Inpainting physics: self-supervised learning for context-driven fluid simulation

Jonas Weidner[1,2], **Yeray Martin-Ruisanchez**[1], **Daniel Rueckert**[1,2,3], **Benedikt Wiestler**[1,2,*], **Julian Suk**[1,2,*]

[1] Technical University of Munich (TUM)
[2] Munich Center for Machine Learning (MCML)
[3] Imperial College London
[*] Shared senior authorship
j.weidner@tum.de

Abstract

Neural surrogate models for computational fluid dynamics (CFD) are typically trained as forward operators that map explicit problem specifications, such as geometry and boundary conditions, to solution fields. This ties the model to the conditioning variables seen during training and limits reuse under boundary-condition shifts or local geometry changes. We propose to reformulate steady CFD inference as an inpainting problem: instead of training on explicit boundary conditions, we learn a self-supervised prior over velocity fields and impose boundary constraints only during inference by fixing known regions such as inlet, outlet or unchanged regions from previous simulations. To scale this idea to large 3D meshes, we introduce a local neighbourhood tokeniser that represents high-resolution velocity fields as compact spatial latent tokens and train latent flow-matching and masked-autoencoder models on these tokens. On simulations of intracranial aneurysm flow, our method reconstructs full velocity fields from sparse boundary context, outperforms supervised neural surrogates under boundary-condition and dataset shift and enables local geometry editing by reusing unchanged simulation context. These results suggest that viewing CFD inference as context-conditioned inpainting can turn neural surrogates from task-specific predictors into reusable flow priors. Code and demo are available at inpainting-physics.com.

TL;DR Train a self-supervised latent model on CFD velocity fields, impose boundary conditions at inference, predict the flow by inpainting, generalise more flexibly to unseen geometries and flow speeds.

1 Introduction

Computational fluid dynamics (CFD) is a central tool in, e.g., automotive engineering, climate science and biomedical modelling, but high-fidelity numerical simulations remain expensive [Ashton et al., 2024, Bodnar et al., 2025, Nannini et al., 2024]. Each new geometry, boundary condition or design variant typically requires solving a large discretised system from scratch. Neural surrogates aim to reduce this cost by learning an approximate map from problem specification to quantities of interest.

Neural surrogates for CFD are typically trained as supervised neural operators: given a defined geometry, physical parameters and boundary conditions, they directly regress the corresponding velocity or pressure field [Suk et al., 2024, Luo et al., 2025, Alkin et al., 2025]. This formulation is powerful in-distribution, but strongly couples the learned model to the conditioning variables and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

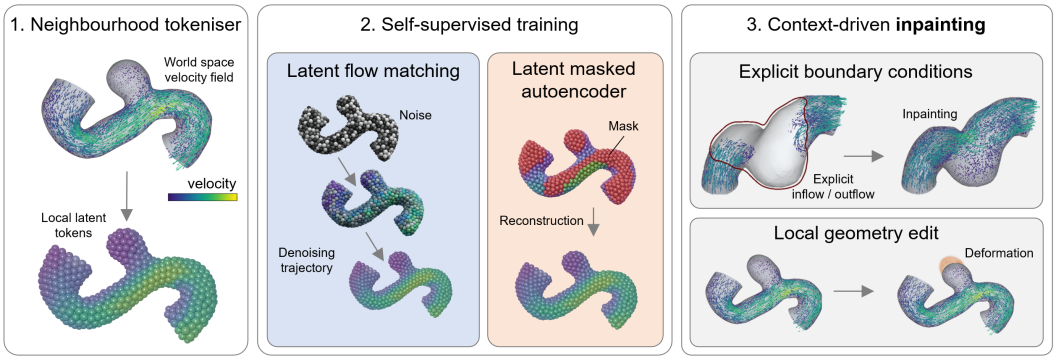

Figure 1: **Inpainting physics.** (1) We tokenise raw velocity fields into local ball-shaped latent representations. (2) We train a self-supervised model on these tokenised velocity fields using latent flow matching or a masked autoencoder. (3) At inference, known regions (e.g., inflow and outflow, measurements, or unchanged parts of a previous simulation) are fixed as context and the rest is inpainted, enabling generalisation to unseen geometries and flow conditions, preserving local flow structure.

data-generating process used during training. As a result, extrapolation to new boundary conditions, geometries or solver settings remains a major obstacle.

We begin by observing that laminar fluid flow correlates highly with its spatial context. Due to viscosity effects and momentum, the union of upstream and downstream dynamics is an excellent prior for local flow.[1] Despite this, supervised neural operators do not explicitly leverage spatial context when producing flow estimates. We argue that including spatial context could greatly improve out-of-distribution performance in neural surrogates.

Inspired by the success of generative modelling in computer vision [Rombach et al., 2022], we formulate (steady) CFD emulation not as deterministic regression but as (probabilistic) generation. In other words, we recast the supervised learning of flow states based on their underlying Dirichlet boundary conditions to self-supervised, *unconditional* learning in conjunction with *conditioning at inference time*. Our goal is to learn a reusable prior over all flow states and impose Dirichlet boundary conditions only at inference time via *inpainting*. The generative model can then leverage upstream and downstream observations and infer dynamical parameters implicitly from the context.[2]

This separation of representation learning and context-driven inference changes how neural surrogates can be used. A pre-trained model can be queried with different conditioning patterns: sparse inlet-outlet constraints (similar to the supervised setting) as well as larger measured or otherwise known spatial context. Furthermore, the inpainting formulation enables local editing, i.e., adaptation of a given simulation to small geometric perturbations of the flow domain.

Contributions. We (i) propose to shift the modelling objective and view CFD emulation as context-conditioned inpainting rather than supervised neural operator modelling, allowing conditioning through observed regions, (ii) show that this approach has favourable generalisation properties for changing boundary conditions and shape distribution, and (iii) propose local editing as a preliminary downstream task tailored to generative models.

2 Related work

Neural surrogates for science and engineering have recently been gaining traction. Their underlying idea is to learn a mapping from the problem parameters to the quantities of interest in order to approximate high-fidelity numerical simulation in certain regimes where data is abundant. Proofs of concept exist, e.g., for weather forecasting [Bodnar et al., 2025], automotive aerodynamics [Alkin et al., 2025], high-energy physics [Spinner et al., 2024] and cardiovascular flow [Suk et al., 2024].

[1] The totality of such local interactions are the basis of continuum-based fluid simulation.

[2] We remark that the inpainting formulation smoothly transitions into the regression formulation if we increase the size of the inpainting region (until the Dirichlet boundary).

Neural operators. Popularised by architectures like DeepONet [Lu et al., 2021] and Fourier Neural Operator [Li et al., 2021], “neural operators" has become an umbrella term for models that map between continuous functions. In the context of neural surrogates, these models are typically set up as forward operators, mapping geometric and physical inputs to the solution field [Luo et al., 2025, Alkin et al., 2025, Pepe et al., 2025, Holzschuh et al., 2025, Wen et al., 2026, Holzschuh et al., 2026]. Conceptually, these approaches are designed for regular grids or point clouds extracted from 3D surface or volume meshes. While grid-based methods lend themselves to Euclidean domains where isotropically high resolution is required, e.g. freestream flow within a volume element, point cloud methods can efficiently model irregular structures and their interaction with the medium, e.g., blood flow in human vasculature.

Generative models. Due to their efficacy in domains like image processing and computer vision, generative models are gradually incorporated into neural surrogates. Diffusion models have been proposed for temporal forecasting of solutions to partial differential equations (PDE) where the main difficulty lies in achieving stable rollouts [Sun et al., 2023, Lippe et al., 2023, Shysheya et al., 2024, Rozet et al., 2026]. Notably, Rozet et al. [2026] highlight the efficacy of encoding the data into a unified latent space in which system dynamics are generated. We adopt latent space modelling as a design choice in our approach.

Instead of autoregressively evolving an initial state over time, Lienen et al. [2024] propose to learn the manifold of possible flow states in a certain (turbulent) regime constituted by the training data. In contrast to our approach, Lienen et al. [2024] impose Dirichlet boundary condition already during training and focus on directly modelling the distribution of turbulence rather than emulating a numerical solver. Lino et al. [2025] propose to generate converged flow states via graph-based latent diffusion conditioned on parameters of the dynamics, such as Reynolds number. While this more closely emulates a numerical solver, Lino et al. [2025] also impose Dirichlet boundary conditions during training which means their model is tied to these parameters, different from our context-driven approach. Du et al. [2024] propose an unconditional approach to temporal turbulence modelling using a latent space learned via conditional neural fields. In contrast to their approach, we include variations in mass flow and domain geometry in our framework while Du et al. [2024] focus on generating complex turbulence patterns within a fixed setting.

Besides diffusion (or flow matching) models, generative adversarial networks (GAN) have been explored in the context of fluid modelling [Buzzicotti et al., 2021, Drygala et al., 2022]. In addition to these two approaches, inference-time flow estimation via inpainting has been previously studied [Buzzicotti et al., 2021, Du et al., 2024]. However, those works are limited to grid-based data (i.e., images). In contrast, we use inpainting as the conditioning mechanism itself, replacing explicit boundary-condition parameters, and lay out a framework for inpainting on point clouds from 3D volume meshes, which is the dominant data modality for state-of-the-art science and engineering problems. Additionally, our work is unique in identifying the utility of masked autoencoders (MAE) [He et al., 2022] as a generative backbone.

3 Methods

3.1 Datasets

Similar approaches to ours commonly study automotive aerodynamics in the context of neural surrogates [Luo et al., 2025, Alkin et al., 2025], characterised by freestream flow around an object. In order to clearly demonstrate the mechanism of our approach, we seek an application in which (laminar) dynamics are dominated by upstream velocity rather than domain geometry. For our application, arterial blood flow is particularly well-suited, with two datasets of intracranial aneurysm fluid simulations [Li et al., 2025, Ding et al., 2025] available. These large-scale public datasets allow for benchmarking and out-of-distribution testing, especially since their samples vary in mass flow and shape.

Specifically, we train and evaluate our method on the Aneumo dataset [Li et al., 2025]. The data contains multiple slightly adapted aneurysm geometries derived from real patients. These samples are, therefore, an ideal test bed for studying both full-velocity field generation and local editing around the aneurysms. The dataset contains 85,280 samples of steady-state fluid simulations across 8 total mass flows, the mass that flows through the artery per given time. Units for velocities are [m/s] and

mass flow [g/s] if not stated otherwise. We train our model on 4 different mass flows (2, 2.5, 3, 3.5) and keep (1, 1.5, 3.75, 4) for testing. For all models, we center the simulation at the centre of mass and add a small translation augmentation. To prevent data leakage, we split the data sequentially (test 1 - 1,093, evaluating on a uniformly sampled subset of 100 due to L-FM integration cost; validation 1,094 - 2,158; training 2,159 - 10,660), as clusters of adjacent samples have similar geometry.

As an additional test, we evaluate our methods on the transient dataset AneuG [Ding et al., 2025], averaging over all timesteps to obtain a stationary estimate. The mass flows per artery are not provided; thus, we estimate a constant scaling factor of 86.97 based on the mean velocity fitted to Aneumo. AneuG uses non-Newtonian Carreau–Yasuda blood, while Aneumo uses a Newtonian fluid. Reference solutions further come from different solver stacks. We use AneuG strictly as a held-out test set to assess model generalisation, with no cases seen during training or validation.

3.2 Model

We formulate CFD prediction as conditional completion under a self-supervised velocity field prior. The model is trained on complete velocity fields v without explicit boundary-condition labels, such as mass flow, inlet profiles or outlet constraints. At inference time, known solution regions are fixed as spatial context and the model reconstructs the unknown (masked) region, estimating $v_{\mathcal{U}} \mid v_{\mathcal{K}}$, where $\mathcal{K}$ denotes known regions such as inlet, outlet, measured, or unchanged simulation regions, and $\mathcal{U}$ denotes the region to be inpainted.

Our method has three stages, illustrated in Figure 1: a local neighbourhood tokeniser compresses the unstructured velocity field into spatial latent tokens (Sec.3.2.1); a self-supervised generative model, either latent flow matching or a latent masked autoencoder, is trained on these tokens (Sec.3.2.2); and inference enforces known context by fixing the corresponding tokens while inpainting the missing region (Sec. 3.2.3).

3.2.1 Neighbourhood tokeniser

The velocity field is represented on an unstructured three-dimensional point cloud. Each point i has position $x_i \in \mathbb{R}^3$ and velocity $v_i \in \mathbb{R}^3$. We additionally provide each point with its Euclidean distance to the closest physical wall, $d_i \in \mathbb{R}$. Direct transformer modelling of the full point cloud is infeasible because each simulation contains approximately 200k points. Moreover, our inpainting formulation requires spatially localised tokens: when a boundary or edited region is fixed, the representation should be constrained to this specific location. We therefore introduce a local neighbourhood tokeniser that maps the dense velocity field to a set of compact latent tokens.

We first select $P = 2500$ neighbourhood centres x_j^p using farthest point sampling [Eldar et al., 1997]. Around each centre, we collect the 512 nearest points. Each point in a neighbourhood is represented by its local coordinate relative to the neighbourhood centre normalised by the radius r (distance to furthest point from the centre), its wall distance, and its velocity:

$$[\tilde{x}_i, d_i, v_i] \in \mathbb{R}^7, \qquad \tilde{x}_i = (x_i - x_j^p)/r. \tag{1}$$

A PointNet-style encoder [Qi et al., 2017a] with shared MLP layers maps the 512 point features in each neighbourhood to pointwise features, followed by max pooling to obtain one latent token $z_j \in \mathbb{R}^{256}$. The decoder reconstructs local velocities by broadcasting z_j to query points and concatenating the local geometric descriptor $(\tilde{x}_i, d_i)$. Each neighbourhood is decoded independently. When a point belongs to multiple decoded neighbourhoods, we combine predictions using inverse-distance weighting in world space. Details are provided in Appendix C. This produces a full-resolution velocity field while preserving the locality needed for inpainting. The tokeniser is trained independently with a mean-squared-error reconstruction objective. Its role is not to solve the CFD task, but to provide a compact, smooth, and spatially localised latent space in which global generative models can operate efficiently.

3.2.2 Self-supervised model

After tokenisation, each simulation is represented by a set of latent tokens Z. The self-supervised model is trained on these latent fields. It receives token positions and wall distances to learn spatially plausible velocity field representations, but it does not receive total mass flow, inlet velocity, outlet

conditions, or a global geometry descriptor. We therefore call the model unconditional with respect to boundary conditions and simulation parameters.

We evaluate two complementary self-supervised backbones: latent flow matching and a latent masked autoencoder.

Latent flow matching. The latent flow-matching model learns a continuous transport from a simple noise distribution to the distribution of latent velocity fields. For each latent field Z, we sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ with the same shape as the latent tokens and time $t \sim \mathrm{LogitNormal}(0, 1)$. We define the linear interpolant

$$z_t^p = (1 - t)\epsilon + tz^p. \tag{2}$$

The model f_θ is trained to predict the velocity:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,\epsilon,z^p}\left[\|f_\theta(z_t^p, t, x^p, d^p) - (z^p - \epsilon)\|_2^2\right]. \tag{3}$$

The backbone is a DiT-style transformer [Peebles and Xie, 2023] with adaLN-Zero conditioning on t. Spatial positions and wall distances are encoded with sinusoidal embeddings. At inference time, unknown tokens are initialised with noise or zeros and integrated toward the data distribution (Sec. 3.2.3), while known tokens are fixed to their encoded values throughout the integration.

Latent masked autoencoder. The latent masked autoencoder is trained directly for conditional completion. Given a latent field, we randomly mask a subset of token indices $\mathcal{M}$. The encoder receives only the visible tokens, together with their spatial and wall-distance embeddings. The decoder receives encoded visible tokens and learned mask tokens at the missing positions, and predicts the original latent tokens at the masked locations. The training objective is

$$\mathcal{L}_{\mathrm{MAE}} = \frac{1}{|\mathcal{M}|}\sum_{j\in\mathcal{M}} \|\hat{z}_j - z_j\|_2^2. \tag{4}$$

This objective closely matches our inference problem: given known regions of a velocity field, predict the missing region. Unlike the flow-matching model, the masked autoencoder does not require iterative sampling at inference time, making it a direct and efficient completion model. We implement a Transformer-based MAE [He et al., 2022].

3.2.3 Explicit boundary inpainting.

At inference time, boundary conditions and other known information are imposed by explicitly fixing the corresponding spatial tokens. We encode the known velocity values with the neighbourhood tokeniser and keep those latent tokens fixed. The self-supervised model then predicts the missing tokens in $\mathcal{U}$, after which the NT decoder reconstructs the full-resolution velocity field.

This procedure converts boundary-condition enforcement into an inpainting operation. Importantly, the same pretrained model can be used with different choices of $\mathcal{K}$ (Figure 1). For full-field prediction, $\mathcal{K}$ may consist only of sparse inlet and outlet regions. For local geometry editing, $\mathcal{K}$ can include most of the original simulation, while only the edited region is masked and regenerated.

We compare the following inference strategies: **L-MAE:** The masked region is predicted in one forward pass using the latent masked autoencoder. **L-MAE-Iterative:** The masked region is filled progressively from the boundary of the known region. We iterate over 5 steps and separate the mask into 5 equal regions. **L-FM-Euler:** We integrate over the FM output starting from random noise using Euler integration with 20 steps while keeping the explicit boundary fix. **L-FM-Euler-Zero:** We integrate in the same way as FM-Euler, but we start at zero instead of Gaussian noise. **L-FM-Physics:** We guide the integration steps by conditioning on zero divergence as a hard side constraint [Utkarsh et al., 2026], which enforces an incompressible flow. Each FM step is extrapolated into world space, projected onto a solution that satisfies the side constraints, and interpolated back into the partial noise space. This forces the integration along physically plausible paths.

The key point is that boundary conditions are never learned as a fixed input format. They are supplied as observed parts of the solution field at inference time. This allows the model to behave like a reusable CFD prior rather than a task-specific forward surrogate.

3.3 Baselines and metrics.

We implement a state-of-the-art neural surrogate model as a supervised baseline. They receive the positions as input and output the velocity conditioned on the mass flow. Therefore, we adopt the anchor-branched universal physics transformer AB-UPT [Alkin et al., 2025], which has demonstrated strong performance on large point clouds as an attention-based supervised model (supervised-Att). Additionally, we add a message passing (supervised-MP) model based on PointNet++[Qi et al., 2017b]. As our task is defined solely in terms of the velocity field, we exclude the surface and geometry branches and the pressure prediction. We provide the exact same input as for our models x and d, but additionally also the Euclidean distance to the inlet d^{in} for orientation information and the total mass flow m conditioning. Further, we add the naive baselines: mean-inpainting, which sets all values to the mean, and interpolation, which interpolates based on distance to the nearest non-masked values.

We evaluate the predicted velocity fields along two complementary axes. First, we measure whether *the predicted velocity vectors match the reference CFD solution* at the original point-cloud locations. For this, we report normalised mean squared error (nMSE), which captures errors in velocity magnitude and components, and cosine similarity (CosSim), which measures directional agreement independently of scale. Second, we assess *local physical consistency*. Since the velocity field represents an incompressible flow, we estimate the normalised divergence from local neighbourhoods as a scale-invariant measure of violation of the continuity constraint. In addition, we compare the predicted and reference vorticity fields to evaluate whether the model captures rotational flow structures. Together, these metrics distinguish numerical agreement with the reference solution from physically relevant properties of the reconstructed field. Exact definitions and implementation details are provided in Appendix A.

4 Experiments

We evaluate three central questions. First, we test whether a locally trained tokeniser can faithfully represent large velocity fields. Second, we investigate whether boundary-conditioned inpainting can recover full velocity fields under both in-distribution and out-of-distribution conditions. Third, we analyse whether inpainting can exploit partial context to adapt velocity fields under local geometry edits.

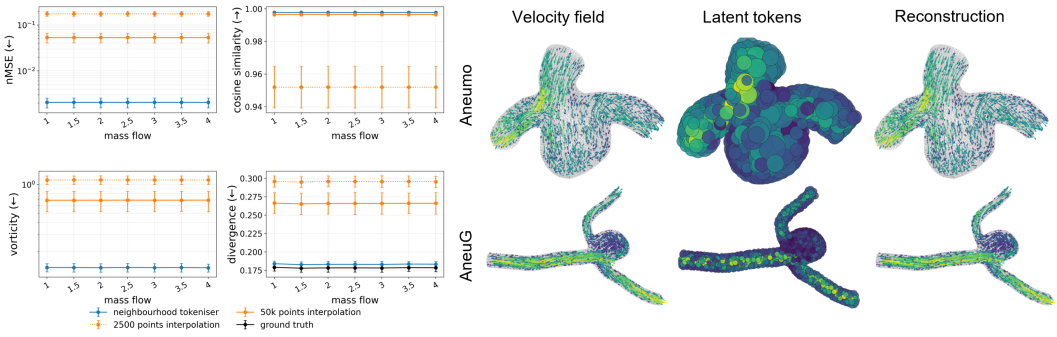

Figure 2: **The neighbourhood tokeniser demonstrates low reconstruction error over all mass flows.** We compare our tokeniser using 2500 tokens with naive baselines of random downsampling and re-interpolation. Additionally, we show examples of the latent ball representation and the reconstruction on both datasets.

Neighbourhood tokens accurately represent velocity fields. Our inpainting formulation requires a representation that is both compact and spatially local: direct transformer modelling of the full point cloud is infeasible for simulations with approximately 200k points due to the quadratic scaling of self-attention, and global pooling would destroy the locality needed to fix inlet, outlet, measurement, or unchanged simulation regions at inference time.

As shown in Figure 2, the neighbourhood tokeniser compresses each velocity field to 2500 local latent tokens while maintaining low reconstruction error across mass flows - even when they are

out-of-distribution. Compared with random downsampling and interpolation, the learned local representation reconstructs velocity magnitude and direction substantially more accurately. Qualitatively, it preserves the main flow structures on both Aneumo and the held-out AneuG dataset; on AneuG, the reconstruction nMSE is $(0.21 \pm 0.01)\%$. This suggests that *the tokeniser accurately captures local geometric and velocity structure in a way that transfers beyond the training geometries*. Since all subsequent models operate in this latent space, the reconstruction quality of the tokeniser defines an approximate lower bound on the error achievable by the inpainting models.

Explicit boundary-conditioned inpainting predicts velocity fields. We next test whether a self-supervised model trained without explicit boundary-condition labels can recover full velocity fields when boundary information is provided at inference time. We evaluate on Aneumo at mass flow $m = 3$, which lies inside the training range for all models. The mask fraction denotes the portion of the domain to be inpainted: at 99%, essentially only inlet and outlet velocities are fixed, while smaller masks (down to 5%) provide more context.

This setting favours the supervised neural surrogates, which are trained directly for forward prediction and receive the mass flow parameter explicitly. As shown in Figure 3, they therefore perform strongly under sparse (mask fraction 99%) in-distribution conditioning. The inpainting models show a different behaviour: their performance improves as more field context is provided. Latent flow matching struggles with very large masks but improves with additional context, and, as expected, physics guidance reduces divergence and vorticity. Further inverse-modelling choices, such as soft boundaries, iterative inpainting, noising the known tokens to the current time, or initialising with the L-MAE prediction, can further improve L-FM (Appendix Figure 9). The latent masked autoencoder is best aligned with the inference task: although slightly weaker than the supervised baseline at the sparsest inlet-outlet setting, it performs best across most metrics once additional context is available. Once about 10% of the domain is given as context (mask fraction $\leq 90\%$), L-MAE outperforms both supervised surrogates in nMSE, vorticity and divergence. In other words, context improves localised completion for the former while the latter shine under spatially averaged metrics. Iterative MAE helps for large masks, but can accumulate errors for smaller ones. Overall, these results show that *boundary-conditioned inpainting can recover full velocity fields from explicit solution contexts, and that its advantage grows as additional known regions become available*.

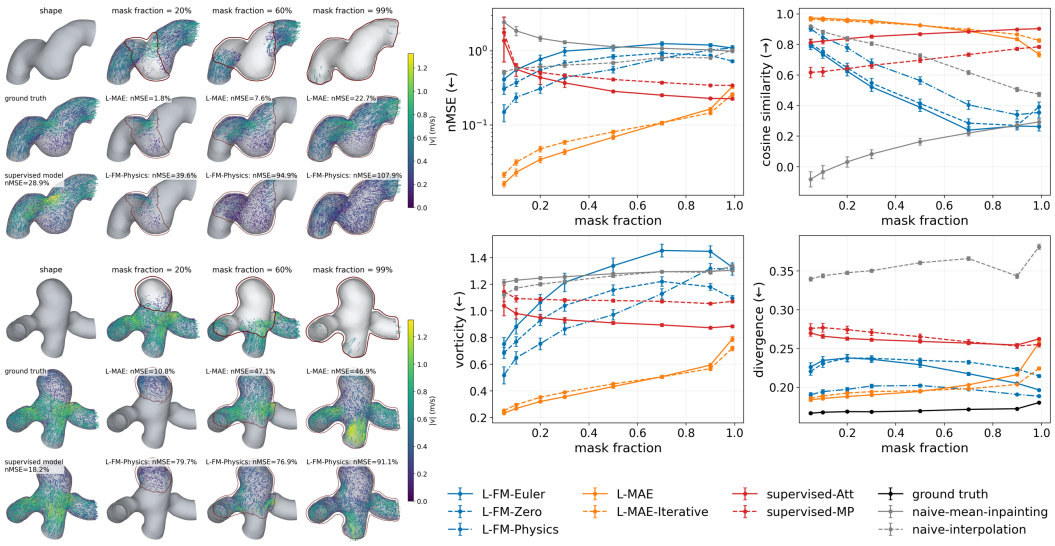

Figure 3: **Supervised models perform best on the forward prediction on in-distribution tasks. L-MAE and L-FM perform better with additional context.** We compare different supervised and self-supervised models on the prediction of velocity fields provided with varying amounts of context (higher masking fraction means less context). The maximum masking ratio is 99%. Metrics for the supervised models are computed within the mask region.

Inpainting improves out-of-distribution generalisation under boundary-condition and dataset shifts. A central motivation for our formulation is robustness to boundary-condition shift. Forward

surrogates must extrapolate in an explicit conditioning space, such as total mass flow. Inpainting instead represents new boundary conditions directly as observed velocity values, so the model solves a completion problem in the same field space used during self-supervised training, even though it hasn't seen these specific boundary conditions.

Figure 4 illustrates the limitation of explicit conditioning. Both supervised baselines are trained on mass flows between 2 and 3.5 and evaluated from 1 to 4. They remain accurate within the training range but collapse for out-of-distribution mass flows, especially for $m = 1$. In the inpainting formulation, such cases are not unseen scalar inputs but observed low-velocity boundary regions to be completed; note, however, that our model was still trained only on velocity fields from mass flows between 2 and 3.5.

We further test dataset shift on the external AneuG dataset (Appendix Figure 8), where geometries, solver stack, mesh resolution, and physical assumptions differ from Aneumo. Under this shift, the supervised forward surrogates degrade strongly and no longer consistently outperform even simple baselines. Among the learned inpainting methods, the latent MAE is the most robust and is the only model that consistently improves over the naive baselines across mask fractions. These results support a main premise of our approach: *conditioning through observed regions of the solution field can be more robust than relying solely on explicit training-time boundary-condition parameters*.

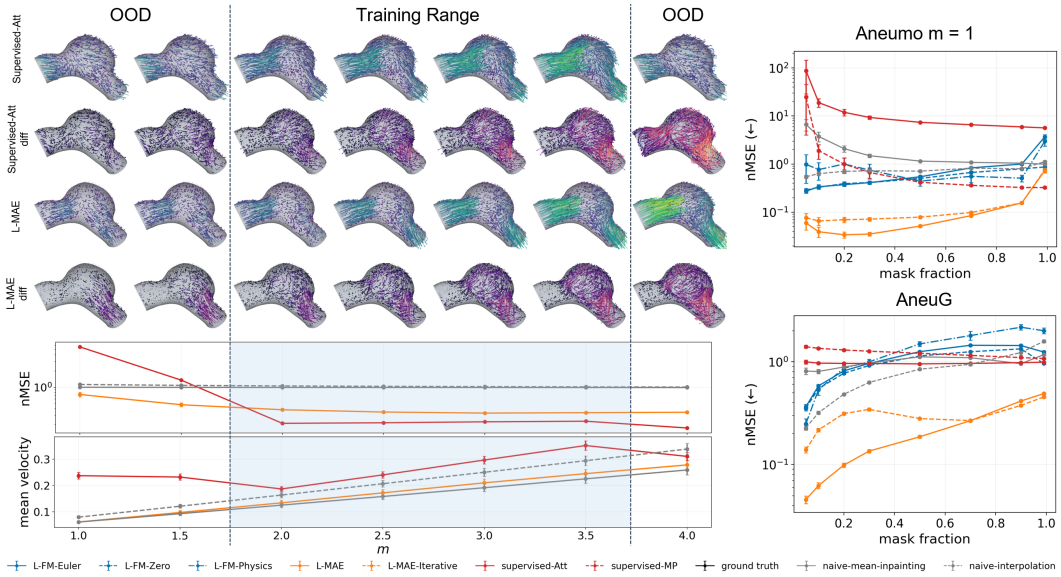

Figure 4: **The best supervised model (red) fails out-of-distribution (ood), while the L-MAE (orange) provides solid results, especially with context.** Left, we show the performance over all mass flows at a masking ratio of 99%. Supervised-Att fails to extrapolate the expected linear mean velocity scaling OOD. Right, we provide the nMSE for varied contexts at $m = 1$ and on the external AneuG test set. The maximum masking ratio is 99%.

Local geometry editing benefits from reusable global context. Local geometry editing highlights another key advantage of inpainting. When only a small part of the domain changes, most of an existing high-fidelity simulation remains valid. A forward surrogate cannot exploit this information and must predict the full edited field from scratch, whereas our model fixes the unchanged region and only infers the modified part.

We evaluate this using local aneurysm deformations, shown in Figure 5. Source and target geometries differ mainly around the aneurysm. We mask this region, keep the source simulation outside the mask fixed, and inpaint the missing velocities. The quantitative results are shown in Table 1. Inside the edited mask, the supervised neural surrogate obtains an nMSE of $72.97\% \pm 9.81\%$, while L-MAE reduces this to $26.43\% \pm 4.19\%$. L-MAE also substantially improves directional agreement, increasing cosine similarity from $47.30\% \pm 4.48\%$ to $80.49\% \pm 3.60\%$, and reduces both divergence and vorticity error. On the full sample, the benefit is even more pronounced because the unchanged part of the simulation is preserved by construction: full-field nMSE decreases from $35.81\% \pm 2.97\%$

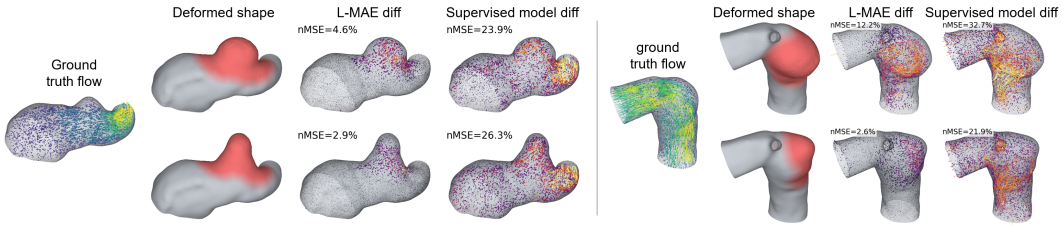

Figure 5: **Inpainting local geometry edits benefits from global context.** We locally deform two example geometries by modelling the growth of an aneurysm. By generously masking the area around it and conditioning our inpainting approach on the original simulation, we achieve superior results compared to neural surrogates that simulate the full geometry from scratch for every edit. We show the difference to the ground truth for the L-MAE and the neural supervised-Att.

Table 1: Quantitative results for small geometry edits. We find that inpainting works especially well for only minor adaptations. We evaluate the masked region and the full simulation.

Method	nMSE ↓ (%)	CosSim ↑ (%)	divergence ↓	vorticity ↓
Inside mask				
Supervised-Att	$72.97\% \pm 9.81\%$	$47.30\% \pm 4.48\%$	0.34 ± 0.01	1.18 ± 0.06
L-FM-Zero	$89.25\% \pm 19.69\%$	$46.63\% \pm 6.03\%$	0.24 ± 0.01	1.22 ± 0.12
L-FM-Physics	$62.73\% \pm 14.56\%$	$57.94\% \pm 5.87\%$	$\mathbf{0.21 \pm 0.01}$	1.09 ± 0.10
L-MAE	$\mathbf{26.43\% \pm 4.19\%}$	$\mathbf{80.49\% \pm 3.60\%}$	0.22 ± 0.01	$\mathbf{0.73 \pm 0.03}$
Full sample				
Supervised-Att	$35.81\% \pm 2.97\%$	$72.49\% \pm 1.16\%$	0.29 ± 0.01	0.97 ± 0.02
L-FM-Zero	$7.88\% \pm 1.57\%$	$88.35\% \pm 1.27\%$	0.20 ± 0.001	0.42 ± 0.03
L-FM-Physics	$7.14\% \pm 1.55\%$	$90.10\% \pm 1.25\%$	$\mathbf{0.19 \pm 0.01}$	0.42 ± 0.03
L-MAE	$\mathbf{4.52\% \pm 0.83\%}$	$\mathbf{94.41\% \pm 0.72\%}$	0.19 ± 0.01	$\mathbf{0.34 \pm 0.02}$

for the forward surrogate to $4.52\% \pm 0.83\%$ for L-MAE. In comparison, the FM models struggle in this task. We hypothesise that L-MAE's superior performance in local editing stems from its training objective, which directly optimises for conditional completion from partial tokens, whereas the global noise integration in latent flow-matching struggles to strictly adhere to narrow, high-frequency spatial boundary constraints. Grouping the edits by volume IoU shows that the inpainting error grows significantly with deformation magnitude for L-FM and L-MAE, giving an empirical estimate of the pipeline's limit, whereas supervised-Att shows no trend since it always predicts from scratch (Appendix D.4).

These results demonstrate a practical advantage of boundary-conditioned inpainting beyond full-field prediction. When local design or anatomical changes are made, *our model can reuse the known global solution and focus its capacity on the region that actually changed.*

5 Discussion

We propose to view steady CFD inference as a context-conditioned inpainting problem rather than as a fully supervised forward prediction task. This changes the role of conditioning: boundary conditions, partial measurements, or unchanged regions from previous simulations are represented directly as observed flow context, rather than as task-specific input variables. Our experiments on intracranial aneurysm hemodynamics show both the promise and the current scope of this formulation. With only sparse inlet-outlet context, inpainting can recover full velocity fields, although a supervised forward surrogate remains highly competitive in the in-distribution setting for which it was explicitly trained. Inpainting is therefore not intended to replace boundary-conditioned surrogates in this classical setting, but to complement them when additional solution context is available, for example from a pre-treatment simulation for virtual interventions, 4D flow MRI or Doppler ultrasound. The advantage of inpainting becomes clearer when additional context is available and under distribution shift. In particular, the latent masked autoencoder is more robust to unseen mass flows and external dataset

shift than the supervised baselines, and local geometry-editing experiments show that inpainting can reuse unchanged regions of an existing simulation instead of predicting the entire field from scratch. These results support the view that self-supervised inpainting can act as a reusable CFD prior, especially when the inference-time conditioning pattern differs from the training-time supervision used by conventional surrogates.

An adjacent, popular approach to neural surrogate modelling is autoregressive solver emulation (e.g., [Lippe et al., 2023]). In contrast, our approach is end-to-end surrogate modelling (compare, e.g., [Alkin et al., 2024]). Both approaches have their advantages and disadvantages: the former sacrifices broadness in scope (e.g., beyond a certain flow case) for close adherence to established solver behaviour, while the latter can be more generalisable.

Broader impact. Inpainting physics could provide fast flow estimates across disciplines from medicine to aerospace, but since the models are not yet flawless, they should complement rather than replace numerical simulation.

Limitations. At the same time, our study is an initial step. We focus on steady flows, mainly velocity fields in internal hemodynamics. Pressure, wall shear stress (WSS) and external aerodynamics are evaluated only in the first experiments in the Appendix. Derived biomedical or engineering metrics beyond WSS are not yet predicted. Extending the formulation to transient flows, via autoregressive rollouts or spatio-temporal inpainting, is left for future work. Moreover, the studied hemodynamic setting is relatively favourable for inpainting because inlet-outlet velocity context provides strong information about the dominant flow structure. We also acknowledge that local context may contain dynamical parameters but may not contain structural parameters that determine the flow or simulation results, such as Reynolds number or choice of turbulence model. More general CFD problems may require additional conditioning, particularly for properties that are difficult to infer from sparse context, such as viscosity or mesh resolution towards the domain boundary (e.g., artery wall). Future work should test how the model handles physically inconsistent context, such as boundary velocities that violate continuity.

Overall, our results suggest that CFD inference need not always be formulated as a task-specific forward surrogate with a fixed set of input parameters. A self-supervised model trained to complete velocity fields can instead be conditioned at inference time on arbitrary known regions of the solution. This inpainting perspective provides a flexible alternative for boundary-condition shifts, partial observations, and local geometry edits, and points toward CFD models that behave less like single-purpose predictors and more like reusable generative priors over physically plausible flow fields, with possible implications for large-scale pre-training and foundation models.

Acknowledgments and Disclosure of Funding

This research was funded by the German Ministry of Research, Technology and Space (BMFTR) within the Software Campus program, funding code: 16IS23069. The responsibility for the content of this publication lies with the authors.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

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

Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. AB-UPT: scaling neural CFD surrogates for high- fidelity automotive aerodynamics simulations via anchored- branched universal physics transformers. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=nwQ8nitlTZ.

Neil Ashton, Charles Mockett, Marian Fuchs, Louis Fliessbach, Hendrik Hetmann, Thilo Knacke, Norbert Schonwald, Vangelis Skaperdas, Grigoris Fotiadis, Astrid Walle, Burkhard Hupertz, and Danielle Maddix. Drivaerml: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics. *arXiv preprint arXiv:2408.11969*, 08 2024. doi: 10.48550/arXiv.2408.11969.

Cristian Bodnar, Wessel Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Jayesh Gupta, Kit Thambiratnam, Alexander Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard Turner, and Paris Perdikaris. A foundation model for the earth system. *Nature*, 641:1180–1187, 05 2025. doi: 10.1038/s41586-025-09005-y.

M. Buzzicotti, F. Bonaccorso, P. Clark Di Leoni, and L. Biferale. Reconstruction of turbulent data with deep generative models for semantic inpainting from TURB-Rot database. *Phys. Rev. Fluids*, 6:050503, May 2021. doi: 10.1103/PhysRevFluids.6.050503. URL https://link.aps.org/doi/10.1103/PhysRevFluids.6.050503.

Wenhao Ding, Yiying Sheng, Simão Nieto de Castro, Hwa Liang Leo, and Choon Hwai Yap. Aneug-flow: A large-scale synthetic dataset of diverse intracranial aneurysm geometries and hemodynamics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

C. Drygala, B. Winhart, F. di Mare, and H. Gottschalk. Generative modeling of turbulence. *Physics of Fluids*, 34(3):035114, 03 2022. ISSN 1070-6631. doi: 10.1063/5.0082562. URL https://doi.org/10.1063/5.0082562.

Pan Du, Meet Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15, 11 2024. doi: 10.1038/s41467-024-54712-1.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9):1305–1315, 1997.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Benjamin Holzschuh, Georg Kohl, Florian Redinger, and Nils Thuerey. P3D: Highly scalable 3d neural surrogates for physics simulations with global context. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8UdCE5nhFl.

Benjamin J. Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-transformer: Efficient and versatile transformers for physics simulations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/holzschuh25a.html.

Xigui Li, Yuanye Zhou, Feiyang Xiao, Xin Guo, Chen Jiang, Tan Pan, Xingmeng Zhang, Cenyu Liu, Zeyun Miao, Jianchao Ge, et al. Aneumo: A large-scale multimodal aneurysm dataset with computational fluid dynamics simulations and deep learning benchmarks. *arXiv preprint arXiv:2505.14717*, 2025.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Marten Lienen, David Lüdke, Jan Hansen-Palmus, and Stephan Günnemann. From zero to turbulence: Generative modeling for 3D flow simulation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ZhlwoC1XaN.

Mario Lino, Tobias Pfaff, and Nils Thuerey. Learning distributions of complex fluid simulations with diffusion graph networks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=uKZdlihDDn.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. Pderefiner: Achieving accurate long rollouts with neural PDE solvers. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 03 2021. doi: 10.1038/s42256-021-00302-5.

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AM7iAh0krx.

Guido Nannini, Simone Saitta, Luca Mariani, Riccardo Maragna, Andrea Baggiano, Saima Mushtaq, Gianluca Pontone, and Alberto Redaelli. An automated and time-efficient framework for simulation of coronary blood flow under steady and pulsatile conditions. *Computer Methods and Programs in Biomedicine*, 257:108415, 2024. ISSN 0169-2607. doi: 10.1016/j.cmpb.2024.108415. URL https://www.sciencedirect.com/science/article/pii/S0169260724004085.

Fabian Paischer, Leo Cotteleer, Yann Dreze, Richard Kurle, Dylan Rubini, Maurits Bleeker, Tobias Kronlachner, and Johannes Brandstetter. Going with the speed of sound: Pushing neural surrogates into highly-turbulent transonic regimes. *arXiv preprint arXiv:2511.21474*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Alberto Pepe, Mattia Montanari, and Joan Lasenby. Fengbo: a clifford neural operator pipeline for 3d PDEs in computational fluid dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VsxbWTDHjh.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.org/10.1109/CVPR52688.2022.01042.

François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, Francois Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=xoNrbfbekM.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for PDE simulations. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/2974844555dc383ea16c5f35833c7a57-Abstract-Conference.html.

Jonas Spinner, Victor Bresó, Pim de Haan, Tilman Plehn, Jesse Thaler, and Johann Brehmer. Lorentz-equivariant geometric algebra transformers for high-energy physics. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/277628cff838927d869cd1f671328ce0-Abstract-Conference.html.

Julian Suk, Baris Imre, and Jelmer M Wolterink. Lab-gatr: geometric algebra transformers for large biomedical surface and volume meshes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 185–195. Springer, 2024.

Luning Sun, Xu Han, Han Gao, Jian-Xun Wang, and Liping Liu. Unifying predictions of deterministic and stochastic physics in mesh-reduced space with sequential flow generative model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/bea78e2bb0abccc14404b24b90d9299f-Abstract-Conference.html.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. *Advances in Neural Information Processing Systems*, 38:160217–160252, 2026.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=HXFvNkNt0n.

Geng Zhou, Yueqi Zhu, Yanling Yin, Ming Su, and Minghua Li. Association of wall shear stress with intracranial aneurysm rupture: systematic review and meta-analysis. *Scientific reports*, 7(1): 5331, 2017.

A Metrics

We validate the velocity field using direct metrics and additional metrics that assess the solution's orientation and physical plausibility. This way, we evaluate not only how well our generative model inpaints the velocity field, but also how strongly the solution adheres to the underlying physics.

Unless otherwise stated, we report the standard error over the test set in plots and tables. For specific implementation details, we provided our code.

Let $\mathbf{x}_i \in \mathbb{R}^3$ be the positions of each point i for a given sample, with predicted velocity $\hat{\mathbf{v}}_i \in \mathbb{R}^3$ and ground-truth velocity $\mathbf{v}_i \in \mathbb{R}^3$.

Normalised MSE (nMSE). We report the sample-wise normalised error:

$$\mathrm{nMSE}(\hat{\mathbf{v}}, \mathbf{v}) \;=\; \frac{\sum_i \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|_2^2}{\sum_i \|\mathbf{v}_i\|_2^2} \tag{5}$$

Cosine similarity. To measure directional agreement independently of magnitude, we report the mean cosine similarity between predicted and ground-truth velocity vectors:

$$\mathrm{CosSim}(\hat{\mathbf{v}}, \mathbf{v}) \;=\; \frac{1}{N} \sum_{i=1}^{N} \frac{\hat{\mathbf{v}}_i \cdot \mathbf{v}_i}{\|\hat{\mathbf{v}}_i\|_2 \, \|\mathbf{v}_i\|_2}. \tag{6}$$

We exclude the lowest 10% of $\mathbf{v}_i$ in each sample.

Normalised divergence. For an incompressible flow, the continuity equation requires $\nabla \cdot \mathbf{v} = 0$. We estimate the velocity Jacobian $J_i \in \mathbb{R}^{3\times 3}$ at each point from its k nearest neighbours by local least squares on the first-order Taylor expansion $\mathbf{v}(\mathbf{x}_j) - \mathbf{v}(\mathbf{x}_i) \approx J_i^\top (\mathbf{x}_j - \mathbf{x}_i)$, and report the divergence residual normalised by the average Jacobian Frobenius norm so that the score is dimensionless and insensitive to the overall flow scale:

$$\mathrm{Div}(\hat{\mathbf{v}}) \;=\; \frac{\frac{1}{N}\sum_{i=1}^{N} |\mathrm{tr}(\hat{J}_i)|}{\frac{1}{N}\sum_{i=1}^{N} \|\hat{J}_i\|_F + \varepsilon}. \tag{7}$$

We use $k = 10$ neighbors and subsample to at most $50{,}000$ query points per sample.

Normalised vorticity RMSE. The vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{v}$ is recovered from the antisymmetric part of the Jacobian, $\omega_i = (\partial_y v_z - \partial_z v_y,\ \partial_z v_x - \partial_x v_z,\ \partial_x v_y - \partial_y v_x)$. We report the RMSE between predicted and ground-truth vorticity fields, normalised by the mean ground-truth vorticity magnitude:

$$\mathrm{Vort}(\hat{\mathbf{v}}, \mathbf{v}) \;=\; \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\hat{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i\|_2^2}}{\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{\omega}_i\|_2 + \varepsilon}. \tag{8}$$

Both fields share the same KNN structure, so the comparison is not biased by differing neighbourhoods.

B Model implementation details

For the main training runs, we use an NVIDIA A100 GPU - 80GB and train each model for two days, selecting the best validation checkpoint. All models are set up to a similar size of 30 million parameters.

L-FM. A per-token DiT [Peebles and Xie, 2023] operates on the $2{,}500 \times 256$ latent tokens produced by the NT. The backbone uses hidden size 512, 8 blocks, 8 attention heads, and an MLP ratio of 4. Token positions are embedded from the ball-centre coordinates; flow-matching time is injected through AdaLN-zero conditioning. We train with the rectified flow-matching objective and a logit-normal time schedule (μ=0, σ=1). Optimisation uses AdamW with learning rate 10^{-4}, weight decay 0.01, gradient clipping at 1.0, batch size 8, and an EMA of 0.9999 (warmup $1{,}000$ steps).

L-MAE. A masked autoencoder is trained over the same NT latent tokens. The encoder and decoder are 4-block transformers with hidden size 512, 8 heads and MLP ratio 4. The decoder receives mask tokens for the held-out positions and is trained to regress the masked latents under MSE. To prevent the trivial position-prior shortcut we observed early on, the mask ratio is sampled per batch from $\mathcal{U}(0.1, 0.9)$. Optimisation uses AdamW with learning rate 4×10^{-4}, weight decay 0.01, gradient clipping at 1.0 and batch size 24.

Supervised-Att. We implement a hierarchical encoder over 5,000 supernodes (constructed from 50,000 subsampled mesh points via k-NN with k=32), a Perceiver-style geometry block, and a cross-attention decoder back to the query points, following [Alkin et al., 2025]. The model has hidden dimension 384, 6 heads, 1 geometry block and a `pscscscscs` encoder pattern with 2 decoder blocks. Mass flow, wall distance and inlet distance are included through Fourier feature conditioning. We train with Lion (β_1=0.9, β_2=0.99, weight decay 0.05), learning rate 2.5×10^{-5}, gradient clipping at 0.25 with 10-epoch warmup and a cosine schedule, mixed-precision, plus per-sample SO(3) rotation and translation augmentation. Predictions on the 50,000-point subset are upsampled to the full mesh via inverse-distance-weighted interpolation before metric evaluation.

Supervised-MP. A 5-level feature-propagation architecture following [Qi et al., 2017b]. Each level downsamples by a factor of 4 (ratio $=$ 0.25) and queries up to 32 neighbours. Feature propagation uses k=4 inverse-distance interpolation. The input is the 50,000-point subsampled mesh and predictions are upsampled the same way as for the Supervised model. We train with AdamW (learning rate 4.24×10^{-4}, weight decay 0.01, gradient clipping at 1.0), batch size 64, cosine schedule with 5-epoch warmup plus per-sample SO(3) rotation and translation augmentation.

C Neighbourhood tokeniser (NT)

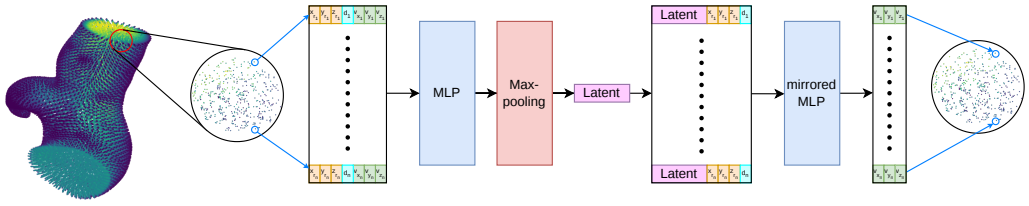

Figure 6: **Overview of the architecture of the neighbourhood tokeniser (NT).** (1) We obtain each neighbourhood and for every point we obtain the corresponding input for the encoder, containing x_r, y_r, z_r as local position, d as distance to the wall and v_x, v_y, v_z for the velocity values. (2) We encode the data through an MLP followed by max-pooling, yielding the latent space for the neighbourhood. (3) We expand by creating copies of the latent space for each point in the neighbourhood and assign via concatenation the information on local position and distance to the wall. (4) Lastly, a similar mirrored MLP architecture predicts the velocity values on each point.

NT is based on a PointNet-style autoencoder [Qi et al., 2017a] presented in Figure 6. It consists of a shared MLP in the encoder and the decoder, and a max pooling layer to reduce the data into a latent space. The MLP of the encoder consists of a concatenation of linear layers with ReLU activation functions and a layer normalisation right at the end of the encoder in order to ensure that the outcome of the MLP is normalised across channels. The MLP of the decoder is also a combination of linear layers and ReLU activation functions, but without a layer norm in comparison to the encoder. To account for the use of ReLUs, the weights of the hidden layers were initialised using He initialisation [He et al., 2015]. The architecture of the designed NT, also performs a concatenation of the local position and distance to the wall after the latent space has been expanded to the same number of points required and before is fed into the decoder. This promotes the NT to focus only into the predictions of the velocity values.

The model is trained with MSE loss, Adam (α=1.6×10^{-4}), batch size 64, and a ReduceLROnPlateau scheduler on training arteries. SO(3) rotation augmentation is applied per artery per epoch, rotating both positions and velocities consistently.

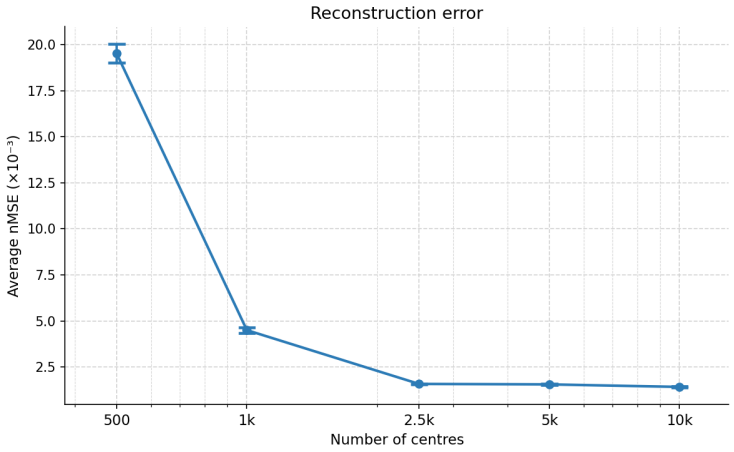

Figure 7: Analysis for nMSE values under different number of centres. 2500 is the best scenario providing the fewest possible number of centres without losing reconstruction capabilities.

The inputs of the autoencoder are the local position normalised, the distance to the wall and the velocity values, a 7-dimensional input. The local position is defined as the relative position to the centre of each of the neighbourhoods, which we normalise so every neighbourhood presents distances from zero (centre) to one (furthest point). The distance to the wall is considered here to be the Euclidean distance. The points in the wall are found by assuming the velocity to be zero near walls.

Moreover, the points that might have not been considered in any ball (large arteries) obtain their prediction based on interpolation from the 8 closest points that have got a prediction.

The number of centres has been evaluated by us. We evaluate the reconstruction quality under varying levels of compression (dictated by the number of centres and therefore local neighbourhoods). We utilize nMSE for the corresponding analysis under 548 arteries that have not been seen during training (the average of them is presented). Each model has been trained on the same arteries and splits for one day.

In Figure 7 we observe that 2500 centers has almost the same nMSE compared to 10000, which allows us to reduce 4 times the number of centres while maintaining almost the same level on reconstruction quality.

The reason why the method establishes a constant number of points per neighbourhood area is for better resource management in terms of GPU. This occurs as by all having same dimensionality between batches, we do not require any masking criteria or alike. The number of points $N = 512$ has been decided based on an Optuna study [Akiba et al., 2019].

Lastly, we also provide here ablation studies that demonstrate the results under the removal of key designs that have led to the final version presented via the NT.

The key decisions under analysis are:

No neighbourhood normalisation: We remove the normalisation of the relative positions to the centre of each ball, leading to the model having to consider different radii on balls as an influence factor.

No data augmentation: We remove the SO(3) rotations on each epoch.

No distance to wall information: We remove the scalar value dictating the distance to the wall as extra information in the input and also from the concatenation before the decoder stage.

No local position and no distance to wall: We remove the concatenation before the decoder part, by not adding any information on the position.

Mean pooling: We test mean pooling instead of max pooling.

We present the results in Table 2 for the listed experiments by using nMSE under the same split as we used for the number of centres and each model being trained for one day. We can observe that the

Table 2: Average nMSE across different neighbourhood tokeniser variants. Final version is the one providing best results.

Variant	**Average nMSE** ($\times 10^{-3}$)
Final version	1.68 ± 0.03
No augmentation	1.74 ± 0.03
No normalisation	4.54 ± 0.07
Mean pooling	3.09 ± 0.05
No wall distance	27.28 ± 0.12
No local pos. in decoder	185.50 ± 0.93

final version provides the best results, as expected. Moreover, we can observe that the model can not reconstruct the data without the local position added before the decoder. In addition, the distance to the wall also represents a large improvement in terms of reconstruction results. The normalisation of balls and usage of max pooling also allow for better results. Finally, we observe that rotating the arteries as augmentation also provides an improvement.

D Additional results

D.1 Additional OOD results

In Figure 8, we show the OOD results for the best methods on AneuG and on an OOD mass flow.

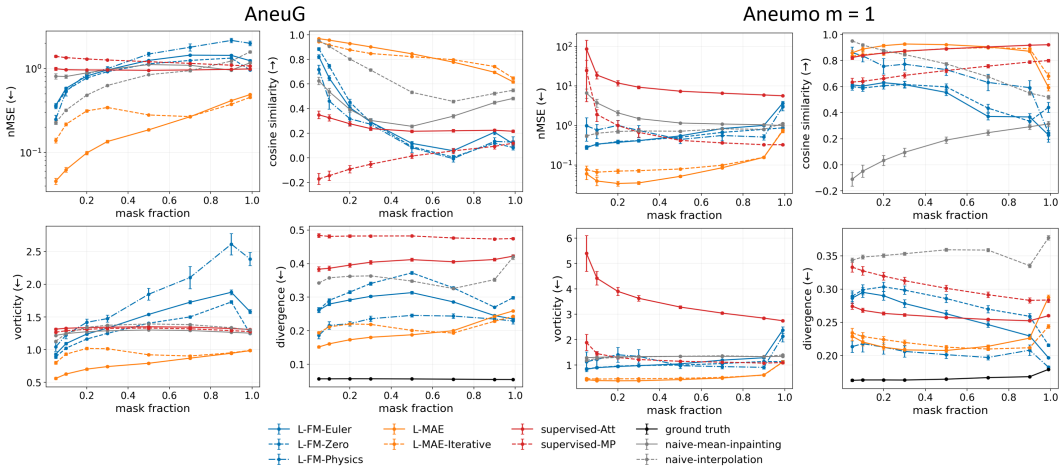

Figure 8: **The supervised models' prediction breaks for the external AneuG dataset and OOD mass flows. L-MAE inpainting is the only method consistently outperforming the naive baselines.** We compare different supervised and self-supervised models on the prediction of velocity fields provided with varying levels of context. Experiments are conducted on the external AneuG [Ding et al., 2025] dataset. We show the respective results for Aneumo in Figure 3.

D.2 Further flow matching integration schemes

In Figure 9, we test additional L-FM integration schemes:

L-FM-Soft: During inpainting, we add soft boundaries by adding noise outside the mask.

L-FM-Iterative: Similar to L-MAE-Iterative, we solve the inpainting problem in steps, starting from the outside and progressing toward the inside of the mask.

L-FM-Init-MAE: We combine L-MAE prediction and refine it with FM integration. We add some noise ($t = 0.7$) to the L-MAE and integrate from there using L-FM.

L-FM-Noise: Fixing the non-masked tokens to their clean latents creates a mixed-time latent state that the model has not seen during training. L-FM-Noise instead applies the correct noise level to all non-masked tokens, $z_t = (1-t)\,\epsilon + t\,z$, with a single fixed noise sample ϵ. Compared to L-FM-Zero, it works better for small mask fractions (50% relative nMSE improvement at mask fraction 10%) and worse for large mask fractions (41% relative nMSE increase at 90%). This can be explained as follows: for small masked regions, a large portion of the latent is at the "wrong" time, whereas for large masked regions, the model mostly sees the correct time.

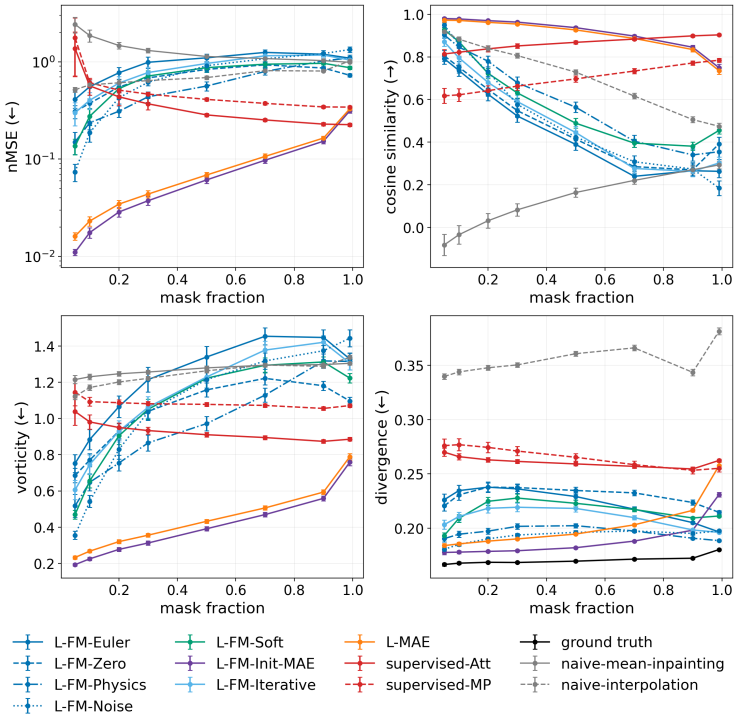

Figure 9: **Combining L-FM with L-MAE improves performance slightly.** We test additional integration schemes for L-FM. Iterative masking or soft boundaries show partial improvement, and noising the known tokens to the current time (L-FM-Noise) helps for small mask fractions, while initialising L-FM with the solution of the L-MAE slightly improves the L-MAE. We evaluate on Aneumo dataset on mass flow 3.

D.3 Including pressure

To showcase the ability to include multiple fields, we add pressure. We trained a separate neighbourhood tokeniser to encode the pressure field. We observed that some non-wall points in the Aneumo dataset had zero pressure and removed these points as artefacts. Using velocity and pressure tokens, 2,500 per field per sample, we train our best method (L-MAE). Using the combined L-MAE, we match velocity accuracy as shown in Figure 10. For the pressure, nMSE drops below 11 % for mask fractions under 0.9, which corresponds to a very sparse context.

With the pressure predictions available, we can compute the residual of the Navier-Stokes momentum equation and compare it to that of the ground-truth simulation. We observe an approximate 5-fold increase in momentum residuals over the ground truth for the sparse case with a 99 % masking fraction. At a 50 % masking fraction, we observe a momentum residual approximately twice as large as the ground truth. Given that this residual is highly sensitive to local errors in both velocity and pressure gradients, these values still indicate good physical consistency, particularly at moderate mask fractions.

Additionally, we evaluate the pressure for mixed boundary conditions, in which we fix the inlet velocity and outlet pressure. This deviation further improves velocity and pressure predictions for small mask fractions, but decreases performance at low context.

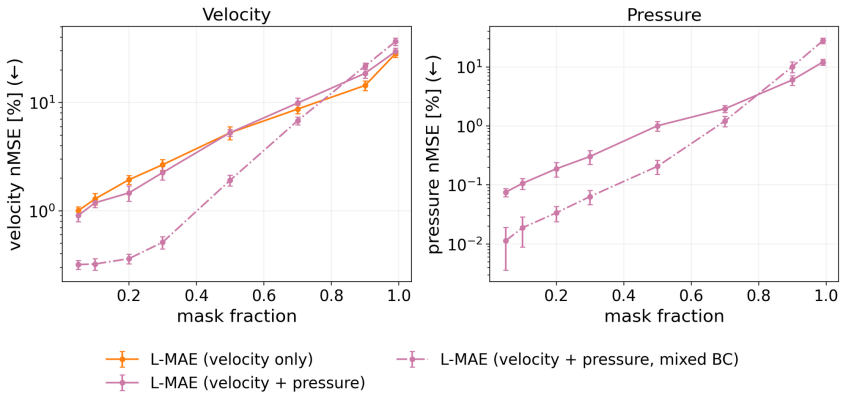

Figure 10: **Adding pressure as a second field preserves velocity accuracy, and mixed boundary conditions improve both fields when context is rich.** We train L-MAE jointly on velocity and pressure tokens and compare it to the velocity-only L-MAE. We show velocity (left) and pressure (right) nMSE over mask fraction for paired masking (the same region is masked in both fields) and mixed boundary conditions (velocity given only at the inlet, pressure only at the outlet). We evaluate on 50 Aneumo test arteries and report the median.

D.4 Detailed analysis of geometric edits

To investigate the relation between the strength of the local geometric edit and performance, we divide the deformation pairs of Table 1 into three equal categories based on their volume intersection-over-union. Table 3 reports the velocity nMSE per category and the Spearman correlation ρ between deformation size and error (positive: larger edits give larger errors). As expected, we find no trend for the supervised case, as the prediction is run from scratch each time. L-FM and L-MAE, on the other hand, show a significant increase in error for larger deformations:

The full-sample numbers quantify the error in the complete domain. We additionally report the error inside the mask and outside the mask in isolation. Inside the mask, the L-MAE error increases with deformation size. Outside the mask, the error purely reflects reusing the source simulation for the edited target geometry. This error is small compared to the inpainting error within the mask. These results indicate that, across the tested deformation range, the dominant error arises within the edited region rather than from reusing the surrounding source simulation.

Table 3: **Inpainting error grows with the size of the geometric edit, mainly inside the edited region.** Velocity nMSE for small, medium, and large deformations, binned by volume IoU, on the full sample, inside the mask, and outside the mask. Mean $\pm$ standard deviation over the pairs in each bin; Spearman ρ between deformation size and nMSE over all pairs.

Method	Small	Medium	Large	Spearman ρ
Full sample				
L-FM-Zero	$4.5\% \pm 4.3\%$	$6.8\% \pm 4.3\%$	$12.7\% \pm 12.7\%$	$+0.51$ ($p = 0.005$)
L-FM-Physics	$4.7\% \pm 5.0\%$	$5.3\% \pm 3.9\%$	$11.5\% \pm 12.8\%$	$+0.40$ ($p = 0.034$)
L-MAE	$\mathbf{1.9\% \pm 1.5\%}$	$\mathbf{4.5\% \pm 3.3\%}$	$\mathbf{7.3\% \pm 6.1\%}$	$+0.59$ ($p = 0.001$)
Supervised-Att	$44.3\% \pm 21.4\%$	$28.6\% \pm 6.5\%$	$30.6\% \pm 8.8\%$	-0.26 ($p = 0.180$)
Inside mask				
L-MAE	$\mathbf{19.7\% \pm 11.1\%}$	$\mathbf{24.7\% \pm 8.3\%}$	$\mathbf{35.8\% \pm 36.7\%}$	$+0.43$ ($p = 0.021$)
Supervised-Att	$78.2\% \pm 33.9\%$	$73.1\% \pm 64.4\%$	$66.1\% \pm 65.9\%$	-0.40 ($p = 0.033$)
Outside mask				
L-MAE	$\mathbf{1.1\% \pm 1.2\%}$	$\mathbf{3.0\% \pm 3.2\%}$	$\mathbf{3.8\% \pm 3.5\%}$	$+0.48$ ($p = 0.009$)
Supervised-Att	$43.1\% \pm 20.5\%$	$26.5\% \pm 6.5\%$	$28.8\% \pm 8.1\%$	-0.32 ($p = 0.094$)

Table 4: **Transfers zero-shot to a T-pipe, while L-FM and the supervised models degrade.** nMSE of the velocity field on an artery-sized T-pipe simulated with OpenFOAM, using the models trained on Aneumo without retraining. The context consists of the inlet and both outlets, and all methods are evaluated on the masked region. Results are for a single geometry.

	Mask fraction			
Method	20% ↓	50% ↓	70% ↓	99% ↓
Supervised-Att	90.6%	71.6%	69.2%	54.4%
Supervised-MP	76.5%	64.4%	59.5%	43.7%
L-FM-Zero	9.1%	42.7%	68.3%	141.2%
L-MAE	**5.8%**	**7.1%**	**8.9%**	**16.2%**

Table 5: **Generalisation to external aerodynamics on Emmi-Wing.** nMSE of the velocity field for models trained on a mid-range band of inflow velocities. The neighbourhood tokeniser, retrained on Emmi-Wing, reconstructs the velocity field with an nMSE of 0.04% in-distribution and at most 0.11% out-of-distribution, far below the inpainting error. We evaluate on wings inside the training band (ID), and on wings with slower and faster inflow (OOD).

Method	ID ↓	OOD-slow ↓	OOD-fast ↓
Supervised-Att	$\mathbf{0.45\% \pm 0.16\%}$	$13.80\% \pm 0.27\%$	$7.69\% \pm 0.32\%$
L-MAE	$1.46\% \pm 0.34\%$	$\mathbf{4.21\% \pm 0.27\%}$	$\mathbf{5.18\% \pm 0.29\%}$

D.5 Generalisation to other flow domains

Internal flow: T-pipe. To evaluate generalisation to an internal flow beyond arteries, we simulate a simple T-pipe geometry, where a pipe is connected orthogonally to the middle of another pipe, using the simpleFoam solver from OpenFOAM to generate a laminar, steady-state flow field. We directly apply the models trained on Aneumo, without any modification or retraining, to study generalisation to a widely out-of-domain problem. As context, we keep the tokens closest to the three openings (inlet and both outlets) and inpaint the rest (Figure 11a). Sweeping over the mask fraction (as in Figure 3), L-MAE outperforms both supervised models by a large margin, whereas L-FM degrades at larger mask fractions (Table 4).

External flow: Emmi-Wing. To demonstrate that our method also generalises to other flow domains, we retrain it on external aerodynamics, the Emmi-Wing dataset [Paischer et al., 2025], a vastly different, turbulent regime from the arterial flows: external flow around wings at inflow velocities of 150–300 m/s instead of internal, low-velocity blood flow. Since the flow orientation is always the same, we removed rotation augmentation. We train our best method, the combination of tokeniser with L-MAE, and the state-of-the-art attention-based supervised model on a mid-range band of inflow velocities. As context, L-MAE only observes the free-stream inflow, given by the 5% most upstream tokens in the far field, and inpaints the remaining 95% of the domain, including the entire near-wing flow, while supervised-Att receives the inflow velocity and angle of attack explicitly (Figure 11b). While performing worse than supervised-Att in the training regime (ID), L-MAE performs better out-of-domain (OOD) for velocities both slower and faster than the training velocities, confirming the superior generalisability of our method (Table 5). This mirrors our findings for unseen mass flows on Aneumo (Figure 4): a self-supervised flow prior conditioned on observed context extrapolates more robustly than a forward surrogate tied to its training-time conditioning range.

D.6 Wall shear stress

The risk of intracranial aneurysm rupture has been associated with wall shear stress (WSS) [Zhou et al., 2017], making it an important downstream biomarker. We estimate WSS based on the predicted velocity field via the near-wall velocity gradient. L-MAE yields the lowest WSS error across all mask fractions (Table 6), whereas L-FM and the supervised baselines perform considerably worse, indicating that L-MAE better preserves the near-wall velocity gradients required for WSS estimation.

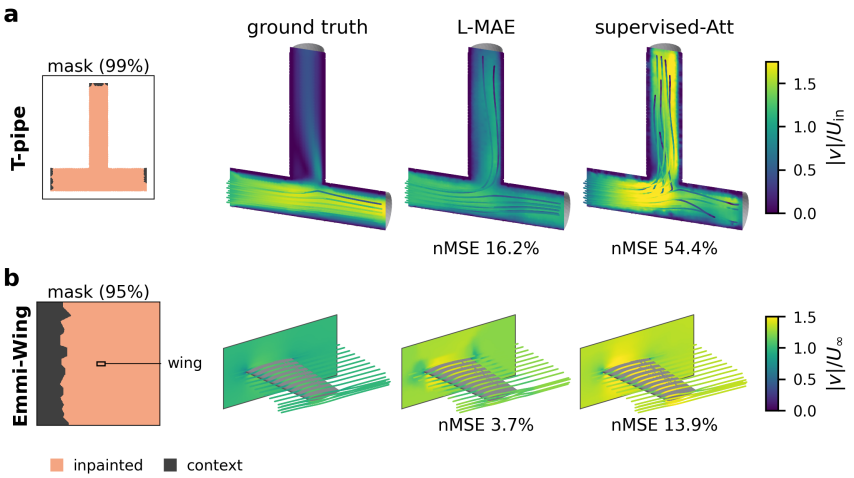

Figure 11: **L-MAE generalises to unseen flow domains and boundary conditions.** (a) Zero-shot T-pipe: models trained on Aneumo only, applied without retraining. Only the three openings are given as context (mask fraction 99%). (b) Emmi-Wing at an OOD-slow inflow (153 m/s): only the upstream free-stream is given as context (mask fraction 95%). In both cases, L-MAE recovers the flow structure, whereas supervised-Att overestimates the velocity.

Table 6: **L-MAE best preserves the near-wall velocity gradients required for WSS estimation.** WSS nMSE derived from the predicted velocity field over mask fraction for supervised and self-supervised models. We evaluated 50 Aneumo test arteries.

Method	20%	50%	70%	90%	99%
Supervised-Att	213.4±34.0	149.3±14.6	129.3±14.2	121.9±14.4	83.9±7.3
Supervised-MP	125.8±28.0	72.6±6.4	66.6±6.3	66.3±6.7	51.2±3.7
L-FM	62.3±13.3	80.5±36.5	57.7±9.2	49.8±2.4	50.5±2.3
L-MAE	**9.3±0.8**	**12.6±0.8**	**14.9±0.8**	**21.0±1.1**	**32.0±1.9**

D.7 Dataset licenses.

We use Aneumo [Li et al., 2025] (CC BY-NC-ND 4.0), AneuG-Flow [Ding et al., 2025] (CC BY-SA 4.0) and Emmi-Wing [Paischer et al., 2025] (CC BY-NC 4.0) in accordance with their licenses, and generated the T-pipe reference simulation with OpenFOAM (GPL-3.0).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims reflect the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:Yes, we discuss limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We have no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we provide the code and description in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide all relevant details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, we report errors.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we follow the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discuss potential societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: That the paper poses no such risks

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, they are mentioned.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, our assets are well documented and provided.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.