FLOWGEN: A FINITE-HORIZON TRUNCATION KERNEL FOR UNSTABLE 3D FLOW FORECASTING ACROSS MESHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Fast surrogate models for unsteady computational fluid dynamics (CFD) need to maintain representation accuracy under complex spatial discretizations and evolution stability in long-term prediction. However, existing methods still have limited joint modeling of the temporal and spatial dimensions. As a result, they cannot maintain accuracy in both dimensions. We propose FlowGEN, a graph evolution network centered on a Finite-Horizon Truncation Kernel (FHTK). FHTK is a separable temporal support kernel over history lag and forecast offset. Along the history axis, a transient-state mask activates only the latest three states, so an older prediction loses its direct feedback edge after leaving the finite-memory window while its dynamically relevant information remains summarized by the evolved recent states. Along the forecast axis, a normalized finite-support kernel determines which recursively generated states contribute loss and gradient. Its support expands from three to six steps during training, and a nested six-/twelve-step validation kernel balances local transition accuracy with delayed instability beyond the optimized horizon. Every supported state is supervised by standardized data fidelity, kinetic energy (KE) matching, and mixed graph divergence (Div) regularization. Across structured and unstructured unstable flows at four Reynolds numbers, FlowGEN obtains the lowest reported mean in all 48 condition–metric comparisons. Relative to the strongest baseline for each metric, it reduces the error-growth ratio by 32.1%, intermediate-horizon MSE by 8.6%, KE error by 31.5%, and Div error by 3.7%. These results show that explicitly controlling state activation and recursive credit provides a practical mechanism for stable flow forecasting across discretizations.

1 INTRODUCTION

High-fidelity CFD is essential for engineering design, control, and scientific discovery. However, repeatedly solving the governing equations is expensive for rapid parameter sweeps and real-time prediction. Learning-based surrogates amortize this cost by approximating a temporal evolution operator from simulation trajectories. Their practical value therefore depends on more than one-step interpolation accuracy. A useful surrogate must preserve the native spatial discretization, maintain important physical statistics, and control error growth when its predictions are fed back recursively.

Spatial and temporal fidelity are coupled. Mapping an irregular CFD field onto a Cartesian grid enables efficient grid-based processing, but interpolation or geometric transformation may alter local neighborhoods, blur boundaries, and discard the nonuniform resolution used around high-gradient regions (Gao et al., 2021; Kashefi et al., 2021; Li et al., 2022). Temporally, one-step training conditions on CFD histories, while recursive inference conditions on imperfect model states. This sequential distribution shift is known to compound prediction errors in imitation and autoregressive sequence learning (Ross et al., 2011; Bengio et al., 2015; Lamb et al., 2016), and it also degrades learned PDE rollouts (Brandstetter et al., 2022; Lippe et al., 2023). Spatial information lost before prediction therefore becomes an initial temporal perturbation that can be repeatedly amplified. Existing methods usually only perform well in one aspect. Grid-based neural operators efficiently learn evolution on regular domains, while geometry-native point and graph networks preserve irregular samples (Li et al., 2021; Kashefi et al., 2021; Kashefi & Mukerji, 2022; Sanchez-Gonzalez

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.

et al., 2020; Pfaff et al., 2021). Long-horizon stabilization is commonly introduced through temporal bundling, noisy inputs, pushforward training, or iterative refinement (Pfaff et al., 2021; Brandstetter et al., 2022; Lippe et al., 2023). However, "rollout length" conflates two different mechanisms. The *feedback memory* determines which generated states remain direct inputs to the next transition, whereas the *credit horizon* determines how many self-generated transitions are connected to the training loss. Leaving these two supports implicit obscures where recursive error can re-enter the model and how deeply products of transition Jacobians affect optimization.

To address this mismatch, we propose FlowGEN, a Flow Graph Evolution Network whose central temporal mechanism is a Finite-Horizon Truncation Kernel (FHTK). FHTK is defined over two temporal axes. A binary lag kernel activates only the latest three states in every recursive update. Thus, a generated frame is not arbitrarily deleted. It contributes while it belongs to the active window and loses its direct feedback edge only after it becomes stale. A normalized forecast-offset kernel independently limits the self-generated states that contribute supervision and gradient. Its support expands from three to six steps as the transition operator becomes more reliable, converting a procedural rollout schedule into an explicit state-and-credit truncation rule. During checkpoint selection, a nested six-/twelve-step kernel preserves local accuracy while rejecting parameters that develop delayed drift beyond the training boundary. Recursive inference itself remains uninterrupted and can extend beyond both supports.

FlowGEN couples this temporal kernel with a geometry-native predictor. It constructs a k-nearest-neighbor graph directly from sample coordinates, augments each point with multiscale Fourier geometry features, and processes local interactions using EdgeConv and residual MLP blocks. The model predicts standardized velocity increments rather than reconstructing the next state from scratch. At every FHTK-supported step, data fidelity is combined with KE matching and a mixed Div penalty. The latter regularizes both the predicted field and the prediction error, retaining an incompressibility prior without forcing the network to remove discretization and interpolation residuals already present in the aligned CFD targets. Experiments on structured and unstructured incompressible flows show that FlowGEN consistently controls field error, physical drift, and recursive error amplification. The work makes three main contributions.

- We introduce the Finite-Horizon Truncation Kernel, a separable support operator over history lag and forecast offset. Its three-state activation mask bounds direct feedback memory, its progressive finite-support objective controls recurrent credit assignment, and its nested six-/twelve-step selection kernel detects delayed instability beyond the optimized horizon.
- We develop a unified graph evolution backbone for structured and unstructured point samples. Coordinate-constructed neighborhoods, multiscale geometric encoding, and residual velocity evolution model local flow dynamics without requiring the connectivity of the original numerical mesh.
- We integrate FHTK with KE matching and mixed Div regularization. Applying these constraints to each kernel-supported generated state improves physical consistency and reduces error amplification across Reynolds numbers and mesh representations.

2 Related Work

Grid-based operators and geometry adaptation. Neural operators learn maps between function spaces and can share parameters across discretizations (Kovachki et al., 2023). FNO parameterizes global convolution in the spectral domain (Li et al., 2021), DeepONet separates input-function and query encoders (Lu et al., 2021), and the Galerkin Transformer interprets attention as an operator approximation (Cao, 2021). PDEBench further shows that time-dependent PDE surrogates require evaluation across equations, parameters, and physical metrics (Takamoto et al., 2022). Extending regular-grid backbones to irregular domains still requires a geometric interface. PhyGeoNet maps an irregular physical domain to a regular reference domain (Gao et al., 2021), while Geo-FNO learns a deformation into a regular latent domain (Li et al., 2022). GINO combines graph and Fourier operators for large-scale 3D PDEs (Li et al., 2023), and Transolver organizes arbitrary points into learned physical states (Wu et al., 2024). These methods broaden geometric coverage, but spatial adaptation and temporal rollout stability remain distinct design concerns. FlowGEN uses one

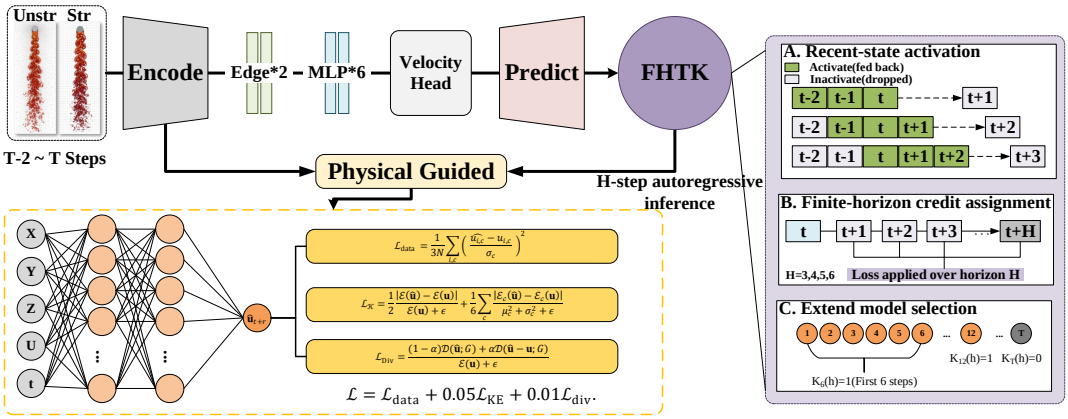


Figure 1: FlowGEN framework with the FHTK. A shared graph-based predictor maps structured or unstructured flow histories to the next velocity field, while data, KE, and Div losses constrain every recursive step. FHTK limits direct feedback to the latest $K = 3$ states, progressively expands the supervised horizon from $H = 3$ to $H = 6$, and uses nested 6-step and 12-step validation to select checkpoints that remain stable beyond the optimized horizon. The resulting model performs long-horizon autoregressive inference without trajectory resets.

coordinate-graph interface for both supplied mesh types and optimizes that representation together with autoregressive deployment.

Geometry-native point-cloud and graph models. Point-cloud models operate directly on irregular samples and avoid the interpolation required by Cartesian projection. Kashefi et al. preserve CFD vertices and boundary geometry with a PointNet-based predictor (Kashefi et al., 2021), and Physics-informed PointNet extends this idea to steady incompressible flows over multiple irregular geometries (Kashefi & Mukerji, 2022). These works establish the spatial value of point-native representations but do not study long-horizon unsteady rollout. For dynamic systems, Graph Network-based Simulators represent particles as graph nodes and learn dynamics by message passing (Sanchez-Gonzalez et al., 2020). MeshGraphNets extend this paradigm to mesh-based physical systems (Pfaff et al., 2021), and message-passing neural PDE solvers explicitly address autoregressive stability through temporal bundling and pushforward training (Brandstetter et al., 2022). FlowGEN does not require the supplied mesh edges used by many graph simulators. It reconstructs every graph from sampled coordinates, making the interface consistent across structured and unstructured meshes. Its local operator follows the EdgeConv principle of learning pairwise neighborhood features (Wang et al., 2019); GCN (Kipf & Welling, 2017), GAT (Veličković et al., 2018), FNO, and MP-PDE (Brandstetter et al., 2022) provide complementary graph, operator, and autoregressive PDE-solver baselines.

Physics guidance and long-horizon evolution. Physics-informed learning incorporates governing structure into optimization (Raissi et al., 2019). In fluid simulation, prior work has learned pressure projection for incompressibility (Tompson et al., 2017), trained physics-constrained autoregressive PDE models (Geneva & Zabaras, 2020), and coupled networks with differentiable numerical solvers (Um et al., 2020; Kochkov et al., 2021). Other approaches learn incompressible updates without simulation labels (Wandel et al., 2021), impose discrete PDE residuals in recurrent architectures (Ren et al., 2022), or use noise and temporal coarsening to stabilize turbulent simulators (Stachenfeld et al., 2022). These methods demonstrate that physical guidance and deployment-aware training matter, but they do not explicitly separate the states retained for direct feedback from the transitions retained for gradient credit.

Outside PDE learning, DAgger aggregates states visited by the learned policy (Ross et al., 2011), scheduled sampling mixes observed and generated inputs (Bengio et al., 2015), and Professor Forcing aligns teacher-forced and free-running dynamics (Lamb et al., 2016). Message-passing neural PDE solvers instead use temporal bundling and pushforward training (Brandstetter et al., 2022),

while PDE-Refiner targets neglected low-amplitude spatial frequencies through iterative refinement (Lippe et al., 2023). Finite-horizon optimization is also related to truncated backpropagation through time, which limits backward dependencies to control computation and gradient bias. FlowGEN differs by representing history lag and forecast offset as the two supports of one state-and-credit truncation kernel. This makes direct feedback memory, training credit depth, and validation extension explicit while coupling every supported forecast state with KE and Div supervision.

3 METHODOLOGY

As shown in Fig. 1, FlowGEN learns a one-step velocity evolution operator on a shared point-based representation and recursively applies it to predict long three-dimensional flow trajectories. The graph backbone provides geometry-aware local evolution, the physics objective constrains each generated state, and the Finite-Horizon Truncation Kernel explicitly determines which recent states remain active and how far recursive errors influence training and model selection across structured and unstructured samples.

3.1 DATA PREPROCESSING

Let $\mathcal{X}_t = \{\mathbf{x}_{t,i}\}_{i=1}^{N_t} \subset \mathbb{R}^3$ and $\mathbf{u}_t \in \mathbb{R}^{N_t \times 3}$ denote the sampled coordinates and velocity field. Given three history frames, FlowGEN learns

$$\widehat{\mathbf{u}}_{t+1} = \mathcal{F}_\theta(\mathcal{X}_t, \mathbf{u}_{t-2:t}, Re, m), \tag{1}$$

where Re specifies the flow regime and $m \in \{0, 1\}$ identifies the mesh type. For adaptive unstructured sequences, the latest coordinates $\mathcal{X}_t$ define the reference set. Earlier fields and future targets are aligned to this set by eight-neighbor inverse-distance weighting, while structured sequences retain direct point correspondence. The weighted mapping distances $\mathbf{d}_{t-2:t,i}$ encode alignment uncertainty, and the mean distance ρ_i to the 12 nearest neighbors describes local sampling density. Each window uses a coordinate-based kNN graph with $k = 12$.

Coordinates are normalized over the physical domain, and each velocity component is standardized with training-set statistics (μ_c, σ_c). The point feature is

$$\mathbf{a}_i = [\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{u}}_{t-2:t,i}, \mathbf{d}_{t-2:t,i}, \rho_i, Re/6400, m]. \tag{2}$$

This representation preserves the native point geometry while providing a shared interface for both mesh types.

3.2 ARCHITECTURE

FlowGEN combines local graph interaction with a deep pointwise processor to model geometric relations and nonlinear state evolution efficiently.

Fourier Features. Normalized coordinates are augmented with Fourier features using six frequencies

$$\gamma(\widetilde{\mathbf{x}}_i) = [\sin(\pi f \widetilde{\mathbf{x}}_i), \cos(\pi f \widetilde{\mathbf{x}}_i)]_{f=1}^{6}. \tag{3}$$

The concatenated feature $[\mathbf{a}_i, \gamma(\widetilde{\mathbf{x}}_i)]$ is projected by a linear layer, LayerNorm, and GELU into a width-192 embedding $\mathbf{h}_i^0$. The raw coordinates preserve absolute position, while the Fourier channels encode multiple spatial scales that are difficult for a plain MLP to represent efficiently. LayerNorm balances the expanded geometric encoding and the standardized history features.

EdgeConv. Spatial interaction is performed on the static coordinate graph. For an embedding $\mathbf{h}_i$, mean-aggregated EdgeConv computes

$$\mathbf{l}_i = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \phi([\mathbf{h}_i, \mathbf{h}_j - \mathbf{h}_i]), \tag{4}$$

where ϕ is a two-layer MLP with LayerNorm and GELU. The center feature retains the local state, while $\mathbf{h}_j - \mathbf{h}_i$ represents spatial variation associated with transport and shear. Mean aggregation

avoids direct dependence on node degree, and the explicit density feature ρ_i reduces sensitivity to irregular neighbor spacing. FlowGEN uses two EdgeConv stages. The first extracts neighborhood features and concatenates them with the pointwise embedding before projection back to width 192. This fusion keeps temporal and neighborhood evidence separately accessible. The second stage refines the fused state through a residual connection. The coordinate graph remains fixed during rollout because physical proximity provides a stable inductive bias and avoids neighborhood changes caused by prediction errors.

Residual MLP. Six residual MLP blocks process the graph features

$$\mathbf{h}_i^{\ell+1} = \mathbf{h}_i^{\ell} + \psi_\ell(\mathrm{LN}(\mathbf{h}_i^{\ell})), \qquad \ell = 1, \dots, 6, \tag{5}$$

where each ψ_ℓ expands the hidden dimension by a factor of two, applies GELU and dropout, and projects it back to width 192. These blocks combine geometry, history, flow regime, and mesh information while preserving the graph features through residual paths.

Velocity Head. A two-layer head predicts a standardized velocity increment

$$\widehat{\mathbf{u}}_{t+1,i} = \mathbf{u}_{t,i} + \boldsymbol{\sigma}_u \odot g_\theta(\mathbf{h}_i^6). \tag{6}$$

This residual form exploits the temporal persistence of unsteady flow and converts prediction into velocity-increment estimation. Multiplication by the component-wise standard deviation restores the physical scale before autoregressive feedback. The explicit identity path allows small corrections to accumulate more smoothly than repeated full-state reconstruction.

3.3 Physics-Constrained Learning

Pointwise supervision alone cannot prevent small KE and Div biases from being repeatedly amplified during autoregressive prediction. FlowGEN therefore uses a data term together with KE matching and graph-based Div regularization. The component-standardized data loss is

$$\mathcal{L}_{\mathrm{data}} = \frac{1}{3N} \sum_{i,c} \left(\frac{\widehat{u}_{i,c} - u_{i,c}}{\sigma_c} \right)^2, \tag{7}$$

which prevents the dominant streamwise component from suppressing transverse errors.

Define $\mathcal{E}(\mathbf{u}) = N^{-1} \sum_i \|\mathbf{u}_i\|_2^2$ and $\mathcal{E}_c(\mathbf{u}) = N^{-1} \sum_i u_{i,c}^2$. The KE loss matches both total magnitude and component-wise second moments

$$\mathcal{L}_{\mathrm{KE}} = \frac{1}{2} \frac{|\mathcal{E}(\widehat{\mathbf{u}}) - \mathcal{E}(\mathbf{u})|}{\mathcal{E}(\mathbf{u}) + \epsilon} + \frac{1}{6} \sum_c \frac{|\mathcal{E}_c(\widehat{\mathbf{u}}) - \mathcal{E}_c(\mathbf{u})|}{\mu_c^2 + \sigma_c^2 + \epsilon}. \tag{8}$$

Because a shared finite-volume stencil is unavailable across mesh types, Div is measured on the coordinate graph. For a vector field $\mathbf{v}$, we use

$$\begin{aligned} \delta_{ij}(\mathbf{v}) &= \frac{(\mathbf{v}_j - \mathbf{v}_i)^\top (\mathbf{x}_j - \mathbf{x}_i)}{\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 + \epsilon}, \\ \mathcal{D}(\mathbf{v}; G) &= \frac{1}{N} \sum_i \left[\frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \delta_{ij}(\mathbf{v}) \right]^2. \end{aligned} \tag{9}$$

This permutation-invariant proxy measures coherent local expansion or contraction without assuming a common numerical stencil. To retain the incompressibility prior while respecting residuals introduced by adaptive sampling and IDW alignment, FlowGEN combines absolute-field and error-field terms

$$\mathcal{L}_{\mathrm{Div}} = \frac{(1-\alpha)\mathcal{D}(\widehat{\mathbf{u}}; G) + \alpha \mathcal{D}(\widehat{\mathbf{u}} - \mathbf{u}; G)}{\mathcal{E}(\mathbf{u}) + \epsilon}, \qquad \alpha = 0.75. \tag{10}$$

The step-wise objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{data}} + 0.05 \mathcal{L}_{\mathrm{KE}} + 0.01 \mathcal{L}_{\mathrm{Div}}. \tag{11}$$

The Div weight is linearly warmed up during the early training. All three terms are evaluated at every recursive step supported by FHTK.

3.4 FINITE-HORIZON TRUNCATION KERNEL

Autoregressive forecasting contains two temporal supports. The history support controls which states directly enter each transition, while the forecast support controls how far losses and gradients propagate through generated states. FHTK couples both supports in one operator.

Let $K = 3$ and let $\widetilde{\mathbf{u}}_s$ denote the ground-truth state for $s \le t$ and the generated state for $s > t$. At offset r, FlowGEN applies

$$\begin{aligned} \mathcal{H}^{(K)}_{t+r-1} &= [\widetilde{\mathbf{u}}_{t+r-K}, \dots, \widetilde{\mathbf{u}}_{t+r-1}], \\ \widehat{\mathbf{u}}_{t+r} &= \mathcal{F}_\theta(\mathcal{X}_t, \mathcal{H}^{(K)}_{t+r-1}, Re, m). \end{aligned} \tag{12}$$

The newest mapping-distance channel is set to zero, the two older channels move with their corresponding fields, and the coordinate graph remains fixed. The lag mask $a_K(j) = \mathbb{I}[0 \le j < K]$ gives only the latest K states a direct edge to the next prediction. Older states are not skipped when they are generated. They leave the active window after K transitions, while their dynamical influence remains encoded in the evolved recent states.

For the step loss $\ell_r(\theta) = \mathcal{L}(\widehat{\mathbf{u}}_{t+r}, \mathbf{u}_{t+r})$, define the normalized offset kernel and the joint truncation kernel as

$$\kappa_H(r) = \frac{1}{H}\mathbb{I}[1 \le r \le H], \qquad \mathcal{K}_{K,H}(j, r) = a_K(j)\kappa_H(r). \tag{13}$$

The resulting training objective is

$$\mathcal{L}_{\mathrm{FHTK}}(\theta; K, H) = \sum_{r=1}^{\infty} \kappa_H(r)\ell_r(\theta), \qquad K = 3. \tag{14}$$

The normalization makes objectives with different H comparable. Truncation limits direct state feedback and recursive credit assignment, but it does not reset the generated trajectory or restrict the inference horizon.

The finite lag has a direct error-propagation interpretation. Under a local first-order expansion,

$$\mathbf{e}_{t+r} = \sum_{j=1}^{K} \mathbf{J}_{r,j}\mathbf{e}_{t+r-j} + \boldsymbol{\xi}_r + \mathcal{O}(\|\mathbf{E}_{r-1}\|^2), \tag{15}$$

where $\boldsymbol{\xi}_r$ collects one-step and alignment errors. The mask therefore fixes the number of direct error-injection paths to K, even as the trajectory grows. We choose $K = 3$ by repeated validation over candidate history lengths. Shorter histories insufficiently represent local temporal evolution, whereas longer histories add stale predicted inputs without consistent gains. Details are provided in Appendix.

The forecast support expands progressively

$$H(e) = 3 + \min\left(3, \left\lfloor \frac{e-1}{4} \right\rfloor\right), \tag{16}$$

which increases the optimized horizon from three to six steps in four-epoch stages. Every state inside the support is model-generated after the initial history, so the procedure directly exposes the operator to feedback error.

To detect delayed drift beyond the optimized support, each checkpoint is also evaluated on one uninterrupted twelve-step trajectory. We use

$$\begin{aligned} \kappa_{\mathrm{sel}}(r) &= \tfrac{1}{2}\kappa_6(r) + \tfrac{1}{2}\kappa_{12}(r), \\ S_{\mathrm{FHTK}} &= \sum_{r=1}^{\infty} \kappa_{\mathrm{sel}}(r)\ell^{\mathrm{val}}_r = \tfrac{1}{2}(\mathcal{L}^{\mathrm{val}}_6 + \mathcal{L}^{\mathrm{val}}_{12}). \end{aligned} \tag{17}$$

This nested kernel assigns 75% of its mass to steps 1–6 and 25% to steps 7–12, preserving local accuracy while screening delayed instability. The selected model is iterated for H steps at test time without resetting or using test trajectories for model selection.

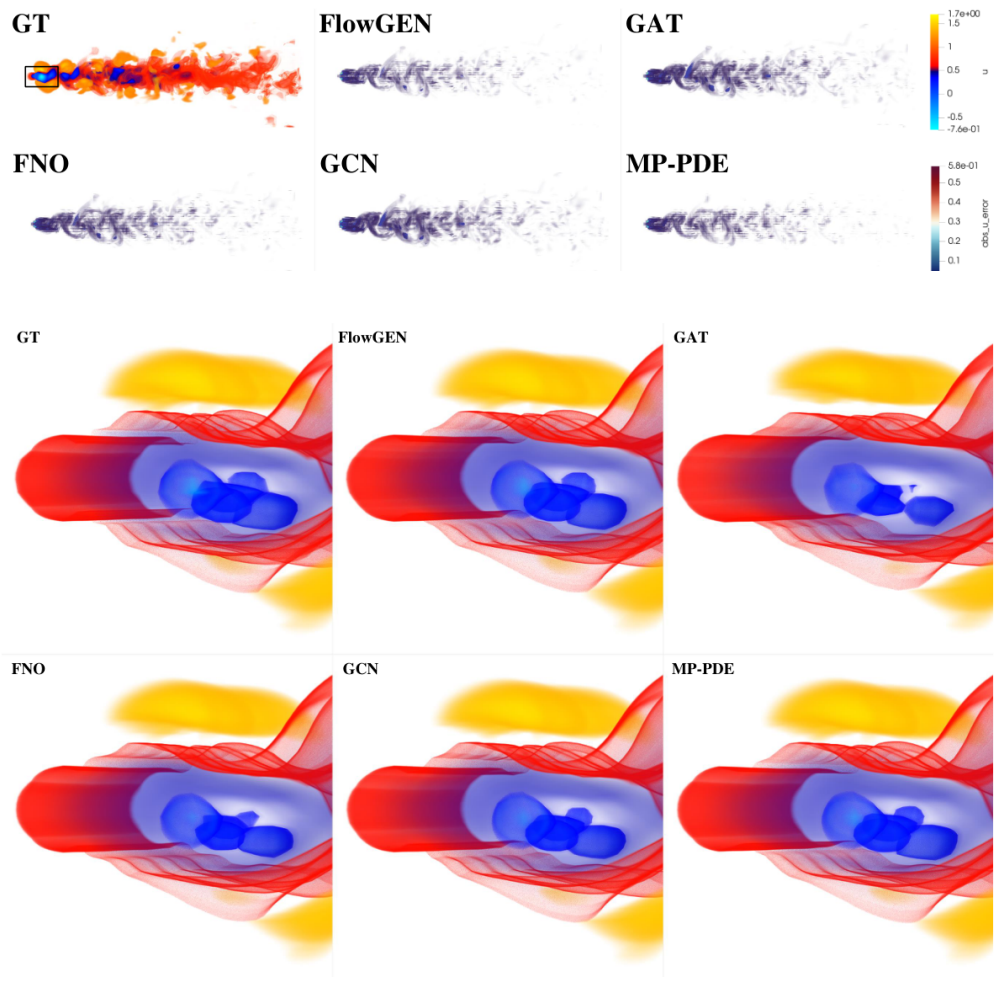


Figure 2: Comparison of the predictions and errors of different models for the unstructured-mesh case with $Re = 6400$. The top image shows the overall flow fields and error distributions, while the bottom image provides a magnified view of the flow around the cylinder. Compared with the other models, FlowGEN produces the flow field most consistent with GT.

4 Experiments

4.1 Experimental Setup

Dataset and Data Split. We evaluate FlowGEN on structured and unstructured representations of the Gerris flow dataset at four Reynolds numbers, $Re \in \{160, 320, 640, 6400\}$. This setting produces eight mesh–flow conditions. For each sequence, the first 75% of the frames are divided into six contiguous temporal blocks. Blocks 2 and 5 are used for validation, and the remaining four blocks are used for training. Prediction windows remain within individual blocks. The final 25% of each sequence forms one contiguous test trajectory. This blocked split prevents overlapping temporal windows from being randomly assigned to different subsets.

Baselines. We compare FlowGEN with GCN, GAT, FNO, and MP-PDE, covering graph convolution, graph attention, neural operators, and message-passing PDE surrogates. All methods share the three-frame history, sampled points, normalization, temporal split, residual output, and base loss coefficients. FlowGEN additionally uses FHTK, mixed Div regularization, and Div warmup, whereas the main baselines use absolute-field Div regularization. Loss-transfer experiments apply the proposed regularization to GCN, GAT, and FNO to separate architecture and objective effects.

Mesh	Re	EGR@35	$\text{MSE}_{7:12}$	$\text{KE}_{1:35}$	$\text{Div}_{1:35}$
Structured	160	**41.15** / 66.41	**1.090** / 1.094	**1.743** / 1.783	**0.1399** / 0.1449
	320	**22.75** / 22.79	**0.911** / 0.971	**2.389** / 2.417	**0.1634** / 0.1723
	640	**9.70** / 9.94	**0.861** / 0.871	**1.585** / 1.616	**0.1665** / 0.1742
	6400	**5.94** / 6.45	**0.753** / 0.794	**1.589** / 1.688	**0.1707** / 0.1746
Unstructured	160	**39.14** / 61.58	**4.034** / 4.989	**3.008** / 4.062	**0.3317** / 0.3422
	320	**14.45** / 15.00	**3.045** / 3.106	**2.087** / 2.150	**0.3482** / 0.3550
	640	**7.32** / 7.51	**2.512** / 2.547	**3.731** / 3.838	**0.3391** / 0.3504
	6400	**4.22** / 4.39	**1.980** / 2.170	**2.469** / 2.553	**0.3199** / 0.3203

Table 1: Rollout comparison across eight conditions. Each entry is FlowGEN / the strongest baseline for that metric (lower is better). MSE and KE are scaled by 10^2. The full five-model results with standard deviations are in Table 4.

Implementation Details. Experiments are conducted on four NVIDIA A40 GPUs with 48 GB memory each, with one GPU per run. All models use width 192 and are trained for 40 epochs using AdamW with batch size 8, an initial learning rate of 10^{-3}, and weight decay of 10^{-5}. Cosine annealing reduces the learning rate to 5×10^{-5}, and automatic mixed precision is used. Results are reported over three independent seeds.

4.2 Evaluation Protocol

The selected checkpoint is evaluated on a contiguous 35-step autoregressive trajectory. Let the velocity error at rollout step s be

$$e_s = \frac{1}{3N} \sum_{i=1}^{N} \|\widehat{\mathbf{u}}_{s,i} - \mathbf{u}_{s,i}\|_2^2 . \tag{18}$$

We quantify temporal error amplification using

$$\text{EGR@35} = \frac{e_{35}}{e_1}. \tag{19}$$

A lower EGR indicates weaker error growth under recursive feedback.

4.3 Comparison

Prediction errors generally increase with the forecast horizon and are larger on unstructured trajectories. FlowGEN leads across mesh types, Reynolds numbers, and all reported metrics, with its largest advantage in recursive error growth. Table 1 summarizes the quantitative results, while Fig. 2 compares the predicted fields and errors.

Behavior Across Reynolds Numbers. The largest differences occur at $Re = 160$, where all methods experience the strongest error amplification. On the structured representation, the best baseline reaches an EGR of 66.41, while FlowGEN reduces it to 41.15. This corresponds to a 38.0% reduction. On the unstructured representation, FlowGEN reduces the best baseline EGR from 61.58 to 39.14, giving a 36.4% reduction. Its unstructured KE error is also 25.9% lower than the best baseline. As Re increases, EGR decreases for all models and the gap becomes smaller. This pattern indicates that the largest advantage of FlowGEN appears when autoregressive accumulation is the dominant source of error. FlowGEN still retains the best mean at $Re = 320$, 640, and 6400, showing that the improvement is not restricted to the most unstable condition.

Structured and Unstructured Representations. The unstructured cases have substantially larger absolute MSE values than the structured cases, especially after repeated feedback. This difference reflects the additional difficulty introduced by irregular spatial sampling and temporal alignment. FlowGEN preserves its leading performance under this change in representation. When the four Reynolds numbers are averaged, its unstructured EGR is 27.0% lower than the best baseline. Its unstructured KE error is 34.6% lower. The corresponding improvements on the structured representation are 24.7% and 17.7%. The larger physical gain on the unstructured cases shows that the method does not rely on regular-grid structure to preserve rollout behavior.

k	2	4	6	8	12	16	20
Structured ↓	0.7292	0.7114	0.6910	0.6790	**0.6626**	0.6646	0.6697
Unstructured ↓	0.6036	0.5763	0.5679	0.5560	**0.5407**	0.5435	0.5421

Table 2: Validation-only neighborhood sensitivity at $Re = 320$.

	Structured			Unstructured		
Variant	MSE	KE	Div	MSE	KE	Div
W/O KE	1.1738	2.0316	0.2215	4.2458	3.2539	0.5029
W/O Div	1.2036	2.4698	0.2273	4.4659	3.6441	0.5233
W/O FHTK	1.3471	4.5859	0.2180	4.3029	1.6592	0.5136
Ours	**1.1228**	**2.0140**	**0.2064**	**4.1483**	**1.4388**	**0.4131**

Table 3: Physics and FHTK ablations at $Re = 320$. MSE and KE values are scaled by 10^2; lower is better.

4.4 State-and-Credit Truncation Analysis

The horizon partition in Table 4 primarily evaluates the forecast-offset support of FHTK. It separates local transition accuracy from error accumulated beyond the training credit horizon. The three-state lag support is held fixed for all FlowGEN runs, so its isolated contribution is examined through the history-length ablation described below rather than inferred from the horizon table. FlowGEN already obtains the lowest average MSE over steps 1–6, but its largest aggregate advantage occurs over steps 7–12. Its MSE in this range is 8.6% lower than that of MP-PDE, compared with a 3.8% difference over steps 1–6. This boundary behavior directly reflects the two FHTK supports. Steps 1–6 lie inside the final training kernel, whereas steps 7–12 are excluded from training but receive 25% of the nested selection-kernel mass. The increased margin immediately beyond step 6 indicates that extended-support selection is not merely improving transitions already optimized by backpropagation. It rejects checkpoints whose local fit is acceptable but whose recursively induced bias emerges after the training truncation boundary. The advantage remains positive over steps 13–35, which lie outside both the training and validation supports. FlowGEN reduces the average MSE in this range by 3.8% relative to MP-PDE and lowers EGR from the strongest baseline average of 26.64 to 18.08. The combination of lower intermediate-horizon MSE and lower EGR shows that the selected operator generalizes beyond the explicit kernel supports and amplifies recursive error more slowly over the complete test trajectory.

4.5 Ablation Studies

Tables 2 and 3 show the neighborhood-size selection and the effects of KE, Div, and the selection kernel. Detailed analysis is provided in Appendix B.

5 Conclusion

FlowGEN reduces field errors, physical deviations, and the error growth ratio throughout recursive forecasting. The experiments show that FHTK improves the learned transition operator beyond the forecast horizons explicitly included in the optimization. These results establish finite-horizon truncation as a practical principle for stable long-horizon neural flow forecasting. Additional results are provided in the appendix.

AI use statement

We used ChatGPT to assist with manuscript organization and language editing, presentation of the method and results, LaTeX template conversion, table layout, and reconstruction of figure crops and bibliography entries from an earlier manuscript. The scientific claims and numerical results originate from the authors' research. The authors are responsible for verifying AI-assisted material against the original results and sources before submission.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Han Gao, Luning Sun, and Jian-Xun Wang. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, 428:110079, 2021.

Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, 2020. doi: 10.1016/j.jcp.2019.109056.

Ali Kashefi and Tapan Mukerji. Physics-informed PointNet: A deep learning solver for steady-state incompressible flows and thermal fields on multiple sets of irregular geometries. *Journal of Computational Physics*, 468:111510, 2022.

Ali Kashefi, Davis Rempe, and Leonidas J. Guibas. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2):027104, 2021.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

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

Nikola B. 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.

Alex M. Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Zongyi Li, Nikola Kovachki, Christopher Choy, Boyi Li, Jean Kossaifi, Subhro Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Phillip Lippe, Bastiaan Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

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

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.

Pu Ren, Chengping Rao, Yang Liu, Jian-Xun Wang, and Hao Sun. PhyCRNet: Physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. *Computer Methods in Applied Mechanics and Engineering*, 389:114399, 2022. doi: 10.1016/j.cma.2021.114399.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 8459–8468, 2020.

Kim Stachenfeld, Drummond Buschman Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned simulators for turbulence. In *International Conference on Learning Representations*, 2022.

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.

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3424–3433. PMLR, 2017.

Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Nils Wandel, Michael Weinmann, and Reinhard Klein. Learning incompressible fluid dynamics from scratch: Towards fast, differentiable fluid models that generalize. In *International Conference on Learning Representations*, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38 (5):146:1–146:12, 2019.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 53681–53705, 2024.

A Full Rollout Results

Re	Model	EGR@35 ↓	$\text{MSE}_{1:6}$ ↓	$\text{MSE}_{7:12}$ ↓	$\text{MSE}_{13:35}$ ↓	$\text{KE}_{1:35}$ ↓	$\text{Div}_{1:35}$ ↓
			Structured				
160	GCN	66.41±12.08	0.427±0.035	1.094±0.047	1.644±0.038	3.159±2.203	0.1465±0.0081
	GAT	77.83±9.32	0.384±0.031	1.098±0.052	1.638±0.071	4.102±1.101	0.1467±0.0035
	FNO	81.79±8.46	0.439±0.027	1.129±0.074	1.632±0.142	1.783±0.832	0.1532±0.0143
	MP-PDE	92.60±19.26	0.381±0.030	1.162±0.146	1.726±0.278	2.210±1.213	0.1449±0.0113
	FlowGEN (Ours)	**41.15±6.41**	**0.377±0.065**	**1.090±0.206**	**1.631±0.164**	**1.743±0.559**	**0.1399±0.0112**
320	GCN	22.79±0.84	0.661±0.002	1.108±0.042	1.809±0.180	3.407±0.797	0.1913±0.0099
	GAT	25.81±1.70	0.600±0.016	1.050±0.037	1.851±0.199	4.530±1.154	0.1913±0.0064
	FNO	27.61±3.20	0.569±0.018	1.418±0.119	1.952±0.173	2.417±0.847	0.2202±0.0279
	MP-PDE	32.55±1.16	0.456±0.021	0.971±0.039	1.683±0.113	2.563±0.340	0.1723±0.0066
	FlowGEN (Ours)	**22.75±2.00**	**0.424±0.038**	**0.911±0.057**	**1.616±0.095**	**2.389±0.981**	**0.1634±0.0033**
640	GCN	9.94±0.82	0.653±0.007	1.035±0.050	1.436±0.040	1.616±0.958	0.1880±0.0080
	GAT	10.64±1.73	0.587±0.035	0.960±0.040	1.433±0.134	1.691±1.032	0.1894±0.0092
	FNO	10.59±0.26	0.595±0.049	1.123±0.029	1.521±0.046	2.702±1.321	0.2134±0.0096
	MP-PDE	12.65±1.38	0.492±0.017	0.871±0.051	1.273±0.122	2.340±0.667	0.1742±0.0043
	FlowGEN (Ours)	**9.70±0.53**	**0.471±0.005**	**0.861±0.007**	**1.253±0.077**	**1.585±0.480**	**0.1665±0.0036**
6400	GCN	6.45±0.27	0.723±0.035	0.909±0.017	1.158±0.014	1.709±0.483	0.2010±0.0059
	GAT	6.58±3.11	0.654±0.032	0.868±0.004	1.323±0.316	1.688±0.541	0.2038±0.0042
	FNO	6.48±0.17	0.599±0.027	0.885±0.046	1.161±0.067	2.977±0.854	0.2032±0.0107
	MP-PDE	7.56±1.15	0.541±0.022	0.794±0.017	1.039±0.079	1.762±1.090	0.1746±0.0033
	FlowGEN (Ours)	**5.94±0.85**	**0.507±0.032**	**0.753±0.025**	**1.036±0.115**	**1.589±0.834**	**0.1707±0.0067**
			Unstructured				
Re	Model	EGR@35 ↓	$\text{MSE}_{1:6}$ ↓	$\text{MSE}_{7:12}$ ↓	$\text{MSE}_{13:35}$ ↓	$\text{KE}_{1:35}$ ↓	$\text{Div}_{1:35}$ ↓
160	GCN	79.53±13.15	1.403±0.025	5.267±0.124	6.187±0.313	6.355±2.779	0.3534±0.0112
	GAT	108.60±23.94	1.410±0.157	5.148±0.278	6.326±0.251	13.599±8.636	0.3659±0.0159
	FNO	61.58±7.76	1.672±0.085	6.338±0.122	6.200±0.357	4.062±1.448	0.4146±0.0146
	MP-PDE	80.95±13.77	1.495±0.036	4.989±0.142	6.198±0.686	9.000±5.588	0.3422±0.0291
	FlowGEN (Ours)	**39.14±6.36**	**1.400±0.084**	**4.034±0.371**	**6.173±0.492**	**3.008±2.410**	**0.3317±0.0101**
320	GCN	15.00±0.89	2.032±0.063	3.829±0.044	5.238±0.051	7.570±1.370	0.3880±0.0022
	GAT	15.13±1.42	1.853±0.113	3.343±0.213	6.015±0.140	5.646±3.151	0.4143±0.0167
	FNO	15.72±1.65	2.024±0.149	3.794±0.486	5.106±0.363	5.447±2.742	0.4550±0.0443
	MP-PDE	23.23±3.30	1.486±0.043	3.106±0.136	5.453±0.590	2.150±0.980	0.3550±0.0160
	FlowGEN (Ours)	**14.45±1.74**	**1.484±0.036**	**3.045±0.208**	**5.088±0.463**	**2.087±1.182**	**0.3482±0.0108**
640	GCN	8.15±0.35	2.027±0.077	2.826±0.032	3.785±0.101	5.343±1.442	0.3733±0.0056
	GAT	8.85±0.33	1.879±0.008	2.796±0.092	3.963±0.160	6.744±2.990	0.3959±0.0020
	FNO	7.51±0.15	2.038±0.109	2.698±0.119	3.601±0.097	3.838±1.809	0.3629±0.0179
	MP-PDE	9.16±0.77	1.616±0.050	2.547±0.149	3.654±0.198	4.953±3.110	0.3504±0.0023
	FlowGEN (Ours)	**7.32±0.20**	**1.575±0.107**	**2.512±0.144**	**3.578±0.107**	**3.731±3.136**	**0.3391±0.0103**
6400	GCN	4.83±0.66	2.002±0.017	2.438±0.021	2.861±0.213	2.553±0.788	0.3791±0.0034
	GAT	7.16±1.87	1.846±0.042	2.293±0.024	3.567±0.571	10.569±5.118	0.3946±0.0017
	FNO	4.39±0.14	1.795±0.047	2.202±0.015	2.440±0.058	3.919±1.205	0.3203±0.0066
	MP-PDE	6.43±0.77	1.515±0.066	2.170±0.007	2.523±0.315	5.104±3.955	0.3412±0.0084
	FlowGEN (Ours)	**4.22±0.05**	**1.442±0.033**	**1.980±0.059**	**2.281±0.042**	**2.469±1.808**	**0.3199±0.0091**

Table 4: Thirty-five-step autoregressive rollout results on structured and unstructured samples at four Reynolds numbers. MSE and KE values are scaled by 10^2. EGR and Div error are unscaled. Values are mean ± sample standard deviation over three seeds. The best mean for each mesh–Re–metric combination is shown in bold.

B Additional Experimental Results

B.1 Ablation Studies

Neighborhood Size. Table 2 evaluates neighborhood size using validation data. Increasing k from 2 to 12 reduces the score by 9.1% on structured data and 10.4% on unstructured data, while $k = 16$ and $k = 20$ provide no further improvement. We therefore use $k = 12$ for both mesh types.

Physics and FHTK Selection. Table 3 shows that the complete FlowGEN configuration obtains the lowest error in every mesh–metric column. The effects of the individual components differ across representations. Removing the KE term has a modest effect on the structured case, but increases unstructured KE error by 126.2% and Div error by 21.7%. The KE constraint therefore becomes more important when the spatial samples are irregular and physical deviations are more difficult to control. Removing Div regularization increases rollout MSE by 7.2% on the structured representation and by 7.7% on the unstructured representation. It also increases Div error by 10.1% and 26.7%, respectively. These changes show that the Div objective contributes to both local physical consistency and long-horizon field accuracy. Removing the extended component of the selection kernel ($\kappa_{\mathrm{sel}} \to \kappa_6$) produces the largest degradation on the structured case. Its MSE increases by 20.0%, and its KE error increases by 127.7%. On the unstructured case, Div error increases by 24.3%. A kernel restricted to the six-step training support can therefore select checkpoints with acceptable local accuracy but larger delayed physical drift. The extended support rejects such checkpoints without using the test trajectory.