Z}$. We write $f_t(x) := f(x, t)$ for the signal at time t, and in practice restrict to finite sequences with $t \in \{0, \ldots, T-1\}$. We assume that the signal domain X admits the action of a symmetry group G. For $g \in G$, the corresponding left action on a single time-slice is written

$$(g \cdot f_t)(x) := f_t(g^{-1} \cdot x), \qquad g \in G,\ x \in X. \tag{1}$$

This follows standard convention: transforming the signal by g is equivalent to evaluating the original signal at the inverse-transformed coordinate (Cohen and Welling, 2016).

Spatial equivariance. A static transformation applies the same group element at every time point, i.e. $(g \cdot f)_t := g \cdot f_t$, where we use the notation $(g \cdot f)_t$ to denote applying the group element to the entire space-time signal first and then indexing the signal at time t. An operator E between signal spaces is spatially equivariant to the action of a group G if transforming its input and then applying the operator gives the same result as transforming its output, i.e., $E[g \cdot f] = g \cdot E[f]\ \forall g \in G$.

Flows. Following Keller (2025), we view motion as a *flow* over space X: a one-parameter subgroup of G generated by an element ν of the corresponding Lie algebra $\mathfrak{g}$. We work in the matrix Lie group setting of Hall (2015), for which the exponential map $\exp : \mathfrak{g} \to G$ is well defined. We denote the time-t group element generated by ν as $\psi_t(\nu) := \exp(t\nu) \in G$. By construction, $\psi_0(\nu) = e$ and $\psi_s(\nu)\psi_t(\nu) = \psi_{s+t}(\nu)$, satisfying the one-parameter group law. For a sequence $f \in \mathcal{F}_K(X, \mathbb{Z})$, the action of the full flow $\psi(\nu)$ is defined frame-wise by

$$\big(\psi(\nu) \cdot f\big)_t(x) := f_t\big(\psi_t(\nu)^{-1} \cdot x\big)\,. \tag{2}$$

Throughout, we use $\psi(\nu) \cdot f$ to denote the action of the entire time-parameterized transformation on the sequence, while $\psi_t(\nu)$ denotes the instantaneous group element acting at time t. In this work, we primarily consider translation flows on images (although the framework naturally generalizes to arbitrary abelian Lie groups, as in Keller (2025)). We therefore consider the group of two-dimensional translations, and restrict the boundary conditions to be periodic for simplicity, matching our synthetic datasets. With S pixels along each axis, the periodic pixel grid is $X = (\mathbb{Z}/S\mathbb{Z})^2$ and an integer velocity $\nu \in \mathbb{Z}^2$ is measured in pixels per frame. At integer observation times,

$$\psi_t(\nu) = t\nu \pmod{S}, \qquad (\psi(\nu) \cdot f)_t(x) = f_t(x - t\nu). \tag{3}$$

In practice, we consider a finite set of flow generators $\mathcal{V} \subset \mathfrak{g}$ When building our datasets and constructing our models. We write $\mathcal{V}_{\text{train}}$ and $\mathcal{V}_{\text{test}}$ for the sets of generators used when constructing the training and evaluation sequences, and $\mathcal{V}_{\text{model}}$ for the set of generators our model is built to be equivariant with respect to. The conditions under which the resulting finite model is flow-equivariant are stated in Section 3.6.

Flow equivariance. A sequence model $\Phi : \mathcal{F}_K(X, \mathbb{Z}) \to \mathcal{F}_{K'}(Y, \mathbb{Z})$, where, as for X, the output domain Y also admits an action of G, is *flow-equivariant* with respect to $\mathcal{V} \subseteq \mathfrak{g}$ if

$$\psi(\nu) \cdot \Phi[f] = \Phi[\psi(\nu) \cdot f], \quad \forall \nu \in \mathcal{V},\ f \in \mathcal{F}_K(X, \mathbb{Z}). \tag{4}$$

In words, applying a flow to the input sequence and then processing it with the model must produce the same result as first processing the original sequence and then applying the corresponding flow to the output sequence. In Figure 1, we provide a simple example demonstrating that standard self-attention is not generally flow-equivariant. Intuitively, this can be understood by observing that standard self-attention applied to a static sequence will have the same attention weights between all elements of the sequence (the attended elements are all identical). When undergoing a visual motion flow, these attention weights naturally change as sequence elements move out of alignment. The aligned output in the static setting therefore cannot be simply flowed to match the linear combination of unaligned frames in the moving setting, and therefore the operation is not flow equivariant.

Flow Equivariant Forecasting. Our experiments use a one-step predictor rather than a sequence-to-sequence readout. Let $I_T := \{0, \dots, T-1\}$ be the observation times and $\mathcal{F}_K(X, I_T)$ the corresponding space of input sequences. Writing $\mathcal{F}_K(X) := \{g : X \to \mathbb{R}^K\}$ for the space of single frames, the predictor is $\Phi_\theta : \mathcal{F}_K(X, I_T) \to \mathcal{F}_K(X)$ and outputs the frame at time T. Its desired equivariance property is

$$\Phi_\theta[\psi(\nu) \cdot f] = \psi_T(\nu) \cdot \Phi_\theta[f]. \tag{5}$$

The output transforms at the predicted time T, not at the last observation time $T-1$.

3 The Patch-Based Flow Equivariant Transformer

To build our patch-based flow equivariant vision transformer, we build off of the Flow Equivariant Transformer (FET) construction of Khaliliya and Keller (2026), extending it to patch-based visual forecasting. Specifically, we preserve the inverse-flow lift and weight sharing across moving frames as the key mechanism to induce equivariance. Primarily, we replace the convolutional blocks in the original architecture with patch embeddings and spatio-temporal attention, bringing the architecture in line with modern Vision Transformers, and especially those built for video, such as the ViViT (Arnab et al., 2021). In addition, the pooled classification head is replaced by a next-frame readout in order to support visual sequence forecasting. In Appendix B, we describe a more sophisticated flow-mixing construction of self-attention and distinguish it from the variant evaluated here.

Let S be the image width and height in pixels, so that $X = (\mathbb{Z}/S\mathbb{Z})^2$ is an $S \times S$ periodic pixel grid. Let $I_T := \{0, \dots, T-1\}$ be the T observation times. We write $\mathcal{F}_K(X) := \{g : X \to \mathbb{R}^K\}$

to denote the set of K-dimensional functions defined over the pixel grid X, and $\mathcal{F}_K(X, I_T) := \{f : X \times I_T \to \mathbb{R}^K\}$ to denote the time-indexed sequences of such functions, i.e. spatiotemporal signals. Explicitly, an input $f \in \mathcal{F}_K(X, I_T)$ contains T frames with K channels: $f_t := f(\cdot, t)$ is a whole frame, whereas $f_t(x)$ is its value at a pixel.

While the transformer backbone outputs a hidden token for every input patch and observation time, and therefore supports sequence-to-sequence readouts, our experiments use a one-step autoregressive readout: we decode only the last hidden time slice to predict the next frame. With learned parameters θ, our one-step model predicts a frame at time T, $\Phi_\theta : \mathcal{F}_K(X, I_T) \to \mathcal{F}_K(X)$, and we iterate this predictor on a sliding context window for longer forecasts.

3.1 Translation flows and the lift.

Flow lifting represents a sequence in multiple moving reference frames simultaneously (Keller, 2025). In this work, we focus on periodic translations, and therefore only define the lift $L[f]$ in this setting. The analogous lift can be defined for any commutative Lie-group action, but the patch construction below is specific to the discrete image grid. A velocity $\nu \in \mathbb{Z}^2$ is measured in pixels per frame. Its time-t translation is written as $\psi_t(\nu) := t\nu \pmod S$, and the notation $\psi(\nu) \cdot f$ denotes the action on the entire sequence:

$$(\psi(\nu) \cdot f)_t(x) := f_t(x - t\nu), \qquad L[f]_t(\nu, x) := f_t(x + t\nu). \tag{6}$$

Thus $L[f]_t(\nu, \cdot) = \psi_t(-\nu) \cdot f_t$ is frame t viewed in the reference frame moving with velocity ν, where we store a finite velocity bank $\mathcal{V}_{\text{model}} \subset \mathbb{Z}^2$. For each lifted 'velocity channel', let $q_t^\nu(x) := L[f]_t(\nu, x)$ and $q^\nu := (q_0^\nu, \dots, q_{T-1}^\nu)$ denote the entire time sequence. Thus $q^\nu \in \mathcal{F}_K(X, I_T)$ is the input viewed in frame ν. The order differs from the convolutional construction in Khaliliya and Keller (2026), which lifts features after a spatially equivariant embedding. Here we lift the pixel frames *before* patch embedding: A fixed nonoverlapping patch embedding does not commute with arbitrary pixel translations. We handle this patch-grid dependence by phase averaging in Section 3.4.

3.2 Patch embedding and transformer blocks

For an input sequence $q = q^\nu$, the same network processes each moving frame independently; no hidden layer mixes velocity slices in this variant, and parameters are shared between velocities (thereby enabling velocity generalization). Let p be the patch width, with p dividing S, and let $Y = (\mathbb{Z}/(S/p)\mathbb{Z})^2$ be the patch grid. A learned affine embedding $\phi : \mathcal{F}_K(X) \to \mathcal{F}_C(Y)$ maps each nonoverlapping $p \times p$ patch to C channels, with the same affine map at every patch and time, as in the Vision Transformer (Dosovitskiy et al., 2021). We keep the hidden width C fixed across layers. Write $h_t^0(y) := \phi[q_t](y)$ and $h^0 := (h_0^0, \dots, h_{T-1}^0)$ to denote the embedded patch sequence. The sequence is updated at each layer by $h^{l+1} := B_l[h^l]$ for $l = 0, \dots, N-1$. Each block $B_l : \mathcal{F}_C(Y, I_T) \to \mathcal{F}_C(Y, I_T)$ applies spatial attention, temporal attention and a width-$4C$ GELU MLP, following a canonical video Transformer construction (Arnab et al., 2021). Using pre-norm residual sublayers (Xiong et al., 2020), the updates are

$$\begin{aligned} \bar{h}^l &:= h^l + \mathrm{SA}_{\mathrm{space},l}[\mathrm{LN}_{l,1}[h^l]], \\ \widetilde{h}^l &:= \bar{h}^l + \mathrm{SA}_{\mathrm{time},l}[\mathrm{LN}_{l,2}[\bar{h}^l]], \\ h^{l+1} &:= \widetilde{h}^l + \mathrm{MLP}_l[\mathrm{LN}_{l,3}[\widetilde{h}^l]]. \end{aligned} \tag{7}$$

Each $\mathrm{LN}_{l,r}$ is a separately parameterized LayerNorm (Ba et al., 2016) over channels at each token, never over space, time or the velocity bank. The MLP acts independently at each token with shared affine maps $C \to 4C \to C$ and a GELU between them. The attention sublayers use scaled dot-product attention (Vaswani et al., 2017) with separate parameters for space and time.

Spatial attention uses periodic rotary position embeddings (Su et al., 2021; Heo et al., 2024): queries and keys are rotated using integer harmonics so that their inner products depend on relative patch positions, including across the periodic boundary. Temporal attention similarly uses relative-time rotations at each patch, with separate projection parameters. It is unmasked within the observed context and receives no future targets. The complete rotation matrices and attention equations are given in Appendix A.

3.3 RESIDUALS AND LOGITS.

After the N blocks, a final channel-only LayerNorm and affine decoder unpatchify the last hidden frame into two outputs: a K-channel residual $\delta_\theta^0[q] \in \mathcal{F}_K(X)$ and a scalar gating-logit map $s_\theta^0[q] \in \mathcal{F}_1(X)$. Explicitly, the network is thus

$$\mathcal{B}_\theta : \mathcal{F}_K(X, I_T) \to \mathcal{F}_K(X) \times \mathcal{F}_1(X), \qquad \mathcal{B}_\theta[q] := (\delta_\theta^0[q], s_\theta^0[q]). \tag{8}$$

The superscript 0 denotes the unshifted patch grid, not a time or layer index – we describe the shifting operating next. The scalar $s_\theta^0[q](x)$ is a learned gating logit. All embedding, block and decoder parameters are shared across velocities.

3.4 RESTORING PIXEL-TRANSLATION EQUIVARIANCE.

A fixed patch grid respects translations by whole patches, but a one-pixel shift generally changes which pixels share a patch. We remove this dependence on the patch origin by processing all p^2 possible grid offsets, or *phases*, with the same network. This uses the group-averaging principle for equivariant maps (Puny et al., 2022, Equation 1), reduced to one representative per patch phase because the backbone already respects whole-patch translations. Let $\mathcal{A} := \{0, \dots, p-1\}^2$ be the set of all 2-dimensional shifts within a patch. For each $a \in \mathcal{A}$, translate the input by $-a$, apply the network, and translate its output back by a before averaging:

$$\begin{aligned} \delta_\theta[q](x) &:= \frac{1}{p^2} \sum_{a \in \mathcal{A}} \delta_\theta^0[(-a) \cdot q](x - a), \\ s_\theta[q](x) &:= \frac{1}{p^2} \sum_{a \in \mathcal{A}} s_\theta^0[(-a) \cdot q](x - a), \qquad D_\theta[q](x) := q_{T-1}(x) + \delta_\theta[q](x). \end{aligned} \tag{9}$$

The evaluation at $x-a$ shifts each output back to the original pixel coordinates; the average therefore combines aligned predictions, not different spatial locations. A translation by $b \in X$ permutes the phases modulo p. Any remaining displacement is a whole-patch translation, which the single-grid network respects because its weights are shared and its spatial RoPE is periodic. Consequently,

$$D_\theta[b \cdot q](x) = D_\theta[q](x - b), \qquad s_\theta[b \cdot q](x) = s_\theta[q](x - b). \tag{10}$$

This is translation *equivariance*, not invariance: shifting the input shifts the predicted image and logits by the same amount, rather than leaving them unchanged. Phase averaging removes the spatial symmetry breaking caused by a fixed patch origin. We note that this does increase computational complexity proportional to patch size, yet for the small patches we find this increase to be manageable. We believe that other methods developed to make ViTs translation equivariant could alleviate some of this burden, and would be amenable to this framework (Rojas-Gomez et al., 2023). We leave further exploration of such efficient implementations to future work.

3.5 OUTPUT HEAD.

Finally, the choice of output head is critical to match the desired equivariance or invariance of the task at hand. In (Khaliliya and Keller, 2026) the authors consider an invariant sequence classification task, and therefore pool over space, time and velocity to produce a single classification score that generalizes over different flows automatically. In this work, we instead focus on the equivariant task of sequence forecasting, which requires a spatially resolved output that transforms appropriately at each future point in time. We therefore retain the moving-frame predictions until after decoding and 'unlift' them before combining them. In practice, for our translation flows, this 'unlifting' is accomplished by simply shifting each moving frame's prediction and gating logits back to the observation frame at time T, thereby 'aligning' the different moving frames after processing. The candidate $o_T^\nu[f](x) \in \mathbb{R}^K$ and gating logit $\ell_T^\nu[f](x) \in \mathbb{R}$ are thus given by

$$o_T^\nu[f](x) := D_\theta[q^\nu](x - T\nu), \qquad \ell_T^\nu[f](x) := s_\theta[q^\nu](x - T\nu), \tag{11}$$

$$\alpha_T^\nu[f](x) := \frac{\exp(\ell_T^\nu[f](x))}{\sum_{\mu \in \mathcal{V}_{\text{model}}} \exp(\ell_T^\mu[f](x))}, \qquad \Phi_\theta[f](x) := \sum_{\nu \in \mathcal{V}_{\text{model}}} \alpha_T^\nu[f](x) o_T^\nu[f](x). \tag{12}$$

The softmax over velocities at each pixel produces the mixing weights $\alpha_T^\nu[f](x)$ to combine the decoded predictions between velocity channels; it is not attention between hidden velocity slices. We

denote the predicted frame by $\widehat{f}_T := \Phi_\theta[f]$ and its value at pixel x by $\widehat{f}_T(x) = \Phi_\theta[f](x)$; hats denote predicted frames. We use $S = 32$, $p = 2$, $C = 64$, $N = 2$, four heads and $\mathcal{V}_{\text{model}} = \{-2, \dots, 2\}^2$ (25 slices) for all experiments. Flow-MNIST has $K = 1$, $T = 10$. The matched ViT predicts $D_\theta[f]$ directly, without a lift or velocity pooling.

3.6 Equivariance.

We can demonstrate simply that this network is indeed flow equivariant, following the construction in Khaliliya and Keller (2026). Specifically, let $\eta \in \mathbb{Z}^2$ be an additional input velocity. For the ideal lift (on an infinite 'untruncated' set $\mathcal{V}_{\text{model}}$), a flow applied to the input by definition yields a shift of the velocity index after performing the lift:

$$L[\psi(\eta) \cdot f]_t(\nu, x) = f_t\big(x + t(\nu - \eta)\big) = L[f]_t(\nu - \eta, x). \tag{13}$$

Therefore, the lift itself introduces the velocity channels, and then instantiates the regular representation of the flow action on the lifted representation. Since all subsequent reindexing permutes all stored slices, the shared backbone simply reindexes its outputs in an analogous manner. Unlifting in the output head then shifts both candidate images and logits by $T\eta$ as desired. The velocity softmax respects the same reindexing, so a change of summation index in Equation 12 gives

$$\Phi_\theta[\psi(\eta) \cdot f](x) = \Phi_\theta[f](x - T\eta) = \big(\psi_T(\eta) \cdot \Phi_\theta[f]\big)(x). \tag{14}$$

In practice, our velocity set $\mathcal{V}_{\text{model}}$ is not closed under arbitrary velocity shifts, but instead incurs equivariance error at the boundaries, identical to the truncation error observed in (Keller, 2025; Khaliliya and Keller, 2026). The finite pooled model therefore has no general exact flow-equivariance guarantee on the full Lie algebra, only within an interior domain of the supported velocity set. In practice, we observe this to be sufficient to demonstrate the hallmark benefits of equivariance, including improved performance on datasets with the corresponding flow symmetries, and improved generalization to both new unobserved flows, and longer sequence lengths than those seen during training. In the following we demonstrate the empirical results supporting these observations.

4 Experiments

In the following, we study two autoregressive forecasting tasks: translating Flowing-MNIST and periodic, unforced Navier–Stokes. We include Flowing-MNIST as a simple video baseline concretely demonstrating the expected generalization and performance benefits of our construction, and ultimately validating the correctness of our approach. We further include the Navier–Stokes PDE forecasting due to the equations' natural symmetry with respect to translation flows (i.e. Galilean symmetry). Following the evaluation of Keller (2025), we examine prediction accuracy, rollout lengths beyond the training horizon, and generalization to unseen motion. The main comparisons use five seeds with with 10,000 updates for MNIST and 10,240 for Navier–Stokes. Longer training and fixed-checkpoint interventions are reported separately in Appendix E.

Shared evaluation. All models use the same training data and update budget within each task. Checkpoints are selected on 64 mixed-motion validation sequences using the training-horizon objective; diagnostics use 128 disjoint validation sequences per condition, with all five seeds retained. Training velocities lie in $\{-1, 0, 1\}^2$, the FET bank is $\mathcal{V}_{\text{model}} = \{-2, \dots, 2\}^2$, and fixed-velocity diagnostics cover $\{-3, \dots, 3\}^2$. We distinguish unseen velocities inside this bank from velocities outside it. Tables report MSE averaged over forecast steps 1–30, whereas rollout curves show MSE at each step. Conditions are averaged equally within each regime before averaging across seeds; error bars in tables are sample standard deviations over seeds. Dataset construction, splits, optimization, and additional plots are in Appendices C–E.

Table 1: **Forecasting accuracy.** Mean $\pm$ sample std of 30-step-average validation MSE, in units of 10^{-3} (lower is better). Mixed ID (In-Distribution) uses the training motion distribution; unseen-in-bank and outside-bank columns average 16 and 24 fixed global velocities, respectively.

Dataset	Model	Mixed ID	Unseen in bank	Outside bank
Flow-MNIST	FET	0.996 ± 0.16	0.0753 ± 0.026	152 ± 11
	ViT	14.2 ± 0.64	255 ± 120	299 ± 140
Navier–Stokes	FET	0.452 ± 0.042	28.1 ± 11	194 ± 53
	ViT	99.9 ± 76	243 ± 47	611 ± 140
	Mean FET	0.47 ± 0.13	0.47 ± 0.13	0.47 ± 0.13
	Mean ViT	234 ± 490	234 ± 490	234 ± 490
	Mean FNO	0.0795 ± 0.0031	0.0795 ± 0.0031	0.0795 ± 0.0031

4.1 Flowing-MNIST Next-Step Prediction

As an initial baseline dataset to verify the correctness of our construction, we demonstrate the expected velocity generalization and length extrapolation abilities of the flow equivariant vision transformer on Flowing-MNIST, while simultaneously demonstrating that standard baseline ViTs and convolutional networks fail to achieve the same performance.

Task and models. We construct sequences of two MNIST digits (LeCun et al., 1998) translating on a periodic 32×32 canvas, adapting the flowing-digit forecasting task of Keller (2025). The two digits have independently sampled velocities during training. Models receive ten frames and learn a one-step predictor unrolled for ten future frames autoregressively; at evaluation we generate 30 frames to test length generalization. We compare patch FET with the matched patch ViT backbone, without the inverse-flow lift or velocity pooling. Both use 2×2 patches, width 64, two blocks and four heads. A separate, two-seed comparison with the original non-patch FET from Khaliliya and Keller (2026) is given in Appendix E.

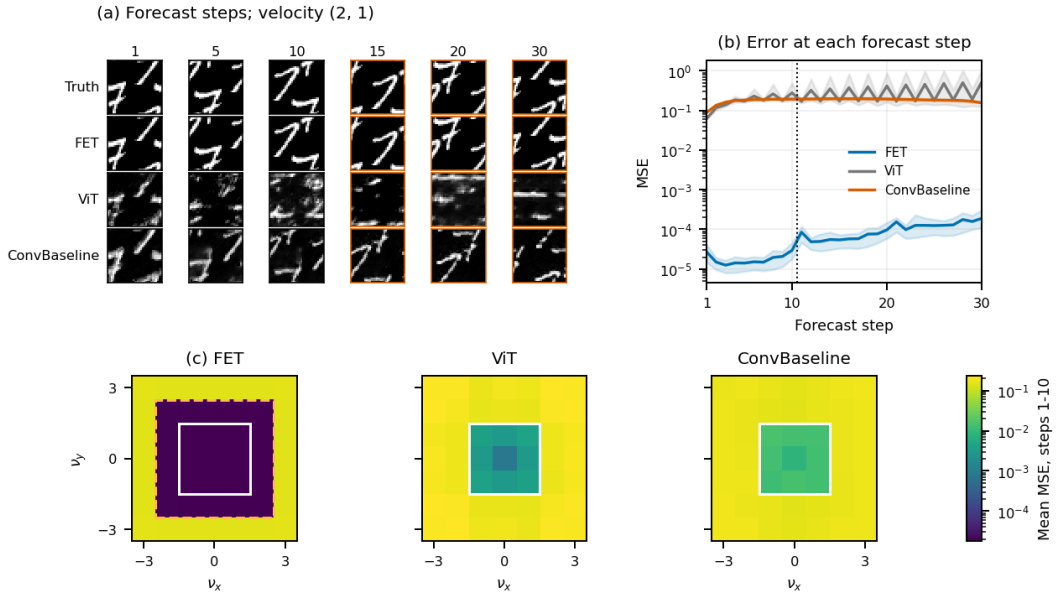


Figure 2: **Flow-MNIST rollout and velocity generalization.** (a) Ground truth and free-running predictions for a fixed validation example at velocity $(2, 1)$; orange borders mark steps beyond the ten-step training horizon. (b) Per-step MSE at the same velocity, averaged over 128 examples and five seeds; shading shows the seed range, not a confidence interval. (c) Five-seed 30-step-average MSE across velocities. White boxes mark training support (the Mixed ID range); dashed red boxes mark the raw FET bank (the 'in bank' set), and 'outside bank' includes all velocities.

Prediction and motion generalization. As expected, we see that the FET reduces mixed-ID 30-step-average MSE from 1.42×10^{-2} to 9.96×10^{-4}, a $14.3\times$ reduction relative to ViT (Table 1), suggesting that the model is better able to model in-distribution velocities better than a comparable non-equivariant model. On unseen global velocities within the bank, FET maintains low error (7.53×10^{-5}), whereas ViT reaches 2.55×10^{-1}, demonstrating that the construction indeed supports generalization to velocities unseen during training – a capability which the baseline model clearly lacks. Figure 2 shows that the advantage persists beyond the training rollout length.

4.2 Navier–Stokes Forecasting

As a second, more complex, sequence forecasting task, we evaluate our model on forecasting a well-known PDE which has the same flow symmetry that is built into our model. Specifically, we evaluate our models by their ability to forecast Navier–Stokes dynamics, a system which naturally exhibits a Galilean (translation-flow) symmetry – i.e., a solution to these equations is also a solution when it undergoes a global translation flow.

Task We predict two-component ($K = 2$) velocity fields from the two-dimensional incompressible, unforced Navier–Stokes system on a periodic 32×32 grid with viscosity 10^{-3}. We introduce additional constant global drifts to produce a dataset of Galilean-related trajectories. To ensure the symmetry is properly represented by the trajectories, we transform both the spatial coordinates and the velocity values for each example. The training set contains 1,024 base trajectories. Models receive six frames, are trained through four predicted steps, and are evaluated for 30 steps. The numerical solver and initial-condition distribution are specified in Appendix C. In the following, $f_t(x) \in \mathbb{R}^2$ is a physical velocity vector with 2 coordinates, $\ell_{\mathrm{box}} = 2\pi$ is the physical box width, $\Delta x = \ell_{\mathrm{box}}/S$ is the grid spacing, and $\Delta t = 1/4$ the time between observations. A pixel velocity ν corresponds to physical drift $c_\nu := \beta\nu$, where $\beta := \Delta x/\Delta t = \pi/4$.

Galilean lift. Unlike image translation, a Galilean transformation of a velocity field shifts both coordinates and velocity values. We denote its sequence action by $(\Gamma_\nu[f])_t(x) := f_t(x - t\nu) + c_\nu$. Its inverse gives the moving-frame input (the lift), and its forward action restores the prediction:

$$L_{\mathrm{NS}}[f]_t(\nu, x) := f_t(x + t\nu) - c_\nu, \tag{15}$$

$$q_t^\nu(x) := L_{\mathrm{NS}}[f]_t(\nu, x), \qquad o_T^\nu[f](x) := D_\theta[q^\nu](x - T\nu) + c_\nu. \tag{16}$$

Here q^ν replaces the image-lifted input in Equations 11–12. The scalar gating logits are shifted spatially but receive no additive velocity offset.

Mean-frame control. As a strong baseline, we include a mean-velocity estimate from the trajectories itself, and consider all models evaluated on the same sequences where this mean-velocity has been removed prior to processing. Specifically, we estimate this mean velocity from the last observed frame by averaging over spatial positions:

$$\bar{u} := \frac{1}{S^2} \sum_{x \in X} f_{T-1}(x) \in \mathbb{R}^2, \qquad \bar{\nu} := \bar{u}/\beta \in \mathbb{R}^2. \tag{17}$$

Since the estimate $\bar{\nu}$ need not be an integer, we implement partial-pixel shifts via the periodic Fourier interpolant of a grid field g, evaluated at real pixel coordinates. We denote this operation $\mathcal{I}[g]$. The implementation takes the real part of the inverse Fourier transform. Let $g_t(x) := f_t(x) - \bar{u}$. With a one-step predictor $H_\theta : \mathcal{F}_2(X, I_T) \to \mathcal{F}_2(X)$, we define

$$\widetilde{f}_t(x) := \mathcal{I}[g_t](x + t\bar{\nu}),$$

$$\Phi_{\mathrm{mean},\theta}[f](x) := \mathcal{I}\big[H_\theta[\widetilde{f}]\big](x - T\bar{\nu}) + \bar{u}. \tag{18}$$

Below, we compare our patch-based Flow Equivariant Vision Transformer (denoted FET for simplicity) and ViT with mean-frame variants whose predictor H_θ is the same FET, ViT, or a Fourier neural operator (Li et al., 2021) (FNO; width 32, four layers, eight modes). The mean-frame transformation adds no learned parameters and uses no target velocities. It removes the bulk drift before prediction, but is not an exact fractional-shift equivariance guarantee.

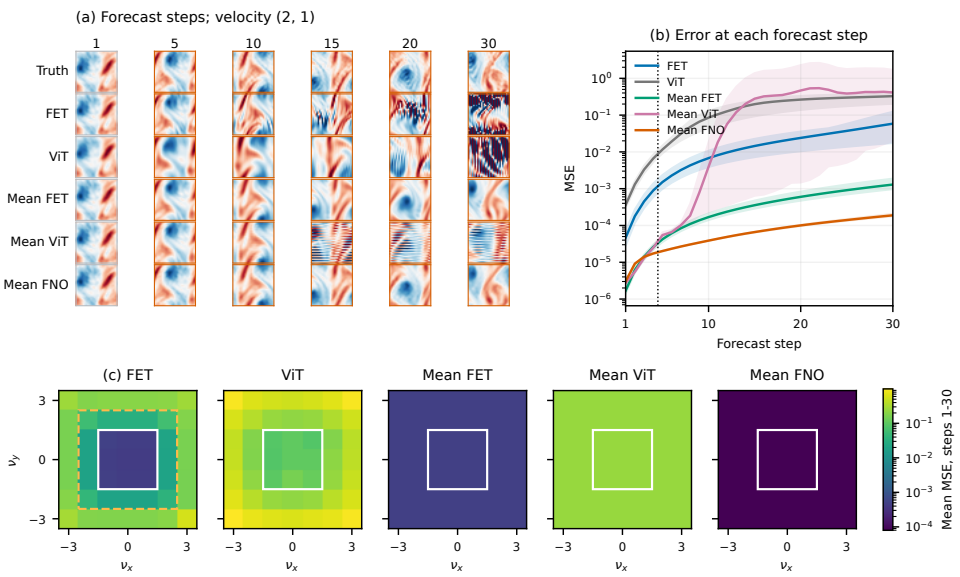


Figure 3: **Navier–Stokes rollout and velocity generalization.** Images show vorticity computed from the predicted velocity, using one ground-truth color range; extreme predictions saturate the display, yielding black pixels. MSE is computed on the unclipped two-component velocity fields. Mean-frame controls are approximately insensitive to the imposed integer boosts, demonstrated by the uniform MSE over velocities, while the raw FET degrades toward the bank boundary due to truncation boundary effects, and the ViT underperforms especially after the training length.

Accuracy and robustness to drift. In Table 1, we see the patch based Flow Equivariant Vision Transformer attains 4.52×10^{-4} mixed-in-distribution MSE, compared with 9.99×10^{-2} for ViT. However, its error rises to 2.81×10^{-2} on unseen in-bank velocities and 1.94×10^{-1} outside the bank. Thus, while there is certainly some improved velocity generalization, the boundary effects are larger than on MNIST. The Mean-frame FET variant maintains approximately 4.70×10^{-4} across all three motion regimes (Table 1, Figure 3), dramatically outperforming the comparable Mean-Frame ViT. Removing the bulk drift velocity is therefore insufficient to fully account for the benefit of flow equivariance on this task, highlighting the potential for future flow equivariant PDE modeling development. Removing the observed bulk drift therefore greatly improves robustness to unseen boosts, although it does not improve raw FET's mixed-ID mean. We include the mean-frame FNO as the strongest accuracy baseline, with MSE 7.95×10^{-5}, about $5.9\times$ lower than mean-frame FET. In future work, we intend to explore model variants that include flow equivariance in combination with state-of-the-art architectures such as the FNO. Finally, we see the Mean-frame ViT remains highly seed-sensitive, so mean normalization alone does not guarantee stable learned dynamics.

5 Conclusion

In this work, we have introduced the Flow Equivariant Vision Transformer and demonstrated improved forecasting performance and zero-shot generalization to velocities unseen during training. We believe time-parameterized symmetries to be an under-explored aspect of data structure in modern sequence modeling and thus hope that this work draws increased attention to model architectures which can exploit this structure, and further serves as a starting point for future development.

Limitations. The main limitation of our construction is the increased computational complexity and memory usage compared with the baseline ViT, scaling linearly with respect to patch size and velocity-bank size. While the current formulation may thus scale less favorably than existing models, the zero-shot generalization and sequence forecasting capabilities demonstrated by our framework are clear and significant benefits which we believe warrant further investigation. We therefore present this work as a preliminary yet valuable step towards flow equivariance in vision transformers, demonstrating both the potential and an initial construction which may open the door for future improvements and encourage further investigation into efficient implementations.

AI USE STATEMENT

In this work, we used generative AI tools for the generation of the MNIST and Navier–Stokes datasets, to implement the models and experiments, and to plot and collate the results. We have not used generative AI tools for the fundamental development of the model architecture of mathematical claims, for writing proofs, or for proposing hypotheses. Additionally, we used generative AI tools for drafting portions of the text, transcribing the results, formatting references, and creating the figures. We have reviewed all AI-assisted work manually. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

In Appendices C, and D we include details on dataset construction and experiment protocol for both the MNIST and Navier–Stokes experiments. Code will be released upon acceptance.

REFERENCES

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. *Fourier Neural Operator for Parametric Partial Differential Equations*. International Conference on Learning Representations, 2021. https://arxiv.org/abs/2010.08895.

Ibrahim Khaliliya and T. Anderson Keller. *Flow Equivariant Transformers*. ICML Workshop on Weight-Space Symmetries, 2026. https://openreview.net/pdf?id=lXTPf5HDHy.

T. Anderson Keller. *Flow Equivariant Recurrent Neural Networks*. Advances in Neural Information Processing Systems, 2025. https://openreview.net/forum?id=N1KPOlcN6P.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. arXiv preprint arXiv:1607.06450, 2016. https://arxiv.org/abs/1607.06450.

Khang Ngo and Siamak Ravanbakhsh. *Scaling Laws and Symmetry, Evidence from Neural Force Fields*. International Conference on Learning Representations, 2026. https://openreview.net/forum?id=qyjaVda7t2.

Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. 2nd edition, Graduate Texts in Mathematics, Vol. 222. Springer, 2015.

Taco S. Cohen and Max Welling. *Group Equivariant Convolutional Networks*. arXiv preprint arXiv:1602.07576, 2016. https://arxiv.org/abs/1602.07576.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. Advances in Neural Information Processing Systems, Vol. 30, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. International Conference on Learning Representations, 2021. https://arxiv.org/abs/2010.11929.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv preprint arXiv:2104.09864, 2021. https://arxiv.org/abs/2104.09864.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. *Rotary Position Embedding for Vision Transformer*. European Conference on Computer Vision, 2024. https://arxiv.org/abs/2403.13298.

Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J. Smith, and Yaron Lipman. *Frame Averaging for Invariant and Equivariant Network Design*. International Conference on Learning Representations, 2022. https://arxiv.org/abs/2110.03336.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. arXiv preprint arXiv:2104.13478, 2021.

Risi Kondor and Shubhendu Trivedi. *On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups*. Proceedings of the 35th International Conference on Machine Learning, Vol. 80, pp. 2747–2755, 2018.

Taco S. Cohen, Mario Geiger, and Maurice Weiler. *A General Theory of Equivariant CNNs on Homogeneous Spaces*. Advances in Neural Information Processing Systems, Vol. 32, 2019.

Taco S. Cohen and Max Welling. *Steerable CNNs*. International Conference on Learning Representations, 2017. https://openreview.net/forum?id=rJQKYt5ll.

Maurice Weiler and Gabriele Cesa. *General E(2)-Equivariant Steerable CNNs*. Advances in Neural Information Processing Systems, Vol. 32, 2019.

Michael Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. *LieTransformer: Equivariant Self-Attention for Lie Groups*. Proceedings of the 38th International Conference on Machine Learning, Vol. 139, pp. 4533–4543, 2021.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. *SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks*. Advances in Neural Information Processing Systems, Vol. 33, pp. 1970–1981, 2020.

Yi-Lun Liao and Tess Smidt. *Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs*. International Conference on Learning Representations, 2023. https://openreview.net/forum?id=KwmPfARgOTD.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. *Self-Attention with Relative Position Representations*. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 464–468, 2018.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. *On Layer Normalization in the Transformer Architecture*. Proceedings of the 37th International Conference on Machine Learning, Vol. 119, pp. 10524–10533, 2020.

Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations, 2015.

Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. International Conference on Learning Representations, 2017. https://openreview.net/forum?id=Skq89Scxx.

Hansen Jin Lillemark, Benhao Huang, Fangneng Zhan, Yilun Du, and Thomas Anderson Keller. *Flow Equivariant World Models: Memory for Partially Observed Dynamic Environments*. arXiv preprint arXiv:2601.01075, 2026. https://arxiv.org/abs/2601.01075.

Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. International Conference on Learning Representations, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A Video Vision Transformer. *arXiv preprint arXiv:2103.15691*, 2021. https://arxiv.org/abs/2103.15691.

Renan A. Rojas-Gomez, Teck-Yian Lim, Minh N. Do, and Raymond A. Yeh. Making Vision Transformers Truly Shift-Equivariant. *arXiv preprint arXiv:2305.16316*, 2023. https://arxiv.org/abs/2305.16316.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021. https://arxiv.org/abs/2010.08895.

A Attention Implementation

This section gives the positional encoding and attention operations used in the patch backbone of Section 3.2.

For one spatial attention head, let $\mathbf{Q}_t(y), \mathbf{K}_t(y), \mathbf{V}_t(y) \in \mathbb{R}^d$ be the query, key and value vectors obtained by learned affine projections of the channel-normalized token $h_t^l(y)$. Here d is the even head width; we suppress layer and head indices in these vectors. Spatial rotary position embeddings (RoPE) (Su et al., 2021; Heo et al., 2024) rotate pairs of query and key channels, but leave the values unchanged. We use integer harmonics adapted to the periodic patch grid, rather than the usual geometric frequency schedule. For pair $j = 1, \dots, d/2$, define

$$\vartheta_j(y) := \frac{2\pi}{S/p}\langle m_j, y\rangle,$$
$$R_j(y) := \begin{pmatrix} \cos\vartheta_j(y) & -\sin\vartheta_j(y) \\ \sin\vartheta_j(y) & \cos\vartheta_j(y) \end{pmatrix}, \qquad \mathcal{R}(y) := \operatorname{diag}\big(R_1(y), \dots, R_{d/2}(y)\big). \tag{19}$$

Here $m_j \in \mathbb{Z}^2$ is a fixed integer harmonic and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. The implementation uses $m_1 = (1, 0)$, $m_2 = (0, 1)$, $m_3 = (2, 0)$, $m_4 = (0, 2)$, and so on. The angle $\vartheta_j(y)$ is distinct from the learned patch embedding ϕ; it specifies the jth rotation block. The rotated queries and keys, and their scaled attention scores, are

$$\widetilde{\mathbf{Q}}_t(y) := \mathcal{R}(y)\mathbf{Q}_t(y), \qquad \widetilde{\mathbf{K}}_t(y) := \mathcal{R}(y)\mathbf{K}_t(y),$$
$$e_t(y, y') := \frac{\widetilde{\mathbf{Q}}_t(y)^\top \widetilde{\mathbf{K}}_t(y')}{\sqrt{d}} = \frac{\mathbf{Q}_t(y)^\top \mathcal{R}(y' - y)\mathbf{K}_t(y')}{\sqrt{d}}. \tag{20}$$

The second equality follows from $\mathcal{R}(y)^\top\mathcal{R}(y') = \mathcal{R}(y' - y)$. Each head returns $\mathbf{O}_t(y) = \sum_{y'\in Y} \operatorname{softmax}_{y'}(e_t(y, y'))\mathbf{V}_t(y')$; The heads are concatenated and projected back to C channels before the residual addition. Integer harmonics make the rotations periodic on Y, so the positional factor depends only on relative patch positions, including across the periodic boundary. Temporal attention acts across observed times at each fixed patch position. Its projections are computed from $\mathrm{LN}_{l,2}[\bar{h}^l]$, after the spatial residual update, rather than from h^l. Its rotation matrix is denoted $\mathcal{R}_{\mathrm{time}}(t)$, distinct from the spatial $\mathcal{R}(y)$ above: its jth 2×2 block has angle $2\pi jt/(2T)$, for $j = 1, \dots, d/2$. Queries and keys are rotated at their respective observation times, and values are not rotated. The softmax is over the source observation time. Temporal attention is unmasked within the observed context window; it does not receive future target frames. No absolute spatial embeddings or velocity-specific parameters are used.

B Flow-Mixing Construction

The flow-independent construction in the main text is the variant used in our experiments. A more general construction allows hidden features to interact across velocity slices, as initially described in (Khaliliya and Keller, 2026). We first describe its convolutional formulation and then describe a patch-token counterpart.

Convolutional formulation. In this formulation, group convolutions are replaced by flow convolutions $\star_{\mathcal{V}\times G}$, which also convolve over the velocity dimension, following Keller (2025, Equation 9). Their kernels depend on relative velocity and relative spatial position. For the commuting translation flows considered here, the relative-flow index $\mu \ominus \nu$ reduces to $\mu - \nu$. Let h denote the lifted hidden state at one layer, with the layer and head indices suppressed. Flow-convolution projections of its channel-normalized features produce query, key and value maps. For a finite discrete spatial group G, write $\mathbf{Q}_t^\nu, \mathbf{K}_t^\nu \in \mathbb{R}^{d_{\mathrm{flat}}}$ for the queries and keys flattened over their channel and spatial coordinates, and $\mathbf{V}_t^\nu(g)$ for the value map at $g \in G$. Here d_{flat} is the flattened query dimension, not the per-patch head width d. Cross-flow attention takes the form

$$\gamma_{t,j,\mu}^\nu := \operatorname{softmax}_{j,\mu}\left(\frac{\langle \mathbf{Q}_t^\nu, \mathbf{K}_j^\mu\rangle}{\sqrt{d_{\mathrm{flat}}}} + b_{t-j,\mu-\nu}\right), \tag{21}$$

$$z_t(\nu, g) := \sum_{\mu\in\mathcal{V}_{\mathrm{model}}} \sum_{j\in I_T} \gamma_{t,j,\mu}^\nu \mathbf{V}_j^\mu(g), \qquad \mathrm{SA}_{\mathrm{mix}}[h]_t := z_t \star_{\mathcal{V}\times G} W_O. \tag{22}$$

The softmax is joint over source time j and source velocity μ for fixed t, ν. Here W_O is the learned output flow-convolution kernel. It acts on the whole function z_t of (ν, g), not on an already selected velocity slice. The scalar bias b depends only on relative time and velocity, analogous to relative-position self-attention (Shaw et al., 2018); it is not an absolute velocity embedding. We use γ for these attention weights to distinguish them from the forecasting gates α. With $b = 0$, masking to $\mu = \nu$ and restricting every flow-convolution kernel to zero velocity offset recovers the flow-independent convolutional construction of Khaliliya and Keller (2026). The equivariance argument uses the same admissible reindexings as Section 3.6; kernels and relative biases must respect any periodic identification of indices.

Patch-token counterpart. For patch tokens, temporal attention can instead be extended from source times j to pairs (j, μ) at a fixed patch position y. The attention weights become $\gamma^{\nu}_{t,j,\mu}(y)$: Equation 21 uses the temporally rotated query and key at that patch, with head dimension d rather than d_{flat}. The temporal RoPE from Section 3.2 is shared across velocity slices. The weighted sum is now $z_t(\nu, y)$, and concatenated heads are followed by a shared affine projection, not the flow convolution in Equation 22. Spatial attention and token-wise MLPs remain as in Equation 7. Masking to $\mu = \nu$ and removing the additional bias recovers the temporal attention of our evaluated model. Thus, cross-flow attention does not require adding flow-convolution layers to the patch backbone. Once slices interact, residuals and logits must be functions of the entire lifted bank, not independently of q^{ν}. Phase averaging would apply each spatial offset jointly to all slices before decoding and aligning the outputs; the time-T unlift and pixel-wise forecasting gates are unchanged. We leave evaluation of flow mixing to future work.

C Periodic Unforced Navier–Stokes Dataset

Let $r \in [0, \ell_{\mathrm{box}})^2$ denote physical position with periodic boundaries, and τ physical time. The fluid velocity $u(r, \tau) \in \mathbb{R}^2$ and pressure $P(r, \tau) \in \mathbb{R}$ satisfy

$$\partial_\tau u + (u \cdot \nabla)u = -\nabla P + \kappa \Delta u, \qquad \nabla \cdot u = 0, \qquad \kappa = 10^{-3}. \tag{23}$$

Here κ is the viscosity; all spatial derivatives are with respect to r. The scalar vorticity is $\omega := \partial_{r_1} u_2 - \partial_{r_2} u_1$. We solve its evolution in float64 using a Fourier pseudo-spectral method, $2/3$ dealiasing and fourth-order Runge–Kutta (RK4), with 16 substeps per observation interval Δt:

$$\partial_\tau \omega + u \cdot \nabla \omega = \kappa \Delta \omega. \tag{24}$$

Fourier coefficients and initial conditions. For the 2π box, let $k = (k_1, k_2) \in \{-S/2, \ldots, S/2 - 1\}^2$ index the Fourier modes. Write $U_k(\tau)$ and $\Omega_k(\tau)$ for the discrete Fourier coefficients of the sampled velocity and vorticity fields, respectively. Uppercase letters distinguish these coefficients from the spatial fields u and ω. For the zero-mean base simulation, velocity is recovered by

$$U_k(\tau) = \frac{i(k_2, -k_1)}{\|k\|_2^2}\, \Omega_k(\tau) \quad (k \neq 0), \qquad U_0(\tau) = 0, \tag{25}$$

where $i^2 = -1$ and $\|\cdot\|_2$ is the Euclidean norm. Draw independent $\epsilon(x) \sim \mathcal{N}(0, 1)$ at grid points, and let E_k denote its discrete Fourier coefficients. Initialize

$$\Omega_k(0) := A\, M(k)(1 + \|k\|_2^2)^{-5/4} E_k, \qquad M(k) := \begin{cases} 1, & k \neq 0,\ |k_1| < S/3,\ |k_2| < S/3, \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

The positive scale A sets the initial root-mean-square (RMS) speed to 0.5: if $u^{(0)}$ denotes the resulting zero-drift solution and $f_t^{(0)}(x) := u^{(0)}(\Delta x\, x, t\Delta t)$ its sampled frames, then

$$\left(\frac{1}{S^2} \sum_{x \in X} \|f_0^{(0)}(x)\|_2^2 \right)^{1/2} = 0.5. \tag{27}$$

The superscript (0) labels the zero-drift trajectory.

Adding drift and splitting the data. After simulating the base trajectory, we add a prescribed constant drift:

$$u^{(\nu)}(r,\tau) := u^{(0)}(r - c_\nu\tau, \tau) + c_\nu,$$
$$f_t^{(\nu)}(x) := u^{(\nu)}(\Delta x\, x, t\Delta t) = f_t^{(0)}(x - t\nu) + c_\nu. \tag{28}$$

The last equality uses integer pixel velocities and $c_\nu \Delta t = \Delta x \nu$. We use 1,024 training and 192 independent validation trajectories in $\mathbb{R}^{36\times 2\times 32\times 32}$, with 64 for checkpoint selection and 128 for diagnostics; final-test data remain unused. We take $\mathcal{V}_{\text{train}} = \{-1, 0, 1\}^2$ and $\mathcal{V}_{\text{eval}} = \{-3, \dots, 3\}^2$. Diagnostics reuse each trajectory across generators, separating seen flows, unseen modeled flows and flows outside $\mathcal{V}_{\text{model}}$.

Forecasting task. Let $f \in \mathcal{F}_K(X, I_{T+H})$ contain the first $T + H$ frames of a trajectory. Given its first $T = 6$ frames, set $\widehat{f}_t := f_t$ for $0 \le t < T$ and recursively predict

$$\widehat{f}_{T+j} := \Phi_\theta[(\widehat{f}_j, \dots, \widehat{f}_{j+T-1})], \qquad j = 0, \dots, H-1. \tag{29}$$

Each input window is reindexed locally from 0 to $T - 1$; H is the forecast horizon. Here Φ_θ denotes the chosen raw or mean-frame predictor. Let $\mathcal{D}_{\text{train}}$ be the training distribution over trajectories and their sampled training drifts. The mean squared error is

$$\mathcal{L}_H(\theta) := \mathbb{E}_{f\sim\mathcal{D}_{\text{train}}}\left[\frac{1}{KHS^2}\sum_{j=0}^{H-1}\sum_{x\in X}\|\widehat{f}_{T+j}(x) - f_{T+j}(x)\|_2^2\right], \qquad K = 2. \tag{30}$$

The original training horizon is $H_{\text{train}} = 4$; diagnostic rollouts use $H_{\text{eval}} = 30$. Dataset checks cover finite values, integration stability, near-zero divergence, and energy decay. These checks are not an additional model loss or a guarantee about predicted trajectories.

D Flow-MNIST Data and Experimental Protocol

Flow-MNIST generation. Each sequence contains two independently sampled MNIST digits. Each 28×28 grayscale digit is embedded in a zero-padded 32×32 canvas, given a uniformly sampled integer offset, and translated periodically at a constant integer velocity. The images are added and clipped to $[0, 1]$. Training and mixed-ID validation sample each digit's velocity independently from $\{-1, 0, 1\}^2$. At each fixed-velocity diagnostic condition, both digits instead receive the same prescribed velocity, keeping their appearance and initial positions paired across conditions. The fixed-grid results therefore probe global-flow generalization, not arbitrary independent object motion.

Training digits come from the standard MNIST training-image pool. The original MNIST test-image pool is partitioned by image index: its first 5,000 images supply validation digits, and the remaining images are reserved for final testing and are not accessed by the reported benchmark. Each sequence uses a deterministic, index- and split-specific random stream; training streams also depend on the data pass. There are 12,000 training sequences per pass and 192 validation sequences, each with 40 frames. Validation sequence indices $[0, 64)$ are used only for checkpoint selection and $[64, 192)$ for diagnostics. Disjoint rendered sequences need not contain disjoint digit identities within the validation image pool.

In addition to mixed ID and the 49 fixed global velocities, MNIST includes four independent-digit conditions with paired velocities $(2, 0)/(0, 2)$, $(2, 1)/(-2, -1)$, $(3, 0)/(0, -3)$ and $(3, 2)/(-3, -2)$. These bring the total to 54 conditions; NS has 50 conditions (mixed ID and the 49 global boosts). Every condition uses 128 diagnostic trajectories. Reusing a trajectory under multiple motions does not create independent trajectory samples.

Training and selection. All original runs start from fresh initialization with seeds 42–46. Initialization seeds and data order are paired within each task. We use AdamW (Loshchilov and Hutter, 2019), batch size four, weight decay 10^{-4} and gradient-norm clipping at one. The learning rate warms up for 5% of updates to 10^{-3}, then follows a cosine schedule to 10^{-4}. MNIST uses 10,000 updates and ten-step training rollouts; NS uses 10,240 updates (40 passes over the base trajectories)

and four-step training rollouts. Both minimize the channel-, pixel-, and time-averaged autoregressive MSE of Equation 30, with task-specific K, T and H. The training loss is differentiated through the rollout; evaluation feeds predictions back without teacher forcing. The context window is shifted and reindexed at every predicted step.

Selection evaluates 64 mixed-ID examples at initialization and every 500 MNIST or 512 NS updates. The checkpoint with lowest training-horizon selection MSE is retained; no 30-step diagnostic is used for original checkpoint selection. Equal-update budgets and shared backbone dimensions do not imply equal parameters, FLOPs, or wall-clock cost. In particular, FET evaluates multiple lifted frames and patch phases. We do not infer convergence from reaching the allotted budget.

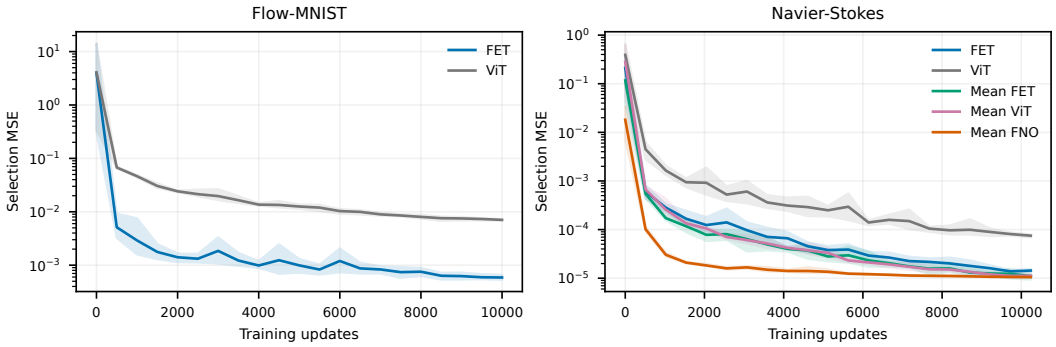


Figure 4: **Original-benchmark selection curves.** Mean and full range over five seeds. The objective averages ten predicted frames for MNIST and four for NS. The horizontal axis counts updates, not equal compute or elapsed time. Every original seed is retained.

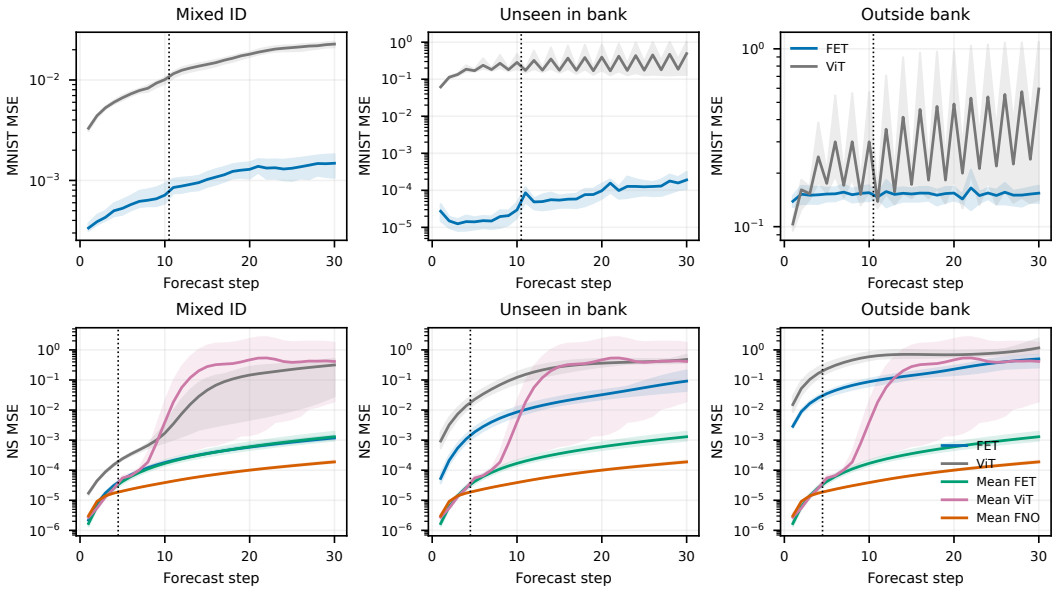


Figure 5: **Complete 30-step rollout curves by motion regime.** Top: Flow-MNIST. Bottom: Navier–Stokes. Curves average conditions within each regime and then the five seeds; shading is the full seed range. Vertical dotted lines separate the trained forecast horizon from longer rollouts. The plotted quantity is error at each step, not cumulative mean error.

E Additional Results and Ablations

All studies below use validation data. Their different training budgets, selection policies, and sample counts are kept separate from the original benchmark.

E.1 Additional Navier–Stokes Training

All 25 NS model/seed runs continue from update 10,240 to 20,480 with full optimizer, random-number-generator, and data-position restoration. A new cosine tail decreases the learning rate from 10^{-4} to 10^{-5}. The original best checkpoint remains a selection candidate, and selection still uses the original 64 examples and four-step objective. Every selected checkpoint in this study comes from the continuation. Diagnostics retain the original 128 examples, 50 conditions, and 30 steps.

Table 2: NS continuation: five-seed mean of 30-step-average MSE in units of 10^{-3}. Each cell compares the original checkpoint with the continued selection; These are not fresh longer-schedule runs.

Model	Mixed ID	Unseen in bank	Outside bank
FET	$0.452 \to 0.221$	$28.1 \to 34.2$	$194 \to 197$
ViT	$99.9 \to 314$	$243 \to 336$	$611 \to 723$
Mean FET	$0.470 \to 0.231$	$0.470 \to 0.231$	$0.470 \to 0.231$
Mean ViT	$234 \to 189$	$234 \to 189$	$234 \to 189$
Mean FNO	$0.0795 \to 0.0697$	$0.0795 \to 0.0697$	$0.0795 \to 0.0697$

Mean FET's mixed-ID error improves by 50.8%, with continued sample SD 4.58×10^{-5}, while mean FNO improves by 12.4%, with sample SD 1.04×10^{-6}. All five seeds improve for both models; mean FNO still has $3.32\times$ lower MSE. Raw FET improves on mixed ID but degrades on unseen motion. ViT's one-step mixed-ID error improves by 41.5%, yet its 30-step-average error increases by 214%, with deterioration in every seed. Short-horizon selection is therefore not sufficient to establish rollout stability. Mean ViT's continued mixed-ID errors range from 0.000565 for seed 44 to 0.920 for seed 42; no unstable seed is removed.

Physical diagnostics. We additionally record velocity-relative L^2 error, mean-velocity and fluctuation MSE, divergence RMS, and relative kinetic-energy and enstrophy errors, all computed from the predicted velocity fields. Neither model enforces incompressibility. Mean FET's mixed-ID divergence RMS falls from 0.0151 to 0.0103, and its relative energy error from 0.0174 to 0.0101. For mean FNO, divergence RMS instead rises from 0.00831 to 0.00942 despite lower prediction MSE. Improved forecast MSE does not imply improvement in every physical diagnostic.

E.2 Training and Selection Horizon Pilot

For seeds 42 and 43, mean FET, mean ViT, mean FNO, and ViT each branch from update 20,480 into 1,024-update continuations with training horizons four and ten. The branches restore the same parent state and use the same explicit cosine restart from 10^{-4} to 10^{-5}. Four- and 30-step checkpoint selection use the same candidate checkpoints on indices $[0, 64)$. Diagnostics use only 32 cases $[64, 96)$ and ten predeclared NS motion conditions. This is a two-seed, equal-update pilot, not a five-seed confirmation.

Holding 30-step selection fixed, changing the training horizon from four to ten reduces mean FET mixed-ID MSE from 2.47×10^{-4} to 2.42×10^{-4} (2.22%). Only seed 42 improves; seed 43 selects the unchanged parent in both branches. Mean FNO remains $3.28\times$ better. Holding ten-step training fixed, selecting on 30 rather than four steps reduces ViT and mean ViT MSE by 62.5% and 95.0%, respectively, but mean ViT remains seed-sensitive. Mean-FET ten-step updates take about $2.52\times$ as long as four-step updates on the same H100; this excludes selection, checkpointing and other non-update costs.

E.3 Matched Early-Budget Non-Patch Comparison

Six fresh MNIST runs compare the earlier non-patch FET, patch FET and ViT at 1,024 updates, seeds 42/43, with identical data, context, rollout and batch protocols. Diagnostics use 32 validation sequences, 14 conditions and 30 steps. Mixed-ID MSE (mean $\pm$ sample SD) is 0.1043 ± 0.00638 for non-patch FET, 0.00324 ± 0.000126 for patch FET, and 0.0797 ± 0.00949 for ViT. Patch FET improves over both baselines in both seeds, but these early-budget results do not establish convergence or equal-compute superiority. The historical two-seed non-patch reports use different splits, training budgets and selection protocols, and are not pooled with the main five-seed benchmark.

E.4 Fixed-Checkpoint Velocity-Bank Expansion

We compare learned-gate radius-two and radius-three inference banks for five fixed raw-FET checkpoints per dataset: the original MNIST selections and the continued NS selections. There is no retraining or reselection. Forty disjoint condition shards cover the ten checkpoint comparisons, with all 128 validation cases $[64, 192)$, 54 MNIST/50 NS conditions and 30 steps. All reports are complete, with no technical failures or nonfinite forecasts. Compatible earlier diagnostic rows are reused, not counted as additional independent samples.

Table 3: **Bank coverage improves unseen-motion error with ID tradeoffs.** Five-seed means of 30-step-average MSE at fixed weights. Motion-regime labels always refer to the original radius-two bank.

Dataset	Original regime	Radius 2	Radius 3	Change
MNIST	Mixed ID	0.000996	0.001067	+7.11%
MNIST	Unseen in bank	0.0000753	0.0000980	+30.10%
MNIST	Outside bank	0.1524	0.000104	−99.93%
NS	Mixed ID	0.000221	0.000235	+5.97%
NS	Unseen in bank	0.03424	0.000330	−99.04%
NS	Outside bank	0.1968	0.03124	−84.13%

Originally outside-bank error improves in every seed. The NS radius-three outside-bank mean has sample SD 0.0227, so appreciable variability and residual error remain. Mixed-ID error worsens in all five NS seeds and four of five MNIST seeds. NS mixed-ID relative energy and enstrophy errors worsen 32.9% and 50.7%, respectively, despite a 2.72% reduction in divergence RMS. The independent-digit outside-bank MNIST cases improve by 99.56%, but this does not establish equivariance to arbitrary local motion.

The velocities originally outside radius two are inside radius three. These results therefore concern expanded coverage, not extrapolation beyond the expanded bank. Coverage and changed competition among learned gates are not causally isolated. In a separate two-seed subset diagnostic, uniform instead of learned gating severely worsens radius-two mixed-ID MSE to 0.155 (MNIST) and 0.268 (NS). The five-seed bank study keeps learned gating throughout. No final-test evaluation or new independent holdout is introduced by any of these analyses.