RESEARCH ARTICLE | MARCH 17 2026

Shift and flip invariant CNNs for predicting laminar flow properties

Yuri Koide ; Jonas Teufel ; Luca Torresi ; Arjun J. Kaithakkal ; Alexander Stroh ; Pascal Friederich

APL Mach. Learn. 4, 016109 (2026)
https://doi.org/10.1063/5.0317297

Shift and flip invariant CNNs for predicting laminar flow properties

Cite as: APL Mach. Learn. 4, 016109 (2026); doi: 10.1063/5.0317297
Submitted: 17 December 2025 • Accepted: 26 February 2026 •
Published Online: 17 March 2026

Yuri Koide,[1,2] Jonas Teufel,[1,2] Luca Torresi,[1,2] Arjun J. Kaithakkal,[3] Alexander Stroh,[3] and Pascal Friederich[1,2,a)]

AFFILIATIONS

[1] Institute of Anthropomatics and Robotics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany
[2] Institute of Nanotechnology, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany
[3] Institute of Fluid Mechanics, Karlsruhe Institute of Technology, Kaiserstr. 10, 76131 Karlsruhe, Germany

[a)] **Author to whom correspondence should be addressed:** pascal.friederich@kit.edu

ABSTRACT

The integration of machine learning into fluid dynamics has accelerated in recent years, driven by the proliferation of high-fidelity data and enhanced computational resources. Acting as efficient surrogate models for computationally intensive simulations, these data-driven approaches provide substantial benefits, particularly during the preliminary stages of design and optimization. Previous investigations have employed convolutional neural networks (CNNs) to predict thermo-fluid flow properties for a variety of channel geometries. These studies have largely relied on data augmentation techniques to handle geometric transformations. However, such augmentation strategies are often inefficient in capturing the inherent flip and shift invariances of flow channel data. In this study, we demonstrate that embedding these invariances directly into the model architecture not only enhances robustness but leads to superior performance while significantly reducing the number of parameters compared to their invariant-unaware counterparts. In particular, we introduce two novel architectures designed to alleviate the sensitivity of CNNs to periodic signal shifts and vertical flips. This approach allows the model to structurally address the geometric symmetries of the flow channel data, offering a more robust alternative to standard data augmentation.

I. INTRODUCTION

Machine learning has emerged as a pivotal tool across the spectrum of scientific and engineering disciplines. Its capacity to analyze extensive datasets, discern intricate patterns, and generate predictive models has fundamentally transformed the methodologies employed in addressing complex challenges.[1] Recent studies have increasingly integrated machine learning into fluid mechanics to enhance control, monitoring, and the development of reduced-order models.[2–4]

Advancements in numerical simulation and experimental techniques have enabled the detailed measurement of complex flow phenomena. Consequently, the volume of high-fidelity data obtained from these sources has increased significantly, fostering notable progress in the application of machine learning within fluid mechanics.[5] This data abundance is particularly critical given the limitations of traditional methods and while large-scale computational fluid dynamics (CFD) simulations provide high spatial and temporal resolution, their computational cost remains high for iterative tasks.

To address this computational bottleneck, particularly for flows with reduced complexity such as laminar flows, machine learning is increasingly applied as a surrogate model to alleviate the computational burden. This approach is highly advantageous in the early stages of design and development, where rapid iteration is essential.[6–9]

In a previous study, Koide *et al.*[10] developed a surrogate regression model based on convolutional neural networks (CNNs) to predict the thermo-fluid properties—friction coefficient (C_f) and Stanton number (St)—of laminar channel flows with arbitrary wall structuring. Trained on high-fidelity numerical simulations of 2D arbitrary, flat, and non-flat channels, the CNN model was then employed to identify wall modifications that enhance (St) for a given C_f. Their findings demonstrated that extending the training dataset

via data augmentation, by incorporating cyclic horizontal shifts and vertical flips, improved prediction accuracy compared to standard CNN models trained without such augmentation.

However, relying on data augmentation to enforce physical invariances presents intrinsic limitations. While simple transformations such as inversion and shifting can improve performance, they compel the model to expend capacity learning symmetries that are already theoretically known. This approach is computationally redundant, as it treats physical laws as data patterns to be approximated rather than constraints to be satisfied. Furthermore, standard data augmentation is typically discrete and finite, making it often unable to cover the continuous spectrum of possible shifts, rendering the invariance incomplete.[11,12] Consequently, there is a need for architectures that inherently handle these geometric variabilities without the computational overhead and lack of guarantees associated with external augmentation.

In this work, inspired by recent breakthroughs in encoding physical invariances within equivariant graph neural networks,[13,14] we address these challenges by embedding invariance properties into the model architectures. This approach enables the network to learn feature representations that are inherently invariant to shifts and flips, improving prediction robustness while significantly reducing the number of trainable parameters.

In particular, we propose two CNN-based architectures: the sum-mirror CNN and the FFT-CNN. Both are designed to strictly enforce invariance to periodic signal shifts and vertical reflections, ensuring predictions remain consistent across geometrically equivalent channel configurations. By structurally addressing the geometric symmetries of the flow channel data, these models offer a more robust and efficient alternative to standard data augmentation. Our evaluation demonstrates that this architectural enforcement leads to superior predictive performance and stability compared to invariance-unaware baselines.

II. RELATED WORK

A. Machine learning in CFD

Beyond standard machine learning applications, rapid advances in modern deep learning methods are increasingly employed to address challenges in CFD.[4] When trained in a supervised manner, for example, ML models can estimate stationary fluid velocity fields orders of magnitude faster than conventional CFD solvers.[15–17] In parallel, Kim *et al.* develop a generative model for fluid dynamics simulations.[18] Their autoencoder structure facilitates latent space reconstructions, frame interpolations, and time extrapolations of time-dependent simulations. Furthermore, Fukami *et al.*[7] demonstrate how CNN models can be adapted for super-resolution reconstruction of laminar and turbulent flow fields. This approach offers the potential to significantly improve insight derived from low-resolution experimental data. Recent studies have further expanded these capabilities. Sofos and Drikakis[19] provide a comprehensive review of deep learning for super-resolution, highlighting the integration of physics-informed constraints. Moreover, for multidimensional turbulent flows, Drikakis *et al.*[20] demonstrated that transformer-based architectures can offer superior generalizability compared to traditional methods. However, while these complex architectures address the high-dimensional fluctuations of turbulence, structurally invariant CNNs remain a highly efficient and robust approach for steady laminar flows dominated by geometric boundary conditions. Distinct from surrogate estimators of full flow fields, machine learning models can also be utilized to directly predict global flow quantities, such as C_f and St. Viquerat and Hachem,[21] for instance, trained a CNN model to predict C_f values of various 2D geometries subjected to an external flow. Finally, Koide *et al.*[10] used a CNN to predict C_f and St in laminar channel flows with arbitrary wall structuring. In related work, this type of efficient surrogate model was coupled with an evolutionary algorithm to optimize wall geometries for enhanced heat transfer.[22]

B. Invariance in CNNs

Functions are called *invariant* toward a transformation if they maintain the same output when that transformation is applied to the input of the function. Explicit consideration of such invariances in the design of neural networks is a key method to include task-specific inductive biases. Consequently, prior work aims to introduce such invariances to various classes of neural network models, including CNNs used in image processing.

One common method to ensure invariance to certain transformations is enforcing it through specific modifications of the network architecture.[23,24] Chaman and Dokmanic,[25] for example, introduces adaptive polyphase sampling (APS) to address CNN's inherent shift-variance related to image subsampling (stride) operations. Cohen and Welling[26] introduces the concept of group equivariant convolutional networks (G-CNNs) as a generalization of standard convolutional operations to capture invariances regarding additional group transformations. Weiler and Cesa[27] extend the notion of G-CNNs and proposes general E(2)-equivariant convolutions, experimentally demonstrating improved invariance to transformations of the Euclidean group.

In addition to explicit architectural modifications, data augmentations are often used to improve transformation invariance. In this regard, Benton *et al.*[28] propose Augerino—a framework for learning transformation invariance through data augmentations. While not intrinsically invariant to transformations, this method can adaptively learn *soft* invariances suited to the training data distribution. More recently, Gerken and Kessel[29] discuss the emergence of invariance and equivariance in deep neural network ensembles as a result of data augmentations. The authors prove complete invariance under idealized conditions and show experimental evidence for converging invariance in realistic settings.

While the aforementioned methods primarily address geometric symmetries (e.g., rotation and translation), the principle of enforcing physical invariances might be extended to other non-geometric properties. Recently, Fukami *et al.*[30] demonstrated that enforcing a fundamental property of scale invariance in isotropic turbulent flows can significantly enhance model robustness. By utilizing Buckingham Pi analysis to derive nonlinear scaling laws for vortical structures, they enabled machine learning models to generalize across varying Reynolds numbers. Although our study focuses on the geometric shift and flip invariances inherent to the considered laminar channel flows, both approaches share the fundamental philosophy of embedding physical symmetries directly into the modeling pipeline to reduce the search space and improve generalization.

III. TASK DESCRIPTION AND THE PROPOSED APPROACH

A. Task description

In this work, we consider the prediction of thermo-fluid flow properties for two-dimensional laminar channel flows (see Fig. 1). In particular, we predict C_f and St, both of which are used in the calculation of the Reynolds analogy factor,

$$RA = \frac{2St}{C_f}. \tag{1}$$

RA is used to evaluate the similarity between momentum and heat transfer. Therefore, increases in RA indicate a larger increase in heat transfer relative to the increase in momentum transfer—a desirable property for the design of energy-efficient systems.[31] The properties C_f and St depend on the exact geometry of the channel walls resulting from the imposed boundary conditions on the fluidic system. Due to the choice of the boundary conditions for the velocity and temperature fields, the wall-normal heat flux remains constant once thermal equilibrium is reached. Furthermore, the given field properties are inherently invariant to horizontal cyclic shifts and vertical flips of the geometry. For a given channel geometry, C_f and St can be obtained from computationally expensive fluid dynamics calculations. To directly estimate the desired properties from a surrogate model, not relying on numerical simulation, we represent channel geometries as bitmap images $\mathbf{x} \in \{0,1\}^{H\times W}$ and, consequently, train a CNN,

$$f_\theta : \{0,1\}^{H\times W} \to \mathbb{R}^2, \quad \mathbf{x} \mapsto C_f, St, \tag{2}$$

with learnable model weights θ.

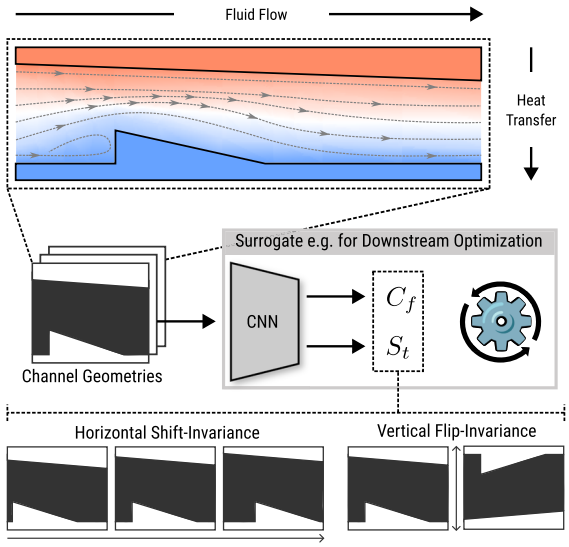

FIG. 1. Task overview. Direct surrogate prediction of thermo-fluid properties (C_f and St) from 2D laminar flow channel geometries created by CFD. Properties are invariant toward cyclic horizontal shifts and vertical flips. Therefore, the surrogate model is expected to reflect the same invariances.

B. Summation and mirroring-based invariant CNN (approach 1)

This is an architecture explicitly designed to embed the physical symmetries of laminar channel flow directly into the deep learning model. In this domain, flow properties such as C_f and St are invariant to translation along the flow direction (x) and reflection across the span (y). To enforce these constraints, our model departs from standard CNN designs by integrating two specialized mechanisms: a dual-path pipeline for flip invariance and a modified convolutional block for shift invariance.

1. Architectural overview and shift invariance

The core feature extractor consists of five serial convolutional blocks. Each block is engineered to process boundary-dominated features without introducing spatial bias. Standard convolutional setups typically rely on zero-padding and max pooling. However, zero-padding creates a foveation effect that degrades performance at image boundaries,[32] while standard pooling is sensitive to small spatial shifts.[12,33] Although anti-aliasing techniques can offer partial robustness, they are limited by non-linear activations and often fail to generalize to unseen patterns.[11]

To resolve these issues and achieve strict shift invariance, we redesign the convolutional block with three specific interventions.

Periodic padding: We replace zero-padding with periodic padding ($p = 4$). As detailed in Sec. III B 3, this wraps pixel information from opposing borders, ensuring convolution kernels operate seamlessly across boundaries.

Adaptive polyphase sampling (APS): Following convolution (8×8 kernel, 1 stride) and batch normalization, we employ APS.[34] Unlike standard pooling, APS dynamically selects the sampling grid based on the highest-energy component, preserving shift robustness before downsampling.

Horizontal aggregation: Crucially, we insert a summation operation immediately after the APS layer. We aggregate the feature representations by summing them horizontally along the image width. This aggregation significantly reduces the parameter count and ensures the feature representation remains invariant to translations along the flow direction.[35]

The number of output channels increases progressively from 16 to 256 across the five blocks. By combining APS with horizontal aggregation, the encoder produces feature maps that are robust to translation along the flow direction.

2. Flip invariance via embedding summation

While the convolutional blocks with APS handle shift invariance, flip invariance is achieved by aggregating the latent representations of both the original and vertically flipped inputs. The network processes the original input image $\mathbf{x}'$ and a vertically flipped version $\mathbf{x}^\dagger$ through identical copies of the encoder described in Fig. 2. This yields two distinct latent embeddings, $\mathbf{z}$ and $\mathbf{z}^\dagger$.

To enforce symmetry, we define the final latent representation $\tilde{\mathbf{z}}$ as the element-wise sum of these embeddings,

$$\tilde{\mathbf{z}} = \mathbf{z} + \mathbf{z}^\dagger. \tag{3}$$

Because vector addition is commutative, $\tilde{\mathbf{z}}$ remains identical regardless of the input's initial vertical orientation. This summed

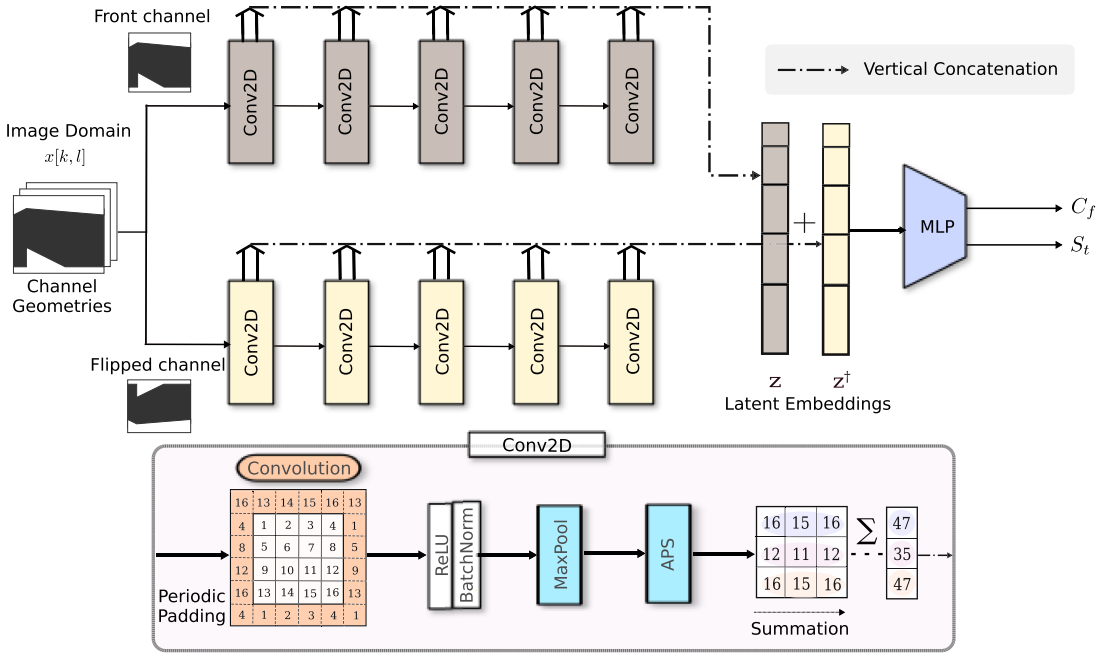

FIG. 2. Schematic of the summation and mirroring-based CNN architecture. Input flow channel geometries are processed via two parallel branches: the original orientation (upper branch) and a mirrored version (lower branch), effectively enforcing flip invariance. Following adaptive polyphase sampling, feature maps are summed along the *x* direction, while the resulting representations from each stage are concatenated along the *y* direction to ensure shift invariance. The dual hidden representations are finally merged and passed to a fully connected layer.

embedding is flattened into a 5616-neuron vector and passed to a regression head comprising three fully connected layers (256, 64, and 2 units) with batch normalization and ReLU, finally predicting the scalar values C_f and S_t.

3. Periodic padding implementation

Given that channel wall structures are located at the image boundaries, preserving edge information is critical. We adopt the circular padding strategy argued by Alsallakh *et al.*,[36] which mitigates boundary artifacts by simulating an infinite, wrapping domain. The effectiveness of periodic padding in preserving physical consistency has been demonstrated by Morimoto *et al.*[37] We leverage this insight to enhance feature continuity across domain boundaries, ensuring edge information is processed as seamlessly as the internal regions.

Formally, we construct the padding regions by slicing the opposite ends of the input tensor $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$. The padding for the top and bottom boundaries is defined as

$$\mathbf{P}_{\text{top}} = \mathbf{x}_{H-p:,:,:} \qquad \mathbf{P}_{\text{bottom}} = \mathbf{x}_{p,:,:,:}. \tag{4}$$

Similarly, for the left and right boundaries,

$$\mathbf{P}_{\text{left}} = \mathbf{x}_{:,W-p:,:} \qquad \mathbf{P}_{\text{right}} = \mathbf{x}_{:,:p,:}. \tag{5}$$

These slices are concatenated to the corresponding sides of $\mathbf{x}$. The corner regions are filled diagonally to complete the padding. The resulting input $\mathbf{x}' \in \mathbb{R}^{(H+2p)\times(W+2p)\times C}$ ensures convolution kernels near the boundaries process features from the opposing physical side, maintaining strict shift invariance. In our experiments, we set $p = 4$.

C. Fourier model (approach 2)

Our second proposed model uses the same mechanism to enforce flip-invariance while taking an alternative approach to reduce the network's sensitivity to input shifts (see Fig. 3). Instead of modifying standard convolutional and pooling layers to incorporate shift invariance, we perform feature extraction in the spectral domain. By transforming input images using the discrete Fourier transform (DFT), we eliminate boundary condition issues, which are inherently handled by the periodic assumption of the DFT.

We base our fast Fourier transform CNN (FTT-CNN) on previous work by Rippel *et al.*[38] We define the convolutional kernels as mappings of the standard filter representations from the spatial domain to the spectral domain, thus keeping the underlying model unaltered. Rippel *et al.*[38] argue that the spectral domain is more suitable for optimizing filter parameters and experimentally demonstrate a faster convergence in training. Moving the pooling layer to the spectral domain offers two main advantages. First, it allows the operation to be reimplemented as a low-pass filter, which, unlike standard max-pooling, does not require significant dimensionality reductions (standard pooling typically reduces dimensionality by a factor of at least 4) and only minimally decreases the information content of the signal, effectively serving as a type of denoising. Second, spectral pooling enhances shift invariance by

avoiding issues related to discretized sampling and stride-related artifacts.[12]

While different methods to replicate the functionality of spatial domain non-linear activation functions in the Fourier-transformed space have been proposed in the literature,[39,40] we opted to retain the ReLU and Batchnorm layers in the spatial domain, as their transfer to the spectral domain has negligible impact on the CNN's modeling capabilities. However, there exists a constraint requiring multiple computationally expensive domain switchings to determine computational overhead. Finally, to ensure the fully connected layers are invariant to shifts in the extracted spectral features, we only take the magnitude of the activations of the last convolutional layer, discarding the phase, and apply a multi-layer perceptron directly in the spectral domain.

1. Architecture Implementation

The overall architecture mirrors that of our first model, consisting of five convolutional blocks, followed by four fully connected layers. Filters are initialized in the spatial domain with kernel sizes of 9×9, 7×7, 5×5, 3×3, and 3×3, respectively, and are subsequently transformed to the spectral domain via DFT. The spectral pooling layers operate as low-pass filters with bandwidths of 0.8, 0.5, 0.3, 0.3, and 0.5. The number of filters scales from 64 in the first block to 512 in the final block. Following the convolutional blocks, the fully connected layers reduce the feature dimensions from 512 to 2. As ReLU activation and batch normalization are applied in the spatial domain, an inverse fast Fourier transform (IFFT) is required after each spectral operation.

IV. EXPERIMENTS

A. Laminar channel flow dataset

We utilize the dataset of structured channel geometries originally introduced by Koide *et al.*[10] These geometries are represented as binary masks, where values of 1 and 0 distinguish solid walls from the fluid domain, respectively. The structures were generated via a random walk algorithm combined with spline interpolation and discretization onto the simulation grid. By varying the number of supporting points n sampled from normal distributions, the dataset encompasses a diverse range of wall configurations with substantial structural variability.

The corresponding thermo-fluid properties were derived from direct numerical simulations (DNS) performed under a constant flow rate. The flow is governed by the incompressible Navier–Stokes and continuity equations, while heat transfer is modeled via the scalar transport equation. Boundary conditions are defined as follows: periodic boundary conditions are applied in the streamwise x-direction for both velocity and temperature fields; no-slip conditions are enforced at the top and bottom walls. For the thermal field, constant temperature boundary conditions are applied, maintaining a fixed temperature difference between the heated top wall and the cooled bottom wall. For a comprehensive description of the numerical solver and generation parameters, we refer the reader to Ref. 10.

The raw dataset initially comprised 10 800 geometries. During preprocessing, samples exhibiting topological anomalies, such as disconnected islands or irregular artifacts, were filtered out to ensure physical validity. This process yielded a final curated dataset of 9121 geometries, covering configurations ranging from flat to complex structured channels. Each sample is represented as a binary image of 129×384 pixels (height $\times$ width), corresponding to the 2D cross section of the channel. For model evaluation, the entire dataset was randomly shuffled and partitioned, with 90% allocated for training and the remaining 10% reserved for testing.

B. Experimental setup

The proposed models were implemented in PyTorch[41] and trained using the Adam optimizer.[42] We employed an initial learning rate of 1×10^{-3}, which was systematically decayed during training to reach a final learning rate of 4.52×10^{-6}. Training was conducted with a batch size of 74 for 150 epochs. All computations were executed on a high-performance computing (HPC) infrastructure utilizing a NVIDIA A100 graphics processing unit (GPU). Further implementation details and hyperparameter settings are provided in the Appendix.

C. Baseline models (standard approach)

To establish a benchmark for performance evaluation, we adopted the standard CNN architectures and data augmentation framework described in the previous study.[10] These reference models were implemented using TensorFlow[43] and Keras.[44] To ensure a rigorous comparison, we strictly adhered to the hyperparameter settings and training protocols outlined in the original experimental method.

V. RESULTS AND DISCUSSION

A. Performance comparison on test data

Our benchmark demonstrates that our invariant models achieved higher R^2 scores, with significantly lower standard deviations across fivefold cross-validation compared to the standard approach models (see Table I). This reduction in variance indicates that the proposed invariant architectures generalize effectively across diverse flow channel patterns, ensuring consistent predictive performance.

In contrast, the variability observed in standard CNNs, even with data augmentation, highlights a lack of invariance. These models remain sensitive to the training distribution, often struggling to distinguish relevant physical features from spatial biases. The architecture is forced to allocate capacity to learning spatial transformations from the data, rather than possessing these invariances by design.

Notably, the R^2 scores for the Stanton number (St) showed a marked improvement in both invariant models, particularly the sum–mirror CNN. Furthermore, these superior results were achieved with substantially fewer trainable parameters. By hard-coding symmetries, our models relieve the architecture of the burden of learning spatial invariance implicitly. This allows the trainable parameters to focus exclusively on the complex non-linear correlations between wall geometries and thermo-fluid properties, rather than consuming resources on learning simple spatial translations.

To assess robustness in the low-data limit, we analyzed the impact of training set size (see Fig. 4), which highlights the enhanced

TABLE I. Model accuracy and number of parameters for predicting C_f and St for different models. The results for a fivefold cross-validation. The best results are highlighted in bold and the second-best results are underlined.

	C_f		St		
CNN model	$r^2 \uparrow$	MAE $\downarrow$	$r^2 \uparrow$	MAE[a] $\downarrow$	No. of Params
Standard	0.918 (±0.036)	0.0020	0.876 (±0.084)	0.1	49.3M
Standard+data augment	0.924 (±0.039)	0.0019	0.853 (±0.082)	0.1	49.3M
FFT-CNN (*ours*)	**0.952** (±0.014)	**0.0018**	<u>0.896</u> (±0.037)	0.1	5.2M
Sum-mirror CNN (*ours*)	<u>0.945</u> (±0.005)	0.0021	**0.923** (±0.008)	0.1	4.3M

[a]Multiplied by a factor of 1000.

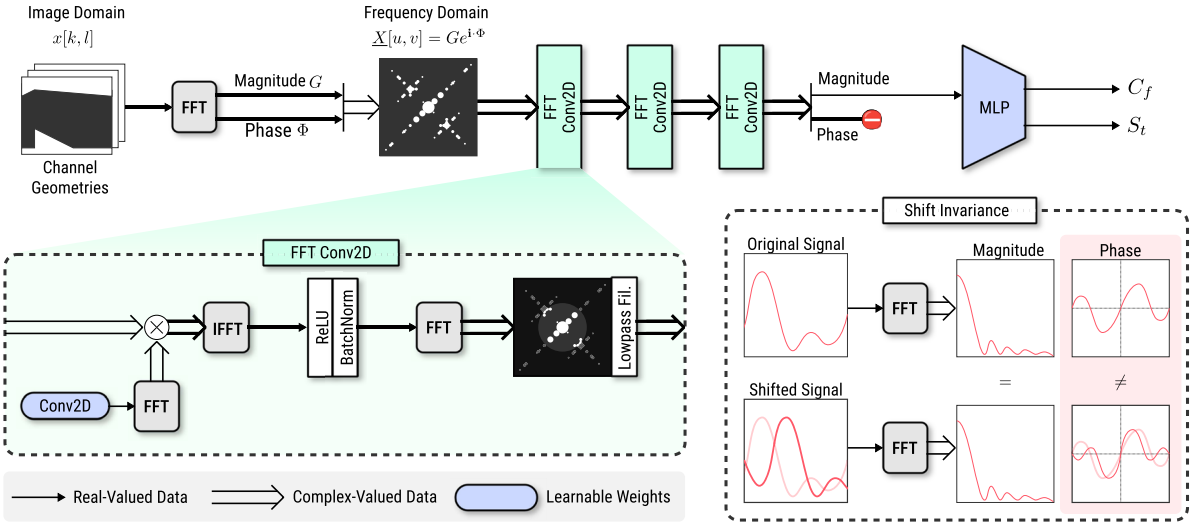

FIG. 3. Overview of the Fourier model architecture. Image representations of the flow channel geometries are Fourier-transformed into the complex frequency domain. The frequency domain representation is subsequently transformed by a series of convolutional filters. Shift invariance is achieved by using only the magnitude of the final spectrum as the input to a fully connected prediction network.

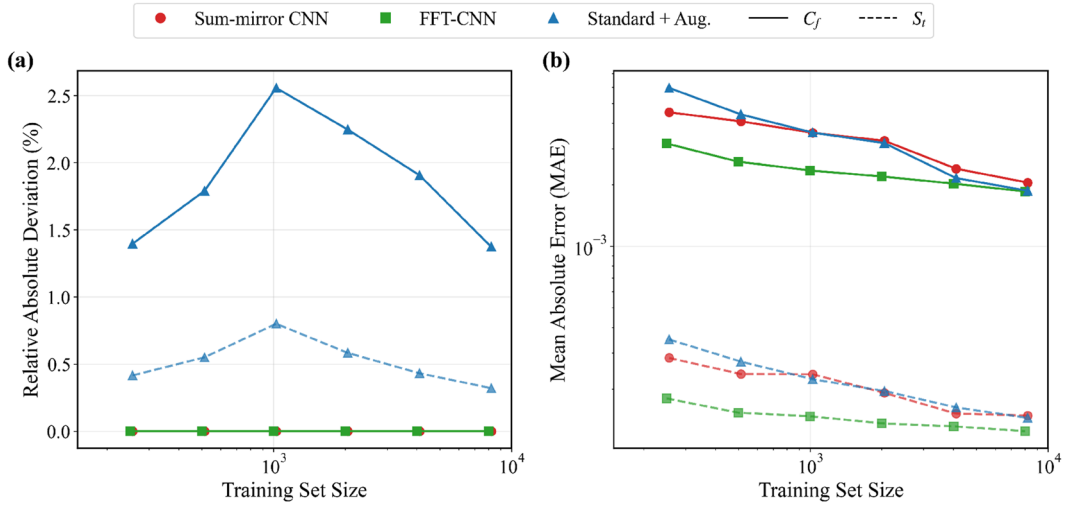

FIG. 4. Dependence of model performance on training set size. (a) Relative absolute deviations $\langle |p_i' - p_i| \rangle_i / \langle p \rangle_i$ ($\langle \cdot \rangle_i$ denotes the mean over samples i) for shifting and (b) mean absolute error (MAE). The results compare the proposed methods with the baseline architecture.

data efficiency and invariance of our approach. Both proposed models achieve lower MAE than the baseline, particularly in the low-data limit. This advantage arises because our architectures maintain negligible shift invariance error regardless of data size, while the baseline relies on large sample sizes to minimize this error. Thus, incorporating symmetries as inductive biases enables effective generalization with limited data, decoupling physical invariance from statistical learning.

B. Quantifying robustness to shift and flip

We assessed model robustness by measuring deviations within a randomly sampled 10% of the full dataset (see Table II). The results confirm that the sum–mirror CNN maintains absolute zero deviation (0%) for both flipping and shifting operations. Similarly, the FFT-CNN achieves strict invariance to flipping and demonstrates almost perfect robustness against shifting. The infinitesimally small deviations observed during shifting operations are practically negligible, stemming primarily from the limits of numerical precision inherent to discrete transformations rather than any architectural vulnerability. This result ensures invariance is hard-coded into the structure, rather than just statistically estimated through augmentation. While standard deep learning models rely on the probabilistic assumption that the training set encompasses all necessary variations, our architecture enforces invariance as a deterministic mathematical constraint. The rigorous invariance of the sum–mirror CNN corroborates the stability observed in the cross-validation results. This implies that the model's high accuracy is not merely a result of memorizing patterns, but of capturing the true, translation-invariant physics of the flow. Our findings align with those of Zhang[12] and Chaman and Dokmanic,[34] confirming that structural invariance provides a superior inductive bias compared to learned invariance.

C. Distribution of prediction deviations

To visualize the robustness of the models against spatial variations, we evaluated test images and their shift-augmented versions over ten iterations (see Fig. 5). In the case of C_f, the deviation distribution of the sum–mirror CNN is concentrated exclusively at 0%. This confirms that the model's architectural constraints successfully enforce exact invariance, effectively decoupling the prediction from the object's position. Similarly, the FFT-CNN exhibits a tight distribution infinitesimally close to zero. As previously established, this minute variance is a numerical artifact of discrete transformations rather than a lack of mathematical invariance, confirming that its frequency domain representation intrinsically suppresses spatial dependencies. In contrast, standard models exhibit a broad, heavy-tailed distribution for both C_f and St. This spread reveals a critical vulnerability that standard CNNs tend to overfit to absolute

TABLE II. Relative absolute deviations $\langle |p_i' - p_i| \rangle_i / \langle p \rangle_i$ ($\langle \cdot \rangle_i$ denotes the mean over samples i) in C_f and St between the single-model predictions of regular inputs and shifted/flipped inputs. The median and confidence limits within a 95% confidence interval are shown.

	Flipping				Shifting			
	C_f		St		C_f		St	
CNN model	95th percentile (%)	Median (%)	95th percentile (%)	Median (%)	95th percentile (%)	Median (%)	95th percentile (%)	Median (%)
Standard	16.737	2.650	16.733	2.650	14.788	1.667	0.791	0.088
Standard+data augment	9.633	1.664	9.633	1.664	8.525	1.180	0.499	0.061
FFT-CNN (*ours*)	0.000	0.000	0.000	0.000	0.417	0.029	0.075	0.005
Sum-mirror CNN (*ours*)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

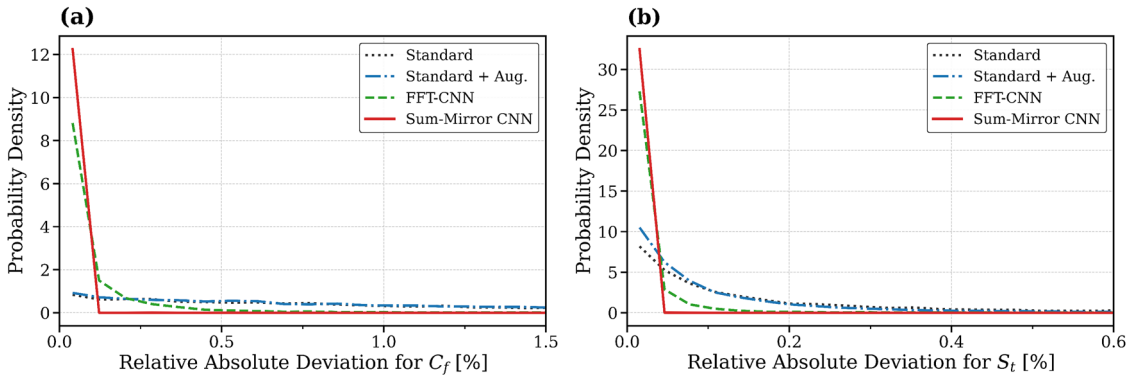

FIG. 5. Probability density of the relative absolute deviations ($|p_i' - p_i|/\langle p \rangle_i$ [%]) on the shifted test set. Panels (a) and (b) show the distributions for the friction coefficient C_f and the Stanton number S_t, respectively, comparing the proposed method against baseline architectures.

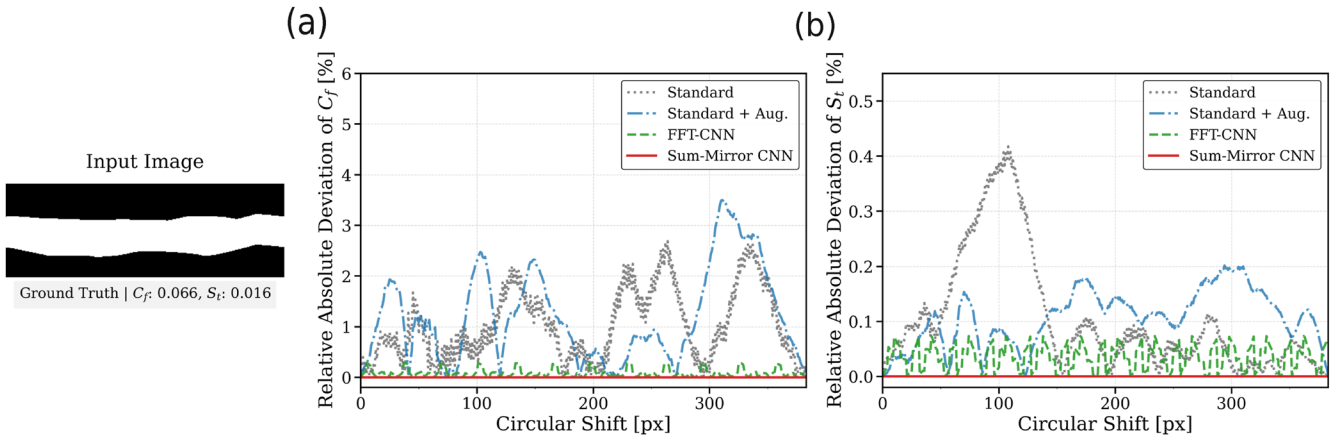

FIG. 6. Comparison of robustness to pixel-wise shifts. The relative absolute deviation of (a) C_f and (b) St was evaluated for a randomly selected flow channel shifted incrementally by 1 pixel. Note that y axis limits vary to accommodate the differing scales of oscillation between models.

pixel coordinates rather than learning the relative physical geometry. Because C_f is derived from near-wall velocity gradients, even minor spatial misalignments in the internal feature maps can lead to significant fluctuations in the output. Consequently, the standard approach is unreliable for practical applications such as design optimization, where consistent evaluation of geometry is paramount. The proposed invariant architectures eliminate spatial noise, ensuring changes in prediction reflect actual physical differences, not merely shifts in the input frame.

D. Mechanism of invariance and stride dependency

Having demonstrated the statistical robustness of the proposed model across the dataset, we further probed the underlying mechanism governing these predictions. To isolate the mechanism of variance, we probed the granular sensitivity of the models using a single, randomly selected flow channel subjected to incremental pixel-wise shifts (see Fig. 6). The sum–mirror CNN demonstrated absolute shift invariance, registering 0% deviation at every step. This confirms that the model's stability is not merely a statistical average but a deterministic property of its architecture. Conversely, the baseline models exhibit a periodic error pattern that exposes the fundamental conflict between discrete downsampling and continuous physical properties. As shown in Fig. 6, the deviation oscillates with a frequency perfectly synchronized with the network's total stride. With $k = 5$ pooling layers, the effective stride is $2^5 = 32$ pixels. This correlation confirms that the error is not random noise but a structural aliasing artifact. According to the sampling theorem, translation equivariance is lost when the input shift is not an integer multiple of the subsampling factor.[12] Standard CNNs operate on a fixed downsampling grid. When the input geometry shifts by an amount that does not align with this grid (sub-period shifts), the feature extraction process suffers from aliasing artifacts, causing the internal representation to fluctuate. The sum–mirror CNN eliminates this grid dependency through its global summation mechanism (APS). By aggregating features globally rather than locally, it effectively breaks the linkage to the downsampling grid, ensuring the prediction remains continuous and valid regardless of the feature's alignment with the pixel lattice. Furthermore, we confirmed that this robust shift-invariance is highly consistent regardless of the input geometry's scale or complexity. A detailed comparison between large-scale and small-scale geometries is provided in the Appendix.

VI. CONCLUSION AND OUTLOOK

In this study, we developed two distinct CNN-based models: the sum–mirror CNN and the FFT-CNN. A distinct feature of these architectures is the explicit integration of invariance regarding vertical reflections and periodic horizontal shifts, capturing the structural symmetries of the laminar flow channel configuration. These architectures were found to enhance both predictive performance and overall model robustness. Furthermore, a comparative analysis revealed a significant decrease in parametric complexity relative to counterparts that lack these invariance properties. These results substantiate the conclusion that the explicit incorporation of shift and flip invariances effectively mitigates the sensitivity of CNNs to cyclical signal variations, enabling predictions that are both more stable and computationally efficient.

Consequently, the proposed architectures serve as highly effective surrogate models, offering a practical pathway to accelerate rapid design iterations and optimization cycles in fluid dynamics applications.

From a methodological perspective, the implementation of geometric invariance is orthogonal to uncertainty quantification. Thus, our architecture retains full compatibility with standard probabilistic frameworks for assessing reliability. For instance, the proposed deterministic layers can be readily extended using Bayesian neural networks to model weight distributions or employed within query-by-committee approaches (e.g., deep ensembles) to capture epistemic uncertainty. These integrations would enable the estimation of confidence intervals alongside friction and heat transfer predictions, ensuring the benefits of invariance do not come at the cost of interpretability or reliability assessment.

In terms of future developments, the proposed principles can be naturally extended to three-dimensional (3D) flows, where the curse of dimensionality makes parameter efficiency even more critical.

Furthermore, while recent trends in the field highlight the potential of large-scale models such as transformers for capturing complex turbulent dynamics, the fundamental philosophy of explicitly embedding geometric symmetries remains universally vital. Future research could, therefore, explore how such inductive biases can be effectively incorporated into these emerging data-driven frameworks to ensure physical consistency and robustness across diverse flow regimes.

ACKNOWLEDGMENTS

Y.K. and P.F. acknowledge funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Grant No. FR 4072/3—within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering." J.T. and P.F. acknowledge funding from the pilot program Core-Informatics of the Helmholtz Association (HGF). L.T. and P.F. acknowledge support from the Federal Ministry of Education and Research (BMBF) under Grant No. 01DM21002A (FLAIM). All authors acknowledge support from the state of Baden-Württemberg through bwHPC. Parts of this work were performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden–Württemberg, and by the Federal Ministry of Education and Research.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Yuri Koide: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Jonas Teufel**: Formal analysis (supporting); Investigation (supporting); Methodology (supporting); Visualization (supporting); Writing – original draft (supporting); Writing – review & editing (supporting). **Luca Torresi**: Formal analysis (supporting); Investigation (supporting); Methodology (supporting); Validation (supporting); Writing – original draft (supporting); Writing – review & editing (supporting). **Arjun J. Kaithakkal**: Conceptualization (supporting); Data curation (equal). **Alexander Stroh**: Conceptualization (supporting); Data curation (supporting); Funding acquisition (lead); Project administration (lead); Supervision (supporting). **Pascal Friederich**: Conceptualization (equal); Funding acquisition (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The code and data that support the findings of this study are openly available on GitHub at https://github.com/aimat-lab/Invariant_CNN.

APPENDIX: IMPLEMENTATION DETAILS

1. Hardware and software specification

All computational experiments were conducted on the *bwUniCluster 2.0* high-performance computing (HPC) infrastructure. The computational nodes are equipped with Intel Xeon Platinum 8358 processors, a NVIDIA A100 GPU (80 GB VRAM), and 128 GB of system RAM. The operating environment is Red Hat Enterprise Linux (RHEL) 8.4.

Our proposed code was implemented using Python 3.10, leveraging PyTorch for the neural network architecture. In particular, the results presented were obtained using the following major library versions:

Sum-Mirror CNN: NumPy 1.26.4, Scikit-Learn 1.4.1, PyTorch 2.2.1.
FFT-CNN: NumPy 2.1.3, Scikit-Learn 1.5.2, Matplotlib 3.9.2, PyTorch 2.5.1.

2. Evaluation metrics

a. Regression

The primary objective of this work is the regression of two continuous properties: the friction coefficient C_f and the Stanton number St. As is standard for regression tasks, we employ the *mean absolute error* (MAE),

$$\mathrm{MAE} = \frac{1}{n}\sum_i |y_i^{\mathrm{true}} - y_i^{\mathrm{pred}}|, \tag{A1}$$

which represents the average absolute difference between the ground truth value y^{true} and the predicted value y^{pred}, averaged over n data points.

In addition, we utilize the *coefficient of determination* (R^2),

$$R^2 = 1 - \frac{\sum_i (y_i^{\mathrm{true}} - y_i^{\mathrm{pred}})^2}{\sum_i (y_i^{\mathrm{true}} - \bar{y})^2}, \tag{A2}$$

where $\bar{y}$ denotes the mean of the ground truth values. The R^2 metric serves as a scale-independent performance measure. Ideally, $R^2 = 1.0$ indicates perfect regression, while negative values indicate performance worse than a simple mean predictor. This metric allows for a generalized assessment of regression performance without requiring prior familiarity with the target value range.

i. Transformation invariance To quantify the model's stability against geometric transformations (vertical flips and horizontal shifts), we calculate the relative deviation Δ_i^{rel},

$$\Delta_i^{\mathrm{rel}} = \frac{|y_i^{\mathrm{pred}} - y_i^{\mathrm{aug}}|}{|y_i^{\mathrm{pred}}|}, \tag{A3}$$

for each ith sample, comparing the original prediction y_i^{pred} with the prediction y_i^{aug} obtained from the augmented (shifted or flipped) input.

3. Ablation study: impact of the APS layer

In our proposed architecture, the anti-aliasing pooling/summation (APS) layer is integrated to theoretically guarantee

complete shift invariance by mitigating aliasing artifacts caused by downsampling. To isolate the contribution of this component, we conducted an ablation study by removing the APS layer while retaining the core sum–mirror architecture.

Figure 7 visualizes the localized stability of the model without APS. While the predictions are relatively stable compared to standard CNNs, minor periodic oscillations persist. These deviations correspond to the aliasing errors predicted by the sampling theorem when the input shift is not an integer multiple of the pooling stride. The baseline performance (R^2) remains high, and strict invariance is lost (see Table III).

The high accuracy maintained even without the APS layer suggests that the core sum–mirror strategy (summing and mirroring features) successfully captures the global symmetry of the

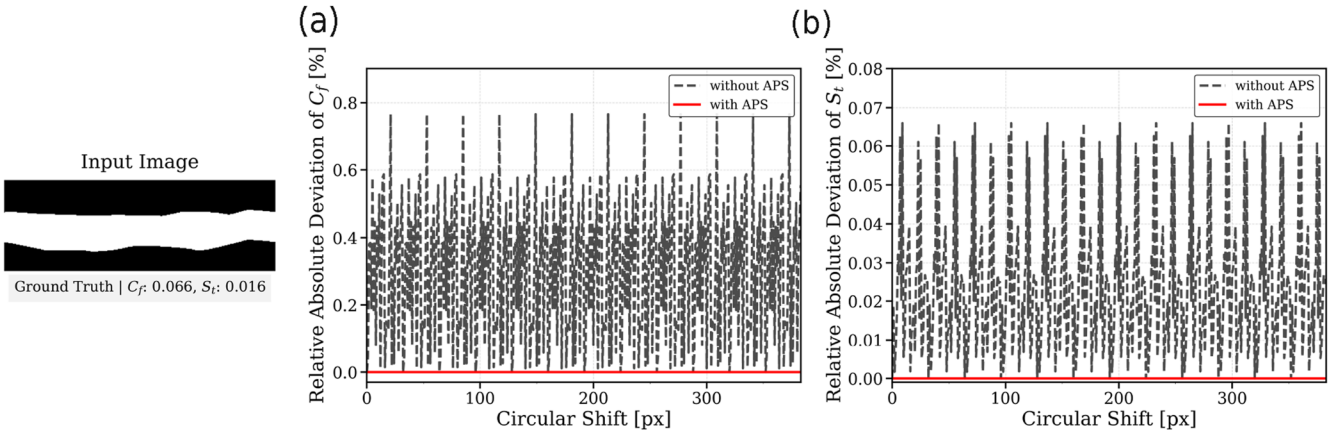

FIG. 7. Comparison of robustness to invariance with sum–mirror CNN and sum–mirror CNN without APS layer. The relative absolute deviation of (a) C_f and (b) St when shifted by 1 pixel was evaluated and compared across models using a randomly selected flow channel (the same example images as shown in Fig. 6). Due to the differing ranges of true values for C_f and St, as well as the variations in oscillation visibility, the y axis limits vary.

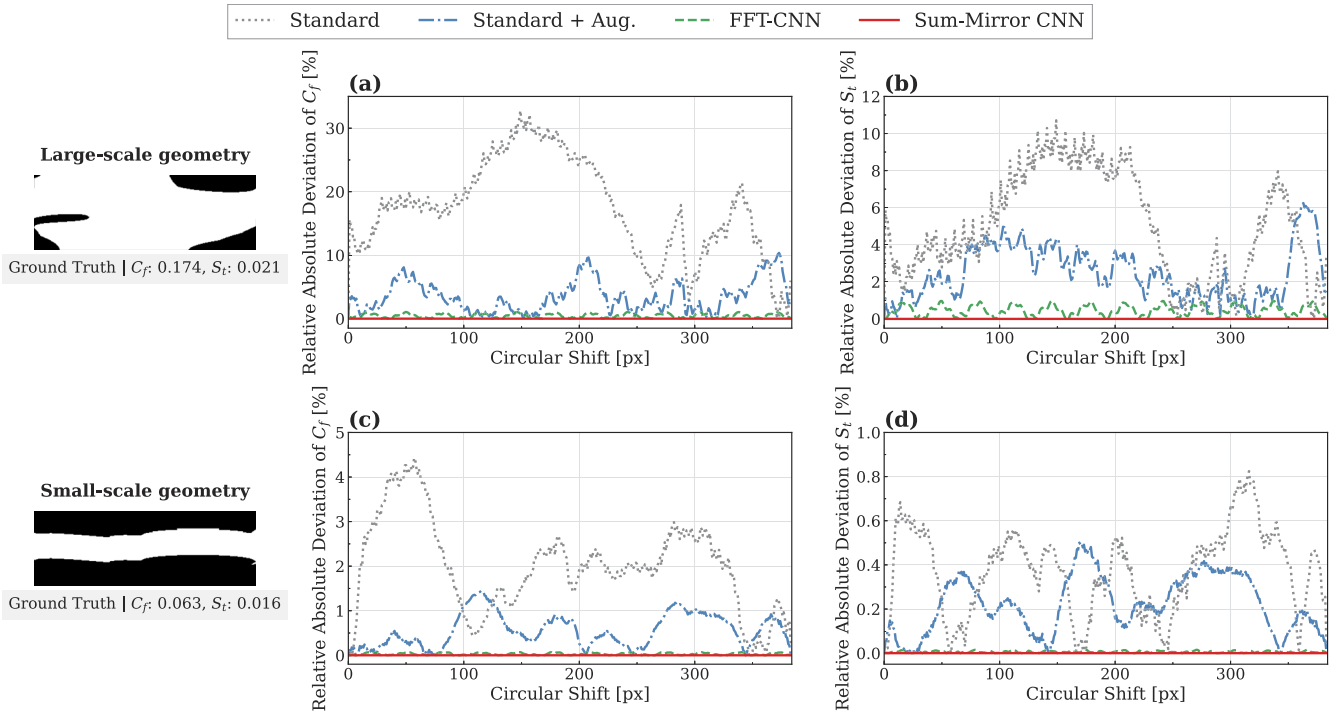

FIG. 8. Robustness to pixel-wise shifts across different geometric scales. Panels [(a) and (b)] and [(c) and (d)] show the relative absolute deviations of C_f and St for large-scale and small-scale geometries, respectively. Note that y-axis scales differ significantly between geometries to visualize the large deviations of the baseline models.

TABLE III. Model accuracy and parameter count for predicting C_f and St. Comparison for the sum–mirror CNN without the APS layer (fivefold cross-validation).

	C_f		St		
CNN model	$R^2 \uparrow$	MAE $\downarrow$	$R^2 \uparrow$	MAE[a] $\downarrow$	No. Parameters
Sum–mirror (w/o APS)	0.954 (±0.009)	0.0019	0.926 (±0.036)	0.1	4.3M

[a]Values multiplied by a factor of 1000.

flow physics. However, the persistence of small oscillations in Fig. 7 reveals a critical insight that while the sum–mirror architecture provides robust feature extraction, the APS layer is indispensable for mathematically ensuring zero variance. Without APS, the model remains susceptible to stride-dependent artifacts, meaning its reliability fluctuates depending on the precise pixel alignment of the input. Therefore, the inclusion of APS is not merely for performance enhancement, but for transforming the model's behavior from "empirically robust" to "structurally invariant."

4. Dependence on the input geometry scale

The proposed invariant models successfully mitigate shape-dependency, unlike standard architectures (see Fig. 8). While the sum–mirror CNN maintains zero deviation and the FFT-CNN keeps errors negligible (e.g., <0.1%) regardless of geometry, standard models exhibit severe performance degradation on large-scale geometries (exceeding 30% deviation for C_f). This vulnerability arises because conventional discrete pooling mechanisms cannot reliably absorb the substantial feature-map variations induced by spatial translations of large-scale geometric features. These results underscore the necessity of strictly invariant architectures for robust evaluation across diverse topologies.

REFERENCES

[1]M. Frank, D. Drikakis, and V. Charissis, "Machine-learning methods for computational science and engineering," Computation **8**(1), 15 (2020).
[2]S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid. Mech. **52**, 477–508 (2020).
[3]D. Fan, L. Yang, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Reinforcement learning for bluff body active flow control in experiments and simulations," Proc. Natl. Acad. Sci. U. S. A. **117**(42), 26091–26098 (2020).
[4]M. Lino, S. Fotiadis, A. A. Bharath, and C. D. Cantwell, "Current and emerging deep-learning methods for the simulation of fluid dynamics," Proc. R. Soc. A **479**(2275), 20230058 (2023).
[5]N. McGreivy and A. Hakim, "Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations," Nat. Mach. Intell. **6**(10), 1256–1269 (2024).
[6]K. Duraisamy, "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence," Phys. Rev. Fluids **6**(5), 050504 (2021).
[7]K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," J. Fluid Mech. **870**, 106–120 (2019).
[8]C. Sabater, P. Stürmer, and P. Bekemeyer, "Fast predictions of aircraft aerodynamics using deep-learning techniques," AIAA J. **60**(9), 5249–5261 (2022).
[9]R. Vinuesa and S. L. Brunton, "Enhancing computational fluid dynamics with machine learning," Nat. Comput. Sci. **2**, 358–366 (2022).
[10]Y. Koide, A. J. Kaithakkal, M. Schniewind, B. P. Ladewig, A. Stroh, and P. Friederich, "Machine learning for rapid discovery of laminar flow channel wall modifications that enhance heat transfer," APL Mach. Learn. **2**(1), 016108 (2024).
[11]A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?," J. Mach. Learn. Res. **20**(184), 1–25 (2019).
[12]R. Zhang, "Making convolutional networks shift-invariant again," in *Proceedings of the 36th International Conference on Machine Learning* (PMLR, 2019), Vol. 97, pp. 7324–7334.
[13]N. Keriven and G. Peyré, "Universal invariant and equivariant graph neural networks," in *Advances in Neural Information Processing Systems*, 32 (Curran Associates, Inc., Red Hook, NY, 2019).
[14]V. Garcia Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," in *International Conference on Machine Learning* (PMLR, 2021), pp. 9323–9332.
[15]J. Chen, E. Hachem, and J. Viquerat, "Graph neural networks for laminar flow prediction around random two-dimensional shapes," Phys. Fluids **33**(12), 123607 (2021).
[16]X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16* (Association for Computing Machinery, New York, NY, 2016), pp. 481–490.
[17]M. Dias Ribeiro, A. Rehman, S. Ahmed, and A. Dengel, "DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks," arXiv:2004.08826 (2020).
[18]B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler, "Deep fluids: A generative network for parameterized fluid simulations," Comput. Graphics Forum **38**(2), 59–70 (2019).
[19]F. Sofos and D. Drikakis, "A review of deep learning for super-resolution in fluid flows," Phys. Fluids **37**(4), 041303 (2025).
[20]D. Drikakis, I. W. Kokkinakis, D. Fung, and S. M. Spottswood, "Generalizability of transformer-based deep learning for multidimensional turbulent flow data," Phys. Fluids **36**(2), 026102 (2024).
[21]J. Viquerat and E. Hachem, "A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number," Comput. Fluids **210**, 104645 (2020).
[22]A. J. Kaithakkal, Y. Koide, M. Schniewind, P. Friederich, and A. Stroh, "Heat transfer enhancement in laminar channel flow by machine learning guided shape optimization of wall geometry," in *International Heat Transfer Conference 17* (Begel House, Inc., 2023).
[23]V. Delchevalerie, A. Bibal, B. Frénay, and A. Mayer, "Achieving rotational invariance with Bessel-convolutional neural networks," in *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21* (Curran Associates, Inc, Red Hook, NY, 2024), pp. 28772–28783.
[24]H. Mo and G. Zhao, "RIC-CNN: Rotation-invariant coordinate convolutional neural network," Pattern Recognit. **146**, 109994 (2024).
[25]A. Chaman and I. Dokmanic, "Truly shift-invariant convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 3773–3783.
[26]T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proceedings of the 33rd International Conference on Machine Learning* (PMLR, 2016), pp. 2990–2999.
[27]M. Weiler and G. Cesa, "General E(2)-Equivariant steerable CNNs," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), Vol. 32.

[28]G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson, "Learning invariances in neural networks from training data," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2020), Vol. 33, pp. 17605–17616.
[29]J. E. Gerken and P. Kessel, "Emergent equivariance in deep ensembles," arXiv:2403.03103 (2024).
[30]K. Fukami, S. Goto, and K. Taira, "Data-driven nonlinear turbulent flow scaling with Buckingham Pi variables," J. Fluid Mech. **984**, R4 (2024).
[31]J. Bons, "A critical assessment of Reynolds analogy for turbine flows," J. Heat Transfer **127**(5), 472–485 (2005).
[32]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (Springer, 2016), pp. 21–37.
[33]X. Zou, F. Xiao, Z. Yu, Y. Li, and Y. J. Lee, "Delving deeper into anti-aliasing in convnets," Int. J. Comput. Vision **131**(1), 67–81 (2023).
[34]A. Chaman and I. Dokmanic, "Truly shift-invariant convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 3773–3783.
[35]K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)* (PMLR, 2019).
[36]B. Alsallakh, N. Kokhlikyan, V. Miglani, J. Yuan, and O. Reblitz-Richardson, "Mind the Pad – CNNs can develop blind spots," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)* (PMLR, 2021).
[37]M. Morimoto, K. Fukami, K. Zhang, A. G. Nair, and K. Fukagata, "Convolutional neural networks for fluid flow analysis: Toward effective metamodeling and low-dimensionalization," Theor. Comput. Fluid Dyn. **35**(5), 633–658 (2021).
[38]O. Rippel, J. Snoek, and R. P. Adams, "Spectral representations for convolutional neural networks," in *Advances in Neural Information Processing Systems*, 28 (Curran Associates, Inc., Red Hook, NY, 2015).
[39]S. O. Ayat, M. Khalil-Hani, A. A.-H. Ab Rahman, and H. Abdellatef, "Spectral-based convolutional neural network without multiple spatial-frequency domain switchings," Neurocomputing **364**, 152–167 (2019).
[40]T. Watanabe and D. F. Wolf, "Image classification in frequency domain with 2SReLU: A second harmonics superposition activation function," Appl. Soft Comput. **112**, 107851 (2021).
[41]A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems, 32* (Curran Associates, Inc., Red Hook, NY, 2019).
[42]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (PMLR, 2015).
[43]M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, http://tensorflow.org/.
[44]F. Chollet *et al.*, Keras (2015), https://keras.io.