PANO: PARALLEL LATENT SPACE NEURAL OPERATOR FOR SOLVING PDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Aggregating observations into a limited number of latent tokens and learning physical interactions in the latent space has emerged as an effective approach for Transformer-based neural operators to solve Partial Differential Equation (PDE) problems involving large-scale observations. However, existing methods typically compress observations using a single aggregation pattern, making physical information not retained by that pattern invisible to the latent space and thereby constraining physical interactions within it. To address this limitation, we propose **PA**rallel Latent Space **N**eural **O**perator (**PANO**), a novel neural operator for solving PDEs that formulates physical interactions among observations as an additive combination of interaction results from parallel latent spaces with shared parameters. Specifically, PANO transforms observations into multiple distinct compressed representations through independent mappings, learns physical interactions in parallel based on these representations, and finally fuses the decoded results to predict the target physical field. Within PANO, we further propose **C**oupled **A**ssignment **P**rojection (**CAP**) to construct bidirectionally consistent encoding and decoding mappings, and **Pa**irwise **L**atent Space **A**ttention Layer (**PaLA**) to jointly update latent representations of each latent space through adaptive cross-space interactions. Extensive experiments on nine benchmarks spanning diverse physical systems, geometries, and observation scales demonstrate that PANO achieves state-of-the-art performance. Notably, PANO reduces equivalent plastic strain prediction errors on Unilateral Stamping by 28.3% (or more) and lift-coefficient prediction errors on AirCraft by 21.4% (or more) compared with existing methods. The source code will be released.

1 INTRODUCTION

Partial Differential Equations (PDEs) play a critical role in describing physical processes in natural and engineering systems (Raissi et al., 2020). Recently, Transformer-based neural operators have emerged as effective methods for solving PDEs on general geometries by embedding observations into point tokens and learning physical interactions among them (Li et al., 2022; Kovachki et al., 2023). For PDE problems involving large-scale observations, recent methods have achieved State-Of-The-Art (SOTA) performance by further aggregating point tokens into a limited number of latent tokens and learning physical interactions in latent spaces, thereby avoiding quadratic interaction costs in the observation space (Alkin et al., 2024; Zhou et al., 2026). However, as shown in Figure 1, these methods typically learn physical interactions among input tokens in a latent space constructed by a single aggregation mapping (Wang & Wang, 2024; Yang et al., 2026), which provides only a compressed representation under a particular aggregation pattern, constraining latent-space physical interactions to the information retained by that pattern.

This limitation is particularly pronounced in models that concentrate multiple interaction layers in a single latent space, as subsequent physical interactions rely on the initial latent tokens obtained through encoding aggregation and cannot recover physical distinctions lost during encoding merely through deeper latent modules (Wang & Wang, 2024). Another type of architecture progressively updates physical states by stacking encoding-interaction-decoding blocks with observation-space residual connections (Luo et al., 2025a). This design provides a bypass for information insufficiently retained by the current aggregation and allows subsequent modules to reaggregate it, but does not ensure that tokens in the current latent space retain enough information to support effective physical

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

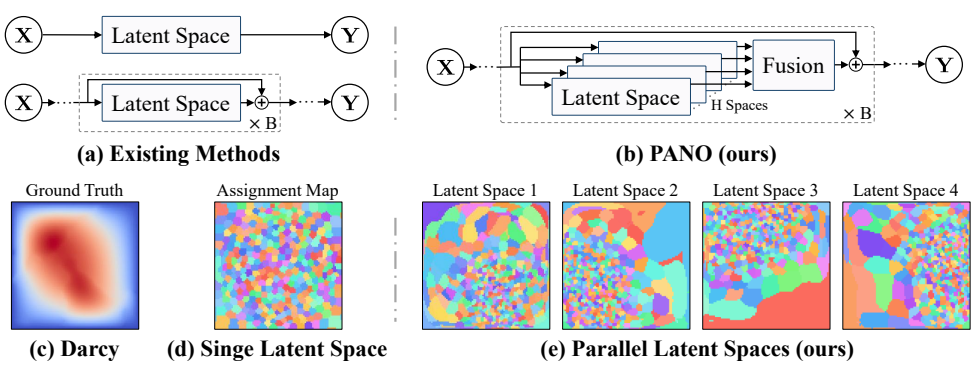


Figure 1: Motivation. (a) Existing methods encode input point tokens into a latent space through a single aggregation pattern. (b) PANO additively combines interaction results from parallel latent spaces. (c) Darcy ground truth. (d–e) Assignment maps of a single-latent-space model (Wang & Wang, 2024) and PANO, with points colored by their highest-weight latent token. Compared with (d), the parallel latent spaces in PANO exhibit distinct local preferences, collectively enabling a more comprehensive latent representation of the input. $\mathbf{X}$ and $\mathbf{Y}$ denote the observation and the target physical field, respectively. B denotes the number of blocks.

interactions (Hu et al., 2026). These issues motivate us to reconsider how information in point tokens can be more fully transferred to latent tokens to support physical interactions in latent spaces.

In this work, we propose **PA**rallel Latent Space **N**eural **O**perator (**PANO**), which formulates physical-state updates of point tokens as an additive combination of physical interaction results from parallel latent spaces. PANO first encodes the same point tokens into multiple latent spaces through independent mappings, forming multiple distinct compressed representations. Each latent space then learns physical interactions in parallel based on its corresponding aggregation results. Finally, the latent tokens from each path are decoded back to the observation space and fused into a combined state update for the point tokens. PANO reduces reliance on any single aggregation mapping, allowing information in point tokens to be more fully transferred to latent tokens through multiple aggregation patterns, thereby alleviating the representational bottleneck of existing methods (Figure 1 (e)). Within PANO, we further propose **C**oupled **A**ssignment **P**rojection (**CAP**) for learnable latent-space construction with bidirectionally consistent encoding and decoding, and **Pa**irwise **L**atent Space **A**ttention Layer (**PaLA**) to jointly update latent tokens across latent spaces through adaptive cross-space interactions. We evaluate PANO on nine benchmarks encompassing standard PDE tasks, as well as challenging engineering tasks involving airfoil flow extrapolation, multi-solid systems, and three-dimensional aerodynamics, showing that PANO outperforms existing SOTA methods. Our main contributions are summarized as follows:

- We propose PANO, a novel neural operator for solving PDEs, which constructs parallel latent spaces to jointly predict target physical fields from observations.
- Within PANO, we propose CAP for bidirectionally consistent encoding and decoding mappings, and PaLA for adaptive cross-space interactions.
- Extensive experiments on nine benchmarks spanning diverse physical systems, geometries, and observation scales demonstrate the advantages of PANO over existing SOTA methods.

2 Related Work

Physics-Informed Neural Networks. Physics-informed neural networks (PINNs) use learnable models to provide continuous approximations of target physical fields (Karniadakis et al., 2021; Raissi et al., 2020; Niaki et al., 2021). Specifically, these models are optimized with physical residual losses derived from the governing equations and solution conditions (Wandel et al., 2022; Yang et al., 2023; Wu & Wu, 2026). Typically, PINNs rely on explicit and sufficiently complete definitions of physical systems, limiting their applicability to complex real-world problems (Luo et al., 2025b).

Neural Operators. Neural operators learn mappings from input functions to target physical fields using large amounts of observation data (Lu et al., 2019; Li et al., 2020a; Kovachki et al., 2023; Zhou et al., 2024; McCabe et al., 2025; Li et al., 2020b; 2023c; Deng et al., 2024; Hao et al., 2023; Tao et al., 2025). Among them, Transformer-based neural operators have achieved substantial progress by adaptively modeling nonlocal relationships and flexibly processing observations across geometries (Li et al., 2022; Xiao et al., 2023; Wang & Wang, 2024). However, the quadratic complexity of standard attention makes direct interactions in the observation space costly (Vaswani et al., 2017). Although some methods (Cao, 2021; Li et al., 2022) reduce interaction complexity through linear attention, scaling to large-scale PDE problems still presents substantial computational and memory challenges (Wu et al., 2024). Recent methods compress observations into latent tokens (Wang & Wang, 2024; Serrano et al., 2024; Alkin et al., 2024; Hu et al., 2026; Yang et al., 2026), enabling PDE solving at the million-point scale (Luo et al., 2025a; Zhou et al., 2026).

3 Parallel Latent Space Neural Operator

Preliminaries. Given a PDE-solving problem defined on an observation domain $\mathcal{D}$ and observations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ collected at N spatial locations, operator learning aims to learn a mapping from the input observations to the target physical field. Depending on the physical system, the observations $\mathbf{X} \in \mathbb{R}^{N \times C_{\text{in}}}$ comprise problem-specific conditions, where C_{in} denotes the observation dimension.

PANO. The overall architecture is illustrated in Figure 2. PANO first embeds observations into point tokens through a pointwise projection, updates them through stacked Parallel Latent Space Blocks (PANO-Blocks), and finally predicts the target physical field through an output projection.

3.1 Parallel Latent Space Block

To reduce the dependence of physical interaction learning on a single aggregation representation, each PANO-Block encodes the same point tokens into H parallel latent spaces through independent mappings, thereby supporting simultaneous physical interaction based on multiple aggregation patterns. A PANO-Block without PaLA can be expressed as:

$$F_{\text{latent}} = \frac{1}{H}\Big(\underbrace{\mathcal{P}^1_{\text{dec}}(\mathcal{M}_{\text{latent}}(\mathcal{P}^1_{\text{enc}}(\text{LN}(F))))}_{\text{Latent space } 1} + \cdots + \underbrace{\mathcal{P}^H_{\text{dec}}(\mathcal{M}_{\text{latent}}(\mathcal{P}^H_{\text{enc}}(\text{LN}(F))))}_{\text{Latent space } H}\Big), \quad (1)$$

where LN denotes LayerNorm, $\mathcal{M}_{\text{latent}}$ denotes the latent module, and $\mathcal{P}^h_{\text{enc}}$ and $\mathcal{P}^h_{\text{dec}}$ denote the h-th latent-space encoding mapping and decoding mapping, respectively. $F, F_{\text{latent}} \in \mathbb{R}^{N \times C}$ denote the input point tokens and the decoded outputs, respectively, with C denoting the feature dimension.

Both $\mathcal{P}^h_{\text{enc}}$ and $\mathcal{P}^h_{\text{dec}}$ are constructed by CAP from the learnable assignments of the h-th latent space. In the full PANO-Block, $\mathcal{M}_{\text{latent}}$ comprises multiple self-attention layers and PaLA, which jointly update latent tokens through within-space and cross-space interactions, respectively. Notably, the parameters of $\mathcal{M}_{\text{latent}}$ are shared across latent spaces within each PANO-Block. Moreover, following common practice (Vaswani et al., 2017; Wu et al., 2024), we combine the input point tokens with the decoded latent interaction results through a residual connection, then apply a pointwise FFN to further integrate features.

3.2 Coupled Assignment Projection

Mapping construction is crucial to the stability of latent spaces (Yang et al., 2026). Therefore, we propose CAP, which constructs bidirectionally consistent encoding and decoding mappings from shared assignments to transfer information between point tokens and latent tokens. Specifically, for the h-th latent space, we first compute row-normalized assignment weights through a learnable mapping, then normalize each column by the total weight received by the corresponding latent token to obtain the assignment matrix $A^h \in \mathbb{R}^{N \times M}$: $R^h = \text{Softmax}_{\text{row}}\left(\text{Proj}^h(\text{LN}(F))\right), A^h = R^h \,\text{Diag}\left((R^h)^\top \mathbf{1}_N\right)^{-1}$, where Proj^h denotes the h-th learnable assignment projector, $R^h \in \mathbb{R}^{N \times M}$ denotes the assignment matrix after row-wise Softmax, M denotes the number of latent tokens per space, $\mathbf{1}_N$ is the all-ones vector of length N, and Diag converts a vector into a diagonal

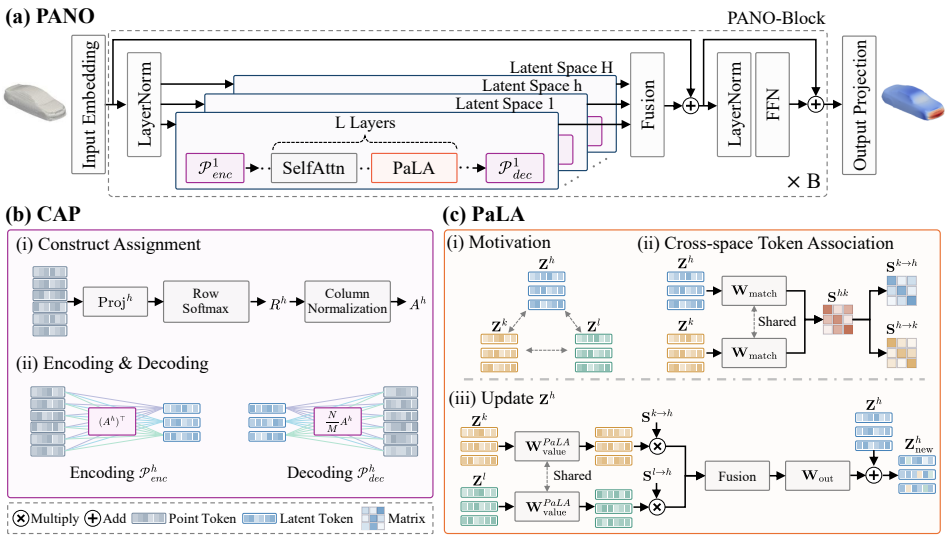


Figure 2: Illustration of PANO. (a) Observations are embedded into point tokens, updated through stacked PANO-Blocks, and projected to the target physical field. (b) CAP constructs shared assignments to obtain bidirectionally consistent encoding and decoding mappings. (c) PaLA learns cross-space token associations. SelfAttn denotes a standard self-attention layer.

matrix. R^h describes the relative assignment weights of each point token across latent tokens. A^h normalizes the total weight received by each latent token to one. Then, we construct the mappings: $\mathcal{P}^h_{\text{enc}}(\cdot) = (A^h)^\top(\cdot), \mathcal{P}^h_{\text{dec}}(\cdot) = \frac{N}{M}A^h(\cdot)$. Encoding aggregates point tokens into latent tokens, while decoding transfers the updated information back to the observation space along the same assignment relations, with N/M adjusting the total-weight scale for the difference in token counts.

Latent-Space Stability Analysis. The construction of $\mathcal{P}^h_{\text{enc}}(\cdot)$ and $\mathcal{P}^h_{\text{dec}}(\cdot)$ ensures that each latent token receives and returns information according to the same relative weight distribution. For convenience, we use $\mathbf{1}_M$ to denote the all-ones vector of length M. Since $(A^h)^\top \mathbf{1}_N = \mathbf{1}_M$, the two directions satisfy the following total-weight constraints: $\mathcal{P}^h_{\text{enc}}(\mathbf{1}_N) = \mathbf{1}_M$ and $\left(\frac{N}{M}A^h\right)^\top \mathbf{1}_N = \frac{N}{M}\mathbf{1}_M$. The first constraint avoids feature-scale differences caused by unequal total weights received by latent tokens. The second ensures that each latent token distributes the same total weight N/M to the point tokens during decoding. These constraints support latent-space stability in terms of feature scale and weight allocation while preserving flexibility in the assignment patterns of latent tokens.

Moreover, we show that the decoding fusion admits an additive kernel representation for $H \geq 1$ parallel latent spaces using CAP and a shared-parameter single-head self-attention layer. We further demonstrate that parallel latent spaces can expand representational capacity while reducing the observation-space information jointly omitted during encoding. The proof of Theorem 1 and proofs of these two properties are provided in Appendix D and Appendix E, respectively.

Theorem 1 (**The Additive Kernel Representation of Parallel Latent Spaces with CAP**). *Let $(\mathcal{D}, \mu)$ be a physical domain equipped with a probability measure μ, and let $\mathcal{F} : \mathcal{D} \to \mathbb{R}^C$ be a bounded measurable point-token field. Assuming finite and measurable assignment logits in each space, the averaged decoded outputs $\mathcal{T}_H$ admit an input-dependent additive kernel representation: $\mathcal{T}_H(\mathcal{F})(x) = \int_{\mathcal{D}} \overline{\kappa}_{\mathcal{F}}(x,\xi)\mathcal{F}(\xi)\mathbf{W}^{self}_{\text{value}}\, d\mu(\xi), \overline{\kappa}_{\mathcal{F}}(x,\xi) = \frac{1}{H}\sum_{h=1}^{H}\kappa^h_{\mathcal{F}}(x,\xi)$, where $x \in \mathcal{D}$ is the output evaluation location, $\xi \in \mathcal{D}$ is the integration variable, $\mathbf{W}^{self}_{\text{value}} \in \mathbb{R}^{C\times C}$ is the shared value projection matrix of self-attention, and $\kappa^h_{\mathcal{F}}$ is the kernel induced by the h-th latent space.*

3.3 Pairwise Latent Space Attention Layer

Decoding-based fusion integrates outputs from latent spaces only in the observation space, without directly establishing interactions across latent spaces. To complement this fusion, we propose PaLA, which learns cross-space token associations to exchange information between latent spaces.

Given latent token sequences $\mathbf{Z}^h, \mathbf{Z}^k \in \mathbb{R}^{M\times C}$ from two distinct latent spaces h and k, we compute cross-space attention scores through a shared matching projection: $\mathbf{S}^{hk} = \frac{(\mathbf{Z}^h\mathbf{W}_{\text{match}})(\mathbf{Z}^k\mathbf{W}_{\text{match}})^\top}{\sqrt{C}}$, where $\mathbf{W}_{\text{match}} \in \mathbb{R}^{C\times C}$ denotes the learnable shared matching matrix, and $\mathbf{S}^{hk} \in \mathbb{R}^{M\times M}$ denotes the token matching scores between the two latent token sequences. We then normalize the scores row-wise in each direction to obtain the attention weights $\mathbf{S}^{h\to k} = \text{Softmax}_{\text{row}}((\mathbf{S}^{hk})^\top)$ and $\mathbf{S}^{k\to h} = \text{Softmax}_{\text{row}}(\mathbf{S}^{hk})$, where the arrows indicate the direction of information transfer. This construction lets latent spaces h and k share token matching relations while independently determining how to aggregate information from each other.

For each latent token sequence $\mathbf{Z}^h$, PaLA averages information from the other latent spaces and applies an output projection and a residual connection to update it:

$$\mathbf{Z}^h_{\text{new}} = \mathbf{Z}^h + (\frac{1}{H-1}\sum_{k\neq h}\mathbf{S}^{k\to h}(\mathbf{Z}^k\mathbf{W}^{PaLA}_{\text{value}}))\mathbf{W}_{\text{out}}, \tag{2}$$

where $\mathbf{W}^{PaLA}_{\text{value}}, \mathbf{W}_{\text{out}} \in \mathbb{R}^{C\times C}$ denote the value and output projections shared across latent spaces, respectively, and $\mathbf{Z}^h_{\text{new}}$ denotes the updated latent tokens. When $H = 1$, no cross-space interaction is available, and PaLA reduces to the identity mapping. We then apply a token-wise FFN with LayerNorm and a residual connection to integrate the updated latent token features.

We further show that decoding fusion of cross-space interactions admits a cross-space kernel representation for $H > 1$ parallel latent spaces using CAP and PaLA with shared parameters.

Theorem 2 (The Cross-Space Kernel Representation of Parallel Latent Spaces with CAP and PaLA). *Under the same assumptions on the domain, input field, and assignments as in Theorem 1, for $H > 1$ parallel latent spaces, the averaged decoded cross-space updates $\mathcal{T}_{\text{cross}}$ admit an input-dependent cross-space kernel representation: $\mathcal{T}_{\text{cross}}(\mathcal{F})(x) = \int_{\mathcal{D}} \kappa^{\text{cross}}_{\mathcal{F}}(x,\xi)\mathcal{F}(\xi)\mathbf{W}^{PaLA}_{\text{value}}\mathbf{W}_{\text{out}}\, d\mu(\xi)$, where $\mathbf{W}^{PaLA}_{\text{value}}$ and $\mathbf{W}_{\text{out}}$ are the shared value and output projection matrices of PaLA, $\kappa^{\text{cross}}_{\mathcal{F}}$ is the kernel induced by cross-space attention of PaLA, and the representation holds for μ-almost every $x \in \mathcal{D}$.*

Building on Theorem 2, we demonstrate that PaLA enables information exchange between latent spaces. We further demonstrate that PaLA can enrich the representation of any latent space by transferring information omitted by its own encoding but retained by other spaces through cross-space interactions. The proof of Theorem 2 and analyses of these properties are provided in Appendix F and Appendix G, respectively.

3.4 PANO Variants

PANO can be scaled by increasing the number of parallel latent spaces H, the number of latent tokens per space M, the feature dimension C, and the depth. To evaluate different scaling configurations, we introduce PANO-Parallel (PANO-P) and PANO-Deep (PANO-D). Both variants contain two PANO-Blocks, with the following configurations:

- **PANO-Parallel (PANO-P)** uses $H = 4$, $M = 512$, and $C = 96$. Each block contains $2\times\text{SelfAttn} \to \text{PaLA} \to 2\times\text{SelfAttn}$, emphasizing physical interaction learning through more parallel aggregation patterns.
- **PANO-Deep (PANO-D)** uses $H = 2$, $M = 1024$, and $C = 192$. Each block contains $3\times\text{SelfAttn} \to \text{PaLA} \to 3\times\text{SelfAttn}$, allocating greater capacity to processing within each latent space.

Both PANO-P and PANO-D use a total of $HM = 2048$ latent tokens per block to learn physical interactions among the input point tokens.

4 Experiments

4.1 Experiment Settings

Benchmarks. We first evaluate PANO on five standard benchmarks (Li et al., 2020a; 2023b), including Airfoil, Plasticity, Navier-Stokes (NS2D), Elasticity, and Darcy.

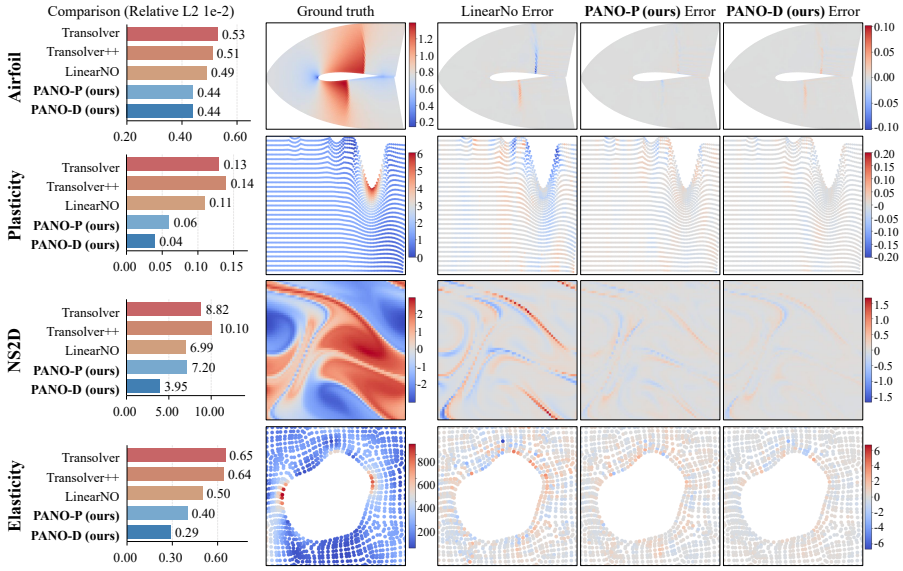


Figure 3: Comparison on standard benchmarks in relative L2 error (↓). PANO-D achieves the best performance. Moreover, compared with LinearNO, PANO exhibits lower prediction errors in regions with sharp variations in the physical fields.

To evaluate the performance of PANO on real-world advanced benchmarks, we further conduct experiments on four benchmarks, including AirfRANS (Bonnet et al., 2022), Unilateral Stamping (Tao et al., 2025), ShapeNet-Car (Umetani & Bickel, 2018), and AirCraft (Luo et al., 2025a). These benchmarks cover two-dimensional airfoil flow prediction and out-of-distribution generalization, stamping in multi-solid systems, three-dimensional car aerodynamics, and three-dimensional AirCraft flow prediction with approximately 330K observations, respectively.

Method	Darcy85	Darcy141	Darcy211
LNO	0.0081	0.0074	0.0073
Transolver	<u>0.0055</u>	0.0064	0.0063
LinearNO	**0.0050**	0.0053	0.0053
PANO-P (ours)	<u>0.0055</u>	<u>0.0042</u>	<u>0.0041</u>
PANO-D (ours)	**0.0050**	**0.0039**	**0.0039**

Table 1: Relative L2 errors (↓) on Darcy at three resolutions. Best and second-best results are **bold** and <u>underlined</u>, respectively.

Baselines. We compare PANO with ten representative methods, including GNOT (Hao et al., 2023), GINO (Li et al., 2023c), LNO (Wang & Wang, 2024), Transolver (Wu et al., 2024), Transolver++ (Luo et al., 2025a), LinearNO (Hu et al., 2026), OFormer (Li et al., 2022), ONO (Xiao et al., 2023), MeshGraphNet (Pfaff et al., 2020), and Unisoma (Tao et al., 2025). Complete results across all benchmarks are provided in the appendix H.

Implementations. Detailed implementation descriptions are provided in the Appendix K.

4.2 Standard Benchmarks

Figure 3 shows the results on standard benchmarks. PANO outperforms existing SOTA methods. Specifically, compared with the best existing method on each task, PANO-D reduces relative L2 errors by 10.2%, 63.6%, 43.5%, and 42.0% on Airfoil, Plasticity, NS2D, and Elasticity, respectively. PANO-P also outperforms existing methods on Airfoil, Plasticity, and Elasticity, while achieving performance comparable to LinearNO on NS2D.

Evaluation at Multiple Observation Densities. Table 1 compares methods on Darcy flow. PANO-D and PANO-P achieve the best performance and the second-best performance at all resolutions, respectively. Compared with the best existing method, PANO-P reduces errors by 20.8% and 22.6% on Darcy141 and Darcy211, respectively, while PANO-D achieves a 26.4% reduction on both. These results indicate that PANO can utilize denser observations more effectively.

Method	Standard Setting				OOD Reynolds		OOD Angles	
	Vol.↓	Surf.↓	C_L ↓	ρ_L ↑	C_L ↓	ρ_L ↑	C_L ↓	ρ_L ↑
GNOT (Hao et al., 2023)	0.0049	0.0152	0.1992	0.9942	0.3268	0.9865	0.3497	0.9868
GINO (Li et al., 2023c)	0.0297	0.0482	0.1821	0.9958	0.4180	0.9645	0.2583	0.9923
LNO (Wang & Wang, 2024)	0.0214	0.0268	0.1480	0.9744	0.7440	0.9399	0.3732	0.9905
Transolver (Wu et al., 2024)	0.0023	0.0085	0.1230	0.9978	0.1622	0.9904	0.2438	0.9948
LinearNO (Hu et al., 2026)	0.0011	0.0077	0.0491	<u>0.9992</u>	0.2400	0.9951	0.0987	0.9963
PANO-P (ours)	**0.0005**	**0.0010**	<u>0.0236</u>	**0.9997**	<u>0.1314</u>	<u>0.9970</u>	<u>0.0671</u>	<u>0.9977</u>
PANO-D (ours)	<u>0.0009</u>	<u>0.0031</u>	**0.0225**	**0.9997**	**0.0834**	**0.9985**	**0.0658**	**0.9980**

Table 2: Comparison on AirfRANS under the standard setting and OOD Reynolds-number and angle-of-attack settings. Vol. and Surf. denote prediction errors for surrounding and surface physical fields, respectively. C_L reports the lift-coefficient prediction error. ρ_L denotes the Spearman rank correlation of C_L. The best and second-best results are **bold** and <u>underlined</u>, respectively.

Method	Geometry		Stress		PEEQ	
	Metal	Rubber	Metal	Rubber	Metal	Rubber
OFormer (Li et al., 2022)	0.0119	0.0098	0.1103	0.0981	0.3269	0.2675
ONO (Xiao et al., 2023)	0.0433	0.0437	0.1377	0.1311	0.5140	0.4748
MeshGraphNet (Pfaff et al., 2020)	0.0391	0.0383	0.1252	0.1417	0.5805	0.5694
Transolver (Wu et al., 2024)	0.0079	0.0075	0.0839	0.0862	0.2938	0.2693
Unisoma (Tao et al., 2025)	0.0071	0.0063	0.0807	0.0773	0.2591	0.2114
PANO-P (ours)	**0.0052**	**0.0044**	**0.0743**	**0.0667**	**0.1827**	**0.1515**
PANO-D (ours)	<u>0.0067</u>	<u>0.0058</u>	<u>0.0756</u>	<u>0.0710</u>	<u>0.2248</u>	<u>0.1822</u>

Table 3: Comparison on Unilateral Stamping (Tao et al., 2025) in relative L2 error (↓). The best and second-best results are **bold** and <u>underlined</u>, respectively.

4.3 Advanced Benchmarks

Evaluation on AirfRANS. Table 2 reports the results on AirfRANS. PANO-P and PANO-D achieve the best or second-best results across all settings.

Under the standard setting, PANO-P reduces prediction errors for surrounding and surface physical fields by 54.5% and 87.0% (or more), respectively. Compared with LinearNO, PANO-P and PANO-D reduce the prediction error of C_L by approximately 51.9% and 54.2%, respectively. Moreover, both achieve the highest ρ_L of 0.9997. Under both OOD settings, PANO-D and PANO-P achieve the best and second-best lift-coefficient prediction results, respectively, highlighting the strong generalization ability of PANO.

Method	Vol.↓	Surf.↓
LNO	0.0269	0.0870
Transolver	0.0221	0.0785
Transolver++	0.0226	0.0800
LinearNO	<u>0.0194</u>	**0.0754**
PANO-P (ours)	0.0214	0.0817
PANO-D (ours)	**0.0187**	<u>0.0784</u>

Table 4: Comparison on ShapeNetCar. Vol. and Surf. report prediction errors for the surrounding velocity field and surface pressure field, respectively.

Evaluation on Unilateral Stamping. We evaluate predictions of deformed geometry, stress, and equivalent plastic strain (PEEQ) on Unilateral Stamping. As shown in Table 3, both PANO variants outperform existing methods across all six metrics. Compared with Unisoma, which is specifically designed for multi-solid systems, PANO-P reduces geometry prediction errors by 26.8% and 30.2%, stress errors by 7.9% and 13.7%, and PEEQ errors by 29.5% and 28.3% for metal and rubber, respectively. Notably, PANO-P outperforms PANO-D across all metrics. This result suggests that providing more parallel aggregation patterns can be more beneficial than increasing model depth for representing multi-solid systems.

Evaluation on ShapeNetCar. Table 4 reports the results on ShapeNetCar. PANO-P and PANO-D achieve comparable performance to existing methods. PANO-D reduces the prediction error of

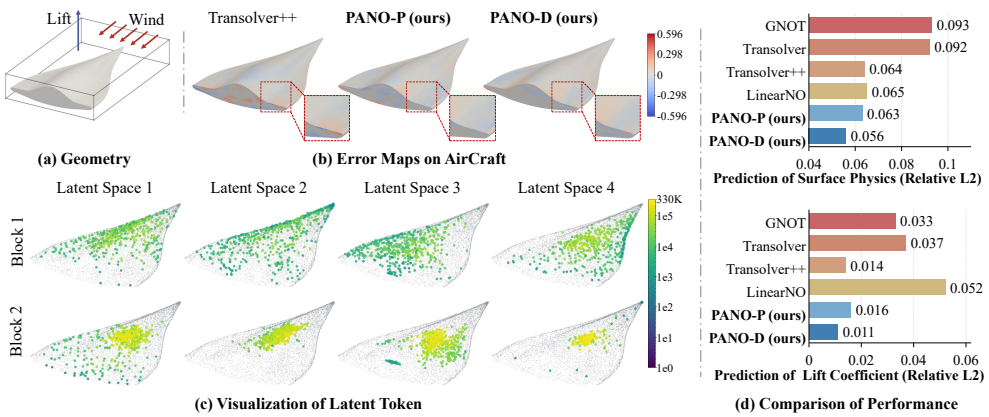


Figure 4: Prediction performance and latent-space representations on AirCraft. (a) AirCraft geometry. (b) Prediction error maps. (c) Visualization of latent tokens learned by PANO-P. We plot each latent token at its weighted center in the physical domain and color it according to the effective number of points it covers. (d) Relative L2 errors (↓) for surface physical fields and lift coefficients. PANO outperforms existing methods. Moreover, PANO-P jointly represents physical states over the AirCraft through multiple latent spaces with distinct regional preferences at each stage.

the surrounding velocity field by 3.6% compared with LinearNO. For surface pressure prediction, PANO-D performs only slightly worse than LinearNO and outperforms the other SOTA methods.

Evaluation on AirCraft. We further evaluate PANO on AirCraft to assess the predictive performance with large-scale observations. As shown in Figure 4, PANO achieves SOTA performance. Compared with Transolver++, PANO-D reduces prediction errors for surface physical fields and the lift coefficient by 12.5% and 21.4%, respectively. These results demonstrate that PANO can scale to large-scale three-dimensional aerodynamic problems.

Model	Params. (M)	GFLOPs
Transolver	3.857	402.03
Transolver++	3.340	<u>356.36</u>
LinearNO	<u>3.322</u>	362.70
PANO-P	**1.989**	**171.78**
PANO-D	8.314	396.21

Table 5: Comparison of parameter counts and computational costs.

Figure 4 (b) further shows that PANO achieves lower prediction errors near AirCraft edges than Transolver++. Figure 4 (c) visualizes the distributions of latent tokens from each latent space of PANO-P in the observation space. Each latent space of PANO-P exhibits distinct regional preferences, indicating that PANO-P jointly represents physical states through multiple spatial aggregation patterns.

4.4 Model Analysis

Efficiency. We compare parameter counts and computational costs using a synthetic input with 100K observations. For Transolver, Transolver++, and LinearNO, we use eight layers, eight attention heads, 256 feature channels, and 32 slices. As shown in Table 5, PANO-P has the fewest parameters and lowest FLOPs among the compared models. PANO-D has more parameters but comparable FLOPs to existing methods.

Parallel Latent Space Block. As shown in Table 6, the model with $H = 1$ substantially degrades performance on both tasks compared with PANO-P. To isolate the effect of parallel aggregation, we further compare the model with PaLA disabled. With CAP enabled, increasing H from 1 to 4 reduces the Airfoil error from 0.0053 to 0.0052 and lowers Geometry, Stress, and PEEQ errors on Unilateral Stamping by 16.7%, 3.1%, and 10.1%, respectively.

CAP. PANO-P outperforms the model without CAP on both Airfoil and Unilateral Stamping. Specifically, with $H = 4$ and PaLA enabled, CAP reduces the prediction error on Airfoil by 8.3%. Notably, CAP also enhances the effectiveness of parallel latent spaces. Specifically, with PaLA disabled, increasing H from 1 to 4 with Softmax does not improve results on Unilateral Stamping,

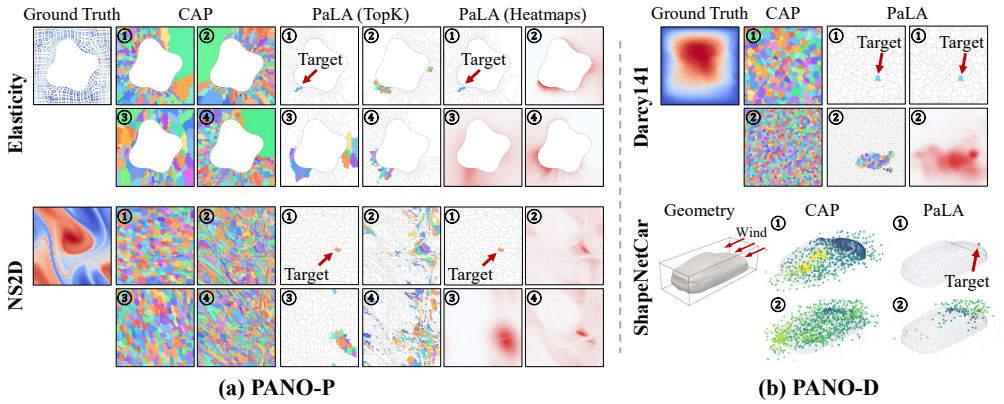


Figure 5: Visualization of CAP and PaLA with (a) PANO-P and (b) PANO-D. CAP is visualized through assignment maps for 2D tasks and latent-token spatial centers for 3D tasks. For PaLA, we select a target token in the first latent space and visualize its attention responses in the other spaces. Specifically, we highlight the Top$K$ tokens with the highest attention weights for the target token in each space. For 2D benchmarks, we additionally visualize the full response heatmaps of PaLA for the target token. Circled numbers (①, ②, ③, ④) indicate the corresponding latent space.

whereas using CAP reduces the PEEQ error by 10.1%. This result shows the importance of stable latent-space construction for realizing the benefits of parallel aggregation.

PaLA. At $H = 4$, PaLA consistently improves predictive performance. Specifically, with CAP disabled, introducing PaLA into the parallel latent spaces reduces the PEEQ error on Unilateral Stamping by 10.6%. Furthermore, compared with PANO-P without PaLA, the full model reduces errors on Airfoil and Unilateral Stamping PEEQ by 15.4% and 10.0%, respectively. These results demonstrate the importance of cross-space interactions.

H	CAP	PaLA	Airfoil	Unilateral Stamping (Metal)		
				Geometry	Stress	PEEQ
1	✗	✗	0.0057	0.0060	0.0759	0.2140
1	✓	✗	0.0053	0.0066	0.0769	0.2256
4	✗	✗	0.0050	0.0061	0.0759	0.2172
4	✓	✗	0.0052	0.0055	<u>0.0745</u>	0.2029
4	✗	✓	<u>0.0048</u>	<u>0.0053</u>	<u>0.0745</u>	<u>0.1941</u>
4	✓	✓	**0.0044**	**0.0052**	**0.0743**	**0.1827**

Table 6: Ablation studies on PANO-P in relative L2 error (↓). Best and second-best results are **bold** and <u>underlined</u>, respectively.

Visualizations. We visualize CAP assignments and PaLA cross-space attention responses across multiple benchmarks in Figure 5. CAP learns distinct spatial aggregation patterns across these benchmarks, allowing parallel latent spaces to represent the same input with different regional preferences and spatial granularities. For PaLA, given a target token in latent space 1, high-weight responses in the other latent spaces exhibit structured spatial distributions that vary with their aggregation patterns.

5 Conclusion

In this work, we proposed PANO, a novel neural operator for solving PDEs on general geometries. PANO formulates physical-state updates of point tokens as an additive combination of interaction results from parallel latent spaces. Within PANO, we propose CAP to construct bidirectionally consistent encoding and decoding mappings, and PaLA to jointly update latent representations through adaptive cross-space interactions. Extensive experiments on nine benchmarks demonstrate the performance, OOD generalization, and scalability of PANO. This work shows that the representational capacity of latent-space neural operators depends not only on latent modules but also on how input tokens are aggregated to latent tokens. By constructing distinct representations in parallel latent spaces with shared parameters, PANO provides a parameter-efficient way to enrich the information available for latent interactions and alleviate the bottleneck of a single compressed representation.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Lucas Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Florent Bonnet, Jocelyn Ahmed Mazari, Paola Cinnella, et al. Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *arXiv preprint arXiv:1611.08097*, 2016.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Jingyang Deng, Xingjian Li, Haoyi Xiong, Xiaoguang Hu, and Jinwen Ma. Geometry-guided conditional adaptation for surrogate models of large-scale 3d pdes on arbitrary geometries. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 5790–5798. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/640. URL https://doi.org/10.24963/ijcai.2024/640. Main Track.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

Wenjie Hu, Sidun Liu, Peng Qiao, Zhenglun Sun, and Yong Dou. Transolver is a linear transformer: Revisiting physics-attention through the lens of linear attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 408–416, 2026.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36:28010–28039, 2023a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023b.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023c.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

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. *arXiv preprint arXiv:2502.02414*, 2025a.

Kuang Luo, Jingshang Zhao, Yingping Wang, Jiayao Li, Junjie Wen, Jiong Liang, Henry Soekmadji, and Shaolin Liao. Physics-informed neural networks for pde problems: A comprehensive review. *Artificial Intelligence Review*, 58(10):323, 2025b.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems*, 37:119301–119335, 2025.

Sina Amini Niaki, Ehsan Haghighat, Trevor Campbell, Anoush Poursartip, and Reza Vaziri. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. *Computer Methods in Applied Mechanics and Engineering*, 384:113959, 2021.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

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*, pp. 652–660, 2017.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

Louis Serrano, Thomas X Wang, Etienne Le Naour, Jean-NoÃĢl Vittaut, and Patrick Gallinari. Aroma: Preserving spatial structure for latent pde modeling with local neural fields. *arXiv preprint arXiv:2406.02176*, 2024.

Shilong Tao, Zhe Feng, Haonan Sun, Zhanxing Zhu, and Yunhuai Liu. Unisoma: A unified transformer-based solver for multi-solid systems. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59132–59153. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/tao25b.html.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021.

Nobuyuki Umetani and Bernd Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Nils Wandel, Michael Weinmann, Michael Neidlin, and Reinhard Klein. Spline-pinn: Approaching pdes without data using fast, physics-informed hermite-spline cnns. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8529–8538, 2022.

Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse pde problems. *Advances in Neural Information Processing Systems*, 37:33085–33107, 2024.

Guoquan Wu and Zhe Wu. A multi-objective optimization framework for adaptive weighting in physics-informed machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26885–26893, 2026.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional pdes with latent spectral models. *arXiv preprint arXiv:2301.12664*, 2023.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Zipeng Xiao, Zhongkai Hao, Bokai Lin, Zhijie Deng, and Hang Su. Improved operator learning by orthogonal attention. *arXiv preprint arXiv:2310.12487*, 2023.

Zijiang Yang, Zhongwei Qiu, and Dongmei Fu. Dmis: Dynamic mesh-based importance sampling for training physics-informed neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5375–5383, 2023.

Zijiang Yang, Xiaomeng Wu, and Dongmei Fu. Mono: Multiscale optimal transport neural operator for solving pdes on general geometries. *arXiv preprint arXiv:2608.09764*, 2026.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: Pde-conditional transformers towards universal neural pde solvers. *arXiv preprint arXiv:2405.17527*, 2024.

Hang Zhou, Haixu Wu, Haonan Shangguan, Yuezhou Ma, Huikun Weng, Jianmin Wang, and Mingsheng Long. Transolver-3: Scaling up transformer solvers to industrial-scale geometries. *arXiv preprint arXiv:2602.04940*, 2026.

APPENDIX CONTENTS

A AI Use Statement **14**

B Reproducibility Statement **14**

C Notation Summary **14**

D Proof of Theorem 1 **14**

D.1 The Integral Kernel Representation for a Single Latent Space 14

D.2 The Additive Kernel Representation for Parallel Latent Spaces 17

E Properties of Theorem 1 **17**

E.1 Discrete Kernel Representation of Parallel Latent Spaces 17

E.2 Expanded Kernel Rank Bound 18

E.3 Reduced Common Encoding Null Space 18

F Proof of Theorem 2 **19**

G Properties of Theorem 2 **20**

G.1 Discrete Kernel Representation of Parallel Latent Spaces with PaLA 20

G.2 Cross-Space Kernel Coupling 20

G.3 Transfer of Information Omitted by Individual Encodings 21

H Detailed Results **21**

H.1 Darcy at Multiple Observation Densities 21

I More Comparison **22**

J Benchmarks **24**

J.1 Standard Benchmarks 24

J.2 Advanced Benchmarks 26

J.3 Data Split 27

K Implementation Details **27**

L Showcases **28**

A AI USE STATEMENT

All major technical contributions and theoretical derivations in this work were conceived and developed by the authors. The core implementation of PANO and its training pipeline were led and completed by the authors. Large language models (LLMs) were used primarily for language polishing and assisting with checks of theoretical derivations, as well as for assisting with visualizations for PANO. All LLM-assisted content was carefully reviewed and verified by the authors.

B REPRODUCIBILITY STATEMENT

In Appendix K, we provide the complete training hyperparameters, hardware environment, and software environment used in our experiments.

C NOTATION SUMMARY

Table 7 summarizes the notation used in the core formulation of PANO. Table 8 summarizes the notation introduced in the theoretical analysis in the main text. Notation used only in the theoretical analysis in the appendix is defined upon first use.

D PROOF OF THEOREM 1

To focus on the kernel representation, the following analysis considers the single-head attention core, omitting LayerNorm, residual connections, and the FFN.

D.1 THE INTEGRAL KERNEL REPRESENTATION FOR A SINGLE LATENT SPACE

We first consider the special case $H = 1$, consisting of CAP encoding, a single self-attention layer, and CAP decoding. Following the proof method of prior work (Wu et al., 2024; Hu et al., 2026; Yang et al., 2026), we show that this composition can be represented as an integral operator.

Theorem 3 (**The Integral Kernel Representation of CAP with a Single Self-Attention Layer**). *Let $(\mathcal{D}, \mu)$ be the original physical domain equipped with a probability measure μ. For any bounded measurable point-token field $\mathcal{F} : \mathcal{D} \to \mathbb{R}^C$, assuming finite and measurable CAP assignment logits, the composition $\mathcal{T}$ of CAP encoding, a single-head latent self-attention layer, and CAP decoding can be represented as an input-dependent integral-kernel operator on the original physical domain:*

$$\mathcal{T}(\mathcal{F})(x) = \int_{\mathcal{D}} \kappa_{\mathcal{F}}(x, \xi)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\, d\mu(\xi), \qquad \mu\text{-a.e. } x \in \mathcal{D}, \tag{3}$$

where $x \in \mathcal{D}$ denotes the spatial location at which the output is evaluated, $\xi \in \mathcal{D}$ is the integration variable, $\mathbf{W}_{\text{value}}^{self}$ is the value projection matrix, and $\kappa_{\mathcal{F}}$ is the input-dependent kernel jointly induced by CAP and latent self-attention.

Proof. Let $a_{\mathcal{F}}(x, j)$ denote the continuous CAP encoding density with respect to μ induced by the input field $\mathcal{F}$, where $x \in \mathcal{D}$ is a spatial location, $j \in \{1, \dots, M\}$ indexes the latent tokens, and M is the number of latent tokens. Specifically, let $r_{\mathcal{F}}(x, j)$ denote the assignment weight obtained by row-wise Softmax. The continuous CAP density is defined as:

$$a_{\mathcal{F}}(x, j) = \frac{r_{\mathcal{F}}(x, j)}{\int_{\mathcal{D}} r_{\mathcal{F}}(\xi, j)\, d\mu(\xi)}. \tag{4}$$

CAP encoding aggregates the input field into a latent token sequence $\mathbf{Z}_{\text{encode}}$ through weighted integration, with its j-th token given by:

$$\mathbf{Z}_{\text{encode}}(j) = \int_{\mathcal{D}} a_{\mathcal{F}}(\xi, j)\mathcal{F}(\xi)\, d\mu(\xi). \tag{5}$$

Notation	Description
Domain, dimensions, and observations	
$\mathcal{D}$	Physical domain of the PDE problem.
N	Number of observation points.
C_{in}	Input observation dimension at each point.
C	Feature dimension of point tokens and latent tokens.
H	Number of parallel latent spaces in each PANO-Block.
M	Number of latent tokens per latent space.
$i \in \{1, \dots, N\}$	Observation-point index.
$h, k \in \{1, \dots, H\}$	Latent-space indices; $h \neq k$ for cross-space interactions.
$\mathbf{x}_i \in \mathbb{R}^{C_{\text{in}}}$	Input observation at the i-th spatial location.
$\mathbf{X} \in \mathbb{R}^{N \times C_{\text{in}}}$	Input observation matrix.
Parallel Latent Space Block	
$F \in \mathbb{R}^{N \times C}$	Input point tokens of a PANO-Block.
$F_{\text{latent}} \in \mathbb{R}^{N \times C}$	Decoded outputs averaged across parallel latent spaces.
$\mathcal{P}^h_{\text{enc}}$	CAP encoding mapping from point tokens to latent tokens in space h.
$\mathcal{P}^h_{\text{dec}}$	CAP decoding mapping from latent tokens in space h to point tokens.
$\mathcal{M}^h_{\text{latent}}$	Latent module for space h.
LN	Layer normalization.
SelfAttn	Self-attention layer for interactions within a latent space.
FFN	Feed-forward network applied independently to each token.
Coupled Assignment Projection	
Proj^h	Learnable assignment projector for space h.
$R^h \in \mathbb{R}^{N \times M}$	Assignment weights after row-wise Softmax.
$A^h \in \mathbb{R}^{N \times M}$	Assignment matrix shared by encoding and decoding in space h.
$\mathbf{1}_N, \mathbf{1}_M$	All-ones vectors of lengths N and M, respectively.
$\text{Softmax}_{\text{row}}$	Softmax applied independently to each matrix row.
Diag	Operator that converts a vector into a diagonal matrix.
Pairwise Latent Space Attention Layer	
$\mathbf{Z}^h \in \mathbb{R}^{M \times C}$	Latent token sequence in space h before the PaLA update.
$\mathbf{Z}^h_{\text{new}} \in \mathbb{R}^{M \times C}$	Latent token sequence after the PaLA attention update.
$\mathbf{W}_{\text{match}} \in \mathbb{R}^{C \times C}$	Matching projection matrix shared across latent spaces.
$\mathbf{W}^{PaLA}_{\text{value}} \in \mathbb{R}^{C \times C}$	PaLA value projection matrix shared across latent spaces.
$\mathbf{W}^{self}_{\text{value}} \in \mathbb{R}^{C \times C}$	Self-attention value projection matrix shared across latent spaces.
$\mathbf{W}_{\text{out}} \in \mathbb{R}^{C \times C}$	PaLA output projection matrix shared across latent spaces.
$\mathbf{S}^{hk} \in \mathbb{R}^{M \times M}$	Matching scores between tokens in spaces h and k.
$\mathbf{S}^{k \to h} \in \mathbb{R}^{M \times M}$	Row-normalized attention weights for transferring information from source space k to target space h.
$\mathbf{S}^{h \to k} \in \mathbb{R}^{M \times M}$	Row-normalized attention weights for the reverse transfer, from space h to space k.

Table 7: Summary of notation used in the core formulation of PANO.

Let $\mathbf{Z}_{\text{decode}}$ denote the latent token sequence after latent interactions and before decoding. CAP maps this sequence back to the original physical domain through the same assignment relations:

$$\mathcal{T}(\mathcal{F})(x) = \frac{1}{M} \sum_{j=1}^{M} a_{\mathcal{F}}(x, j) \mathbf{Z}_{\text{decode}}(j). \tag{6}$$

Here, $\mathbf{Z}_{\text{encode}}(j), \mathbf{Z}_{\text{decode}}(j) \in \mathbb{R}^C$ are the representations of the j-th latent token before and after latent interactions, respectively, and $1/M$ is the CAP decoding normalization factor under the probability measure.

Let $\mathbf{B}_{\mathcal{F}} \in \mathbb{R}^{M \times M}$ be the self-attention weight matrix computed from $\mathbf{Z}_{\text{encode}}$, where $(\mathbf{B}_{\mathcal{F}})_{jk}$ denotes the attention weight assigned by the j-th latent token to the k-th latent token. A single self-attention layer updates the encoded sequence as $\mathbf{Z}_{\text{decode}} = \mathbf{B}_{\mathcal{F}} \mathbf{Z}_{\text{encode}} \mathbf{W}^{self}_{\text{value}}$. Substituting the

Notation	Description
μ	Probability measure on the physical domain $\mathcal{D}$.
$\mathcal{F} : \mathcal{D} \to \mathbb{R}^C$	Bounded measurable point-token field.
$x \in \mathcal{D}$	Spatial location at which the output field is evaluated.
$\xi \in \mathcal{D}$	Integration variable over the physical domain.
$\mathcal{T}_H$	Operator that averages the decoded outputs of H parallel latent spaces using CAP and shared-parameter self-attention.
$\kappa_{\mathcal{F}}^h(x, \xi)$	Input-dependent integral kernel induced by CAP encoding, self-attention, and CAP decoding in space h.
$\overline{\kappa}_{\mathcal{F}}(x, \xi)$	Additive kernel obtained by averaging the kernels of parallel latent spaces.
$\mathcal{T}_{\text{cross}}$	Operator that averages the CAP-decoded cross-space updates produced by PaLA without shortcut.
$\kappa_{\mathcal{F}}^{\text{cross}}(x, \xi)$	Input-dependent cross-space kernel induced by CAP encoding, PaLA attention, and CAP decoding.

Table 8: Summary of notation introduced in the theoretical analysis in the main text.

self-attention update and the CAP encoding expression successively into Equation equation 6 gives:

$$\begin{aligned}\mathcal{T}(\mathcal{F})(x) &= \frac{1}{M}\sum_{j=1}^{M} a_{\mathcal{F}}(x,j)\mathbf{Z}_{\text{decode}}(j) \\ &= \frac{1}{M}\sum_{j=1}^{M} a_{\mathcal{F}}(x,j)\sum_{k=1}^{M}(\mathbf{B}_{\mathcal{F}})_{jk}\mathbf{Z}_{\text{encode}}(k)\mathbf{W}_{\text{value}}^{self} \\ &= \frac{1}{M}\sum_{j,k=1}^{M} a_{\mathcal{F}}(x,j)(\mathbf{B}_{\mathcal{F}})_{jk}\int_{\mathcal{D}} a_{\mathcal{F}}(\xi,k)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\, d\mu(\xi).\end{aligned} \tag{7}$$

Finite and measurable assignment logits ensure that the CAP column-normalization denominators are positive, so $a_{\mathcal{F}}(x, j)$ is finite. In addition, by the CAP normalization, $a_{\mathcal{F}}(\xi, j)$ is nonnegative and integrates to one over the physical domain for each latent token:

$$a_{\mathcal{F}}(x,j) \geq 0, \qquad \int_{\mathcal{D}} a_{\mathcal{F}}(\xi,j)\, d\mu(\xi) = 1, \qquad j = 1,\dots,M. \tag{8}$$

Since $\mathcal{F}$ is bounded and the attention weights are nonnegative with $\sum_k (\mathbf{B}_{\mathcal{F}})_{jk} = 1$, we have:

$$\begin{aligned}&\frac{1}{M}\sum_{j,k=1}^{M} a_{\mathcal{F}}(x,j)(\mathbf{B}_{\mathcal{F}})_{jk}\int_{\mathcal{D}} a_{\mathcal{F}}(\xi,k)\|\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\|\, d\mu(\xi) \\ &\qquad \leq \frac{1}{M}\sum_{j=1}^{M} a_{\mathcal{F}}(x,j)\|\mathcal{F}\|_{L^\infty(\mu)}\|\mathbf{W}_{\text{value}}^{self}\| < \infty,\end{aligned} \tag{9}$$

where $\|\mathbf{W}_{\text{value}}^{self}\|$ denotes the operator norm induced by the vector norm.

Therefore, we can exchange the finite sums with the integral to obtain:

$$\mathcal{T}(\mathcal{F})(x) = \int_{\mathcal{D}} \underbrace{\left[\frac{1}{M}\sum_{j,k=1}^{M} a_{\mathcal{F}}(x,j)(\mathbf{B}_{\mathcal{F}})_{jk}a_{\mathcal{F}}(\xi,k)\right]}_{\kappa_{\mathcal{F}}(x,\xi)} \mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\, d\mu(\xi). \tag{10}$$

This establishes the desired input-dependent integral-kernel representation. □

D.2 The Additive Kernel Representation for Parallel Latent Spaces

Building on Theorem 3, we consider $H \geq 1$ parallel latent spaces with independent CAP mappings and shared attention projection parameters. The proof of Theorem 1 is as follows.

Proof. Let $\mathcal{T}^h$ denote the composition of CAP encoding, self-attention, and CAP decoding in the h-th latent space. By Theorem 3, averaging the outputs of the latent spaces gives:

$$\begin{aligned} \mathcal{T}_H(\mathcal{F})(x) &= \frac{1}{H}\sum_{h=1}^{H} \mathcal{T}^h(\mathcal{F})(x) \\ &= \frac{1}{H}\sum_{h=1}^{H} \int_{\mathcal{D}} \kappa_{\mathcal{F}}^h(x,\xi)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\, d\mu(\xi). \end{aligned} \tag{11}$$

The single-space proof establishes the absolute integrability of each integrand. Since H is finite, these representations hold on a common set of full measure and satisfy:

$$\frac{1}{H}\sum_{h=1}^{H} \int_{\mathcal{D}} \left\| \kappa_{\mathcal{F}}^h(x,\xi)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self} \right\| d\mu(\xi) < \infty, \qquad \mu\text{-a.e. } x \in \mathcal{D}. \tag{12}$$

Therefore, we can exchange the finite sum with the integral and combine the individual kernels into an additive kernel:

$$\mathcal{T}_H(\mathcal{F})(x) = \int_{\mathcal{D}} \underbrace{\left[\frac{1}{H}\sum_{h=1}^{H} \kappa_{\mathcal{F}}^h(x,\xi)\right]}_{\overline{\kappa}_{\mathcal{F}}(x,\xi)} \mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{self}\, d\mu(\xi). \tag{13}$$

When $H = 1$, $\overline{\kappa}_{\mathcal{F}} = \kappa_{\mathcal{F}}^1$, so the single-space case is naturally included in this representation. □

E Properties of Theorem 1

We further use the discrete kernel representation of Theorem 1 to analyze two properties of parallel latent spaces: 1) a larger kernel rank bound and 2) a smaller common encoding null space.

E.1 Discrete Kernel Representation of Parallel Latent Spaces

We provide the discrete representation of Theorem 1 to connect it with the forward computation of parallel CAP encoding, latent self-attention, and decoding fusion in PANO. Let the i-th row of the discrete input matrix F be $\mathcal{F}(x_i)$ at observation location x_i, and use the CAP assignment matrix A^h defined in the main text. CAP encodes the input as $\mathbf{Z}_{\text{encode}}^h = (A^h)^\top F$, after which self-attention produces the sequence for decoding:

$$\mathbf{Z}_{\text{decode}}^h = \mathbf{B}_F^h \mathbf{Z}_{\text{encode}}^h \mathbf{W}_{\text{value}}^{self}, \qquad \mathbf{B}_F^h = \text{Softmax}_{\text{row}}\left(\frac{(\mathbf{Z}_{\text{encode}}^h \mathbf{W}_q)(\mathbf{Z}_{\text{encode}}^h \mathbf{W}_k)^\top}{\sqrt{d_a}}\right), \tag{14}$$

where $\mathbf{B}_F^h \in \mathbb{R}^{M\times M}$ is the attention weight matrix of the h-th latent space, $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{C\times d_a}$ and $\mathbf{W}_{\text{value}}^{self} \in \mathbb{R}^{C\times C}$ are the shared query, key, and value projection matrices, and d_a is the query and key feature dimension.

Applying CAP decoding to the updated sequences and averaging across spaces gives:

$$\begin{aligned} F_{\text{latent}} &= \frac{1}{H}\sum_{h=1}^{H} \mathcal{P}^h_{\text{dec}}\left(\mathbf{Z}^h_{\text{decode}}\right) \\ &= \frac{1}{H}\sum_{h=1}^{H} \frac{N}{M} A^h \mathbf{B}^h_F \mathbf{Z}^h_{\text{encode}} \mathbf{W}^{self}_{\text{value}} \\ &= \frac{1}{H}\sum_{h=1}^{H} \frac{N}{M} A^h \mathbf{B}^h_F (A^h)^\top F \mathbf{W}^{self}_{\text{value}} \\ &= \underbrace{\left[\frac{N}{HM}\sum_{h=1}^{H} A^h \mathbf{B}^h_F (A^h)^\top\right]}_{\overline{\mathbf{K}}_F} F \mathbf{W}^{self}_{\text{value}}. \end{aligned} \tag{15}$$

Thus, the discrete additive kernel matrix acting on the observation space is:

$$\overline{\mathbf{K}}_F = \frac{1}{H}\sum_{h=1}^{H} \mathbf{K}^h_F, \qquad \mathbf{K}^h_F = \frac{N}{M} A^h \mathbf{B}^h_F (A^h)^\top, \qquad \overline{\mathbf{K}}_F, \mathbf{K}^h_F \in \mathbb{R}^{N\times N}. \tag{16}$$

Equation (16) is an exact algebraic identity for finite N, M, H, with input dependence arising from the assignment matrices and attention weights in each space.

E.2 Expanded Kernel Rank Bound

For a given input F, let $A_\oplus = [A^1, \ldots, A^H] \in \mathbb{R}^{N\times HM}$ be the horizontal concatenation of the assignment matrices. The discrete additive kernel of Equation (15) can be written as:

$$\overline{\mathbf{K}}_F = \frac{N}{HM} A_\oplus \,\text{BlockDiag}\left(\mathbf{B}^1_F, \ldots, \mathbf{B}^H_F\right) A_\oplus^\top, \tag{17}$$

where BlockDiag arranges the attention matrices into a block-diagonal matrix. Since the rank of a matrix product cannot exceed the rank of any factor, we have:

$$\text{rank}(\mathbf{K}^h_F) \le \text{rank}(A^h) \le \min(N, M), \qquad \text{rank}(\overline{\mathbf{K}}_F) \le \text{rank}(A_\oplus) \le \min(N, HM). \tag{18}$$

Thus, parallel construction expands the allowable rank bound of the spatial kernel matrix from $\min(N, M)$ to $\min(N, HM)$. This analysis shows that parallel latent spaces can expand the representational capacity of the model while sharing parameters.

E.3 Reduced Common Encoding Null Space

To illustrate how information can be omitted during encoding, we consider a perturbation $\Delta F \in \mathbb{R}^{N\times C}$ to the input point tokens while holding the assignment matrices fixed. For the h-th latent space, the difference between the encoded representations before and after the perturbation is:

$$(A^h)^\top (F + \Delta F) - (A^h)^\top F = (A^h)^\top \Delta F. \tag{19}$$

When a nonzero perturbation satisfies $(A^h)^\top \Delta F = 0$, the input changes while its encoded representation remains unchanged, making the information carried by this perturbation indistinguishable to that latent space. We characterize these invisible perturbations through the null spaces of the encoding mappings to analyze the information jointly omitted by multiple latent spaces.

Since the condition applies column-wise to ΔF, we analyze a spatial perturbation direction $u \in \mathbb{R}^N$ in a single feature channel. Specifically, for a fixed assignment matrix, the spatial information directions omitted by the h-th encoding form the null space $\mathcal{U}^h = \ker((A^h)^\top)$. We further consider such a direction that is invisible to all latent spaces:

$$A_\oplus^\top u = 0 \quad \Longleftrightarrow \quad (A^h)^\top u = 0 \quad \text{for all } h = 1, \ldots, H. \tag{20}$$

Thus, the directions jointly omitted by all parallel latent spaces form the intersection of their null spaces:

$$\mathcal{U}_{\text{common}} = \bigcap_{h=1}^{H} \mathcal{U}^h = \ker\left(A_{\oplus}^{\top}\right). \tag{21}$$

Therefore, the common null space is contained in the encoding null space of every latent space:

$$\mathcal{U}_{\text{common}} \subseteq \mathcal{U}^h, \qquad h = 1, \ldots, H. \tag{22}$$

This shows that parallel latent spaces constructed through distinct mappings can reduce jointly omitted information, allowing information missed by one latent space to be captured by others.

F Proof of Theorem 2

We further consider the composition of CAP encoding, a single PaLA layer, and CAP decoding for $H > 1$ to characterize the cross-space kernels introduced by interactions between latent spaces. As in the analysis of Theorem 1, we focus on the single-head attention core, omitting LayerNorm, residual connections, and the FFN. The proof of Theorem 2 is as follows.

Proof. Following the notation of Theorem 1, let $a_{\mathcal{F}}^h(x, j)$ denote the continuous CAP encoding density of the h-th latent space, which is nonnegative and satisfies $\int_{\mathcal{D}} a_{\mathcal{F}}^h(\xi, j)\, d\mu(\xi) = 1$. CAP encodes the input field into latent token sequences in each space:

$$\mathbf{Z}_{\text{encode}}^h(j) = \int_{\mathcal{D}} a_{\mathcal{F}}^h(\xi, j)\mathcal{F}(\xi)\, d\mu(\xi), \qquad h = 1, \ldots, H. \tag{23}$$

Following the PaLA construction in the main text, the attention weights for information transfer from space k to space h are:

$$\mathbf{S}^{k\to h} = \text{Softmax}_{\text{row}}\left(\frac{(\mathbf{Z}_{\text{encode}}^h \mathbf{W}_{\text{match}})(\mathbf{Z}_{\text{encode}}^k \mathbf{W}_{\text{match}})^{\top}}{\sqrt{C}}\right), \qquad k \neq h, \tag{24}$$

where $\mathbf{W}_{\text{match}}$ is the shared matching projection matrix and $(\mathbf{S}^{k\to h})_{j\ell}$ is the attention weight assigned by the j-th token in target space h to the ℓ-th token in source space k. With the residual connection omitted, PaLA averages incoming information from the other spaces and applies the output projection to obtain the sequence for decoding:

$$\mathbf{Z}_{\text{decode}}^h = \frac{1}{H-1} \sum_{k\neq h} \mathbf{S}^{k\to h} \mathbf{Z}_{\text{encode}}^k \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}}. \tag{25}$$

Applying CAP decoding and averaging across spaces, then successively substituting the PaLA update and encoding expressions, gives:

$$\begin{aligned}
\mathcal{T}_{\text{cross}}(\mathcal{F})(x) &= \frac{1}{HM} \sum_{h=1}^{H} \sum_{j=1}^{M} a_{\mathcal{F}}^h(x, j) \mathbf{Z}_{\text{decode}}^h(j) \\
&= \frac{1}{HM(H-1)} \sum_{h=1}^{H} \sum_{k\neq h} \sum_{j,\ell=1}^{M} a_{\mathcal{F}}^h(x, j) (\mathbf{S}^{k\to h})_{j\ell} \mathbf{Z}_{\text{encode}}^k(\ell) \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}} \\
&= \frac{1}{HM(H-1)} \sum_{h=1}^{H} \sum_{k\neq h} \sum_{j,\ell=1}^{M} a_{\mathcal{F}}^h(x, j) (\mathbf{S}^{k\to h})_{j\ell} \int_{\mathcal{D}} a_{\mathcal{F}}^k(\xi, \ell) \mathcal{F}(\xi) \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}}\, d\mu(\xi).
\end{aligned} \tag{26}$$

By the normalization of the CAP densities, the boundedness of the input field, and the nonnegativity and row normalization of the attention weights, we have:

$$\begin{aligned}
&\frac{1}{HM(H-1)} \sum_{h=1}^{H} \sum_{k\neq h} \sum_{j,\ell=1}^{M} a_{\mathcal{F}}^h(x, j) (\mathbf{S}^{k\to h})_{j\ell} \int_{\mathcal{D}} a_{\mathcal{F}}^k(\xi, \ell) \|\mathcal{F}(\xi) \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}}\|\, d\mu(\xi) \\
&\quad \leq \frac{1}{HM} \sum_{h=1}^{H} \sum_{j=1}^{M} a_{\mathcal{F}}^h(x, j) \|\mathcal{F}\|_{L^\infty(\mu)} \|\mathbf{W}_{\text{value}}^{PaLA}\| \|\mathbf{W}_{\text{out}}\| < \infty,
\end{aligned} \tag{27}$$

where $\|\mathbf{W}_{\text{value}}^{PaLA}\|$ and $\|\mathbf{W}_{\text{out}}\|$ denote the operator norms of the value and output projection matrices induced by the vector norm, respectively. Therefore, we can exchange the finite sums with the integral to obtain:

$$\begin{aligned}\mathcal{T}_{\text{cross}}(\mathcal{F})(x) &= \int_{\mathcal{D}} \kappa_{\mathcal{F}}^{\text{cross}}(x,\xi)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{PaLA}\mathbf{W}_{\text{out}}\, d\mu(\xi),\\ \kappa_{\mathcal{F}}^{\text{cross}}(x,\xi) &= \frac{1}{HM(H-1)}\sum_{h=1}^{H}\sum_{k\neq h}\sum_{j,\ell=1}^{M} a_{\mathcal{F}}^{h}(x,j)(\mathbf{S}^{k\to h})_{j\ell}a_{\mathcal{F}}^{k}(\xi,\ell).\end{aligned} \tag{28}$$

This completes the proof. □

In the full PaLA interaction core (Equation (28)), the cross-space update is added to the original latent representation through a residual connection. Therefore, after CAP decoding and averaging across latent spaces, the output contains both the contribution preserved within each latent space and the contribution transferred across latent spaces. Following the kernel formulation in Theorem 1, the residual branch induced by CAP encoding and decoding within each latent space can be represented by an input-dependent kernel. Combining this residual contribution with the cross-space kernel in Equation (28), the full PaLA interaction can be written as their superposition:

$$\begin{aligned}\mathcal{T}_{\text{PaLA}}(\mathcal{F})(x) &= \int_{\mathcal{D}} \kappa_{\mathcal{F}}^{\text{res}}(x,\xi)\mathcal{F}(\xi)\, d\mu(\xi) + \int_{\mathcal{D}} \kappa_{\mathcal{F}}^{\text{cross}}(x,\xi)\mathcal{F}(\xi)\mathbf{W}_{\text{value}}^{PaLA}\mathbf{W}_{\text{out}}\, d\mu(\xi),\\ \kappa_{\mathcal{F}}^{\text{res}}(x,\xi) &= \frac{1}{HM}\sum_{h=1}^{H}\sum_{j=1}^{M} a_{\mathcal{F}}^{h}(x,j)a_{\mathcal{F}}^{h}(\xi,j).\end{aligned} \tag{29}$$

$\kappa_{\mathcal{F}}^{\text{res}}$ characterizes the same-space information preserved by the residual connection, while $\kappa_{\mathcal{F}}^{\text{cross}}$ characterizes the cross-space information exchange introduced by PaLA.

G Properties of Theorem 2

We analyze the discrete structure of PaLA and the information transfer across latent spaces.

G.1 Discrete Kernel Representation of Parallel Latent Spaces with PaLA

Corresponding to Equation (29), the CAP-encoded sequence is $\mathbf{Z}_{\text{encode}}^{h} = (A^h)^\top F$, and the PaLA update with the residual connection is:

$$\mathbf{Z}_{\text{new}}^{h} = \mathbf{Z}_{\text{encode}}^{h} + \frac{1}{H-1}\sum_{k\neq h}\mathbf{S}^{k\to h}\mathbf{Z}_{\text{encode}}^{k}\mathbf{W}_{\text{value}}^{PaLA}\mathbf{W}_{\text{out}}. \tag{30}$$

Applying CAP decoding to the updated sequences and averaging across spaces gives:

$$\begin{aligned}F_{\text{PaLA}} &= \frac{1}{H}\sum_{h=1}^{H}\frac{N}{M}A^h\mathbf{Z}_{\text{new}}^{h}\\ &= \underbrace{\left[\frac{N}{HM}\sum_{h=1}^{H}A^h(A^h)^\top\right]}_{\mathbf{K}_F^{\text{res}}}F + \underbrace{\left[\frac{N}{HM(H-1)}\sum_{h=1}^{H}\sum_{k\neq h}A^h\mathbf{S}^{k\to h}(A^k)^\top\right]}_{\mathbf{K}_F^{\text{cross}}}F\mathbf{W}_{\text{value}}^{PaLA}\mathbf{W}_{\text{out}},\end{aligned} \tag{31}$$

where $\mathbf{K}_F^{\text{res}}, \mathbf{K}_F^{\text{cross}} \in \mathbb{R}^{N\times N}$ denote the same-space residual kernel and the cross-space interaction kernel, respectively.

G.2 Cross-Space Kernel Coupling

With $A_{\oplus} = [A^1, \ldots, A^H]$, we construct a block matrix $\mathbf{S}_{\oplus} \in \mathbb{R}^{HM\times HM}$ whose (h,k)-th block is:

$$(\mathbf{S}_{\oplus})_{hk} = \begin{cases}\mathbf{S}^{k\to h}, & h\neq k,\\ \mathbf{0}, & h=k.\end{cases} \tag{32}$$

The two discrete kernels can therefore be written as:

$$\mathbf{K}_F^{\text{res}} = \frac{N}{HM} A_{\oplus} A_{\oplus}^{\top}, \qquad \mathbf{K}_F^{\text{cross}} = \frac{N}{HM(H-1)} A_{\oplus} \mathbf{S}_{\oplus} A_{\oplus}^{\top}. \tag{33}$$

Compared with the block-diagonal structure of independent within-space interactions, PaLA introduces off-diagonal blocks that directly connect encoding in source space k with decoding in target space h. Consequently, information extracted through one aggregation pattern can return to the observation space through another, enabling cross-space combinations of latent representations.

G.3 Transfer of Information Omitted by Individual Encodings

To analyze cross-space information transfer, we fix the assignment matrices and attention weights and consider an input perturbation $\Delta F \in \mathbb{R}^{N \times C}$. Suppose that the perturbation is invisible to the h-th encoding, satisfying $(A^h)^{\top} \Delta F = 0$, but is retained by at least one other space. By Equation (30), the change in the representation after the PaLA update is:

$$\begin{aligned} \Delta \mathbf{Z}_{\text{new}}^h &= (A^h)^{\top} \Delta F + \frac{1}{H-1} \sum_{k \neq h} \mathbf{S}^{k \to h} (A^k)^{\top} \Delta F \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}} \\ &= \frac{1}{H-1} \sum_{k \neq h} \mathbf{S}^{k \to h} (A^k)^{\top} \Delta F \mathbf{W}_{\text{value}}^{PaLA} \mathbf{W}_{\text{out}}. \end{aligned} \tag{34}$$

When the total incoming cross-space contribution is nonzero, $\Delta \mathbf{Z}_{\text{new}}^h \neq 0$, allowing the h-th latent space to distinguish input changes that its own encoding did not retain. This property shows that PaLA can transfer information retained by other aggregation patterns into the current latent space, enabling updates beyond the information obtained through its own encoding.

H Detailed Results

Across the benchmarks, we compare PANO with 21 baseline methods, including MLP, PointNet (Qi et al., 2017), Graph U-Net (Bronstein et al., 2016), MeshGraphNet (Pfaff et al., 2020), GNO (Li et al., 2020b), FNO (Li et al., 2020a), GEO-FNO (Li et al., 2023b), F-FNO (Tran et al., 2021), Galerkin (Cao, 2021), OFormer (Li et al., 2022), GNOT (Hao et al., 2023), FactFormer (Li et al., 2023a), ONO (Xiao et al., 2023), LSM (Wu et al., 2023), LNO (Wang & Wang, 2024), GINO (Li et al., 2023c), 3D-GeoCA (Deng et al., 2024), Transolver (Wu et al., 2024), Transolver++ (Luo et al., 2025a), LinearNO (Hu et al., 2026), and Unisoma (Tao et al., 2025). The following tables provide the complete results for the methods evaluated on each benchmark.

Standard Benchmarks. Table 9 reports the comparison on Airfoil, Plasticity, NS2D, and Elasticity in relative L2 error.

H.1 Darcy at Multiple Observation Densities

Table 10 reports results on Darcy flow at three observation resolutions, evaluating how effectively the models utilize observations at different densities.

AirfRANS. Table 11 reports the complete physical-field and lift-coefficient prediction results under the standard AirfRANS setting. Table 12 further reports all metrics under Reynolds-number and angle-of-attack extrapolation to evaluate out-of-distribution generalization.

Unilateral Stamping. Table 13 reports predictions of geometry, stress, and equivalent plastic strain for metal and rubber in the multi-solid stamping system.

ShapeNetCar. Table 14 reports the complete comparison of surrounding velocity, surface pressure, and drag-coefficient predictions on ShapeNetCar.

AirCraft. Table 15 reports surface physical-field and lift-coefficient prediction results on the large-scale three-dimensional AirCraft benchmark.

Method	Airfoil	Plasticity	NS2D	Elasticity
FNO (Li et al., 2020a)	-	-	0.1556	-
GEO-FNO (Li et al., 2023b)	0.0138	0.0074	0.1556	0.0229
F-FNO (Tran et al., 2021)	0.0078	0.0047	0.2322	0.0263
Galerkin (Cao, 2021)	0.0118	0.0120	0.1401	0.0240
OFormer (Li et al., 2022)	0.0183	0.0017	0.1705	0.0183
GNOT (Hao et al., 2023)	0.0076	0.0336	0.1380	0.0086
FactFormer (Li et al., 2023a)	0.0071	0.0312	0.1214	-
ONO (Xiao et al., 2023)	0.0061	0.0048	0.1195	0.0118
LSM (Wu et al., 2023)	0.0059	0.0025	0.1535	0.0218
LNO (Wang & Wang, 2024)	0.0053	0.0028	0.0830	0.0066
Transolver (Wu et al., 2024)	0.0053	0.0013	0.0882	0.0065
Transolver++ (Luo et al., 2025a)	0.0051	0.0014	0.1010	0.0064
LinearNO (Hu et al., 2026)	<u>0.0049</u>	0.0011	<u>0.0699</u>	0.0050
PANO-P (ours)	**0.0044**	<u>0.0006</u>	0.0720	<u>0.0040</u>
PANO-D (ours)	**0.0044**	**0.0004**	**0.0395**	**0.0029**

Table 9: Comparison on standard benchmarks in relative L2 error (↓). "-" indicates an unavailable result. The best and second-best results are **bold** and <u>underlined</u>, respectively.

Method	Darcy85	Darcy141	Darcy211
LNO	0.0081	0.0074	0.0073
Transolver	<u>0.0055</u>	0.0064	0.0063
LinearNO	**0.0050**	0.0053	0.0053
PANO-P (ours)	<u>0.0055</u>	<u>0.0042</u>	<u>0.0041</u>
PANO-D (ours)	**0.0050**	**0.0039**	**0.0039**

Table 10: Relative L2 errors (↓) on Darcy flow at three resolutions. The best and second-best results are **bold** and <u>underlined</u>, respectively; ties are determined at the reported precision.

I More Comparison

We further compare PANO with MoNo (Yang et al., 2026), a recent latent-space neural operator that focuses on constructing stable multiscale latent representations. MoNo progressively compresses physical states into a hierarchy of increasingly compact latent spaces, and identifies assignment imbalance and token collapse as key issues in this hierarchical construction. To address these issues, MoNo introduces CoTAP, an optimal-transport-based projection that enforces balanced assignments between adjacent spaces, thereby stabilizing the progressive latent-space hierarchy.

MoNo is fundamentally different from PANO. In particular, PANO focuses on the representational bottleneck introduced by compressing observations through a single latent space. Since MoNo compresses the representation from the previous level through only a single latent space at each stage, it remains subject to the representational bottleneck induced by single-latent-space compression. In particular, each deeper latent space in MoNo is constructed solely from the compressed representation retained by the preceding level. Therefore, once some observation-space information is discarded during an earlier encoding stage, subsequent deeper latent spaces can only further process the information that has already been preserved, without directly recovering the omitted information.

Based on the theoretical framework of Theorems 1 and 2, this limitation can be further formalized. For the single latent-space encoding at the l-th level, let $\mathbf{A}^{(l)}$ denote its assignment matrix. Any perturbation $\Delta\mathbf{F}^{(l-1)}$ satisfying:

$$\left(\mathbf{A}^{(l)}\right)^{\top} \Delta\mathbf{F}^{(l-1)} = 0 \tag{35}$$

is invisible to the latent representation constructed at that level. Since subsequent deeper levels are constructed only from the already-compressed representation, information omitted by the current encoding cannot re-enter the later latent representations. Therefore, MoNo remains subject to the single-latent-space representational bottleneck characterized in this work.

Method	Vol.↓	Surf.↓	C_L ↓	ρ_L ↑
MLP	0.0081	0.0200	0.2108	0.9932
PointNet (Qi et al., 2017)	0.0253	0.0996	0.1973	0.9919
Graph U-Net (Bronstein et al., 2016)	0.0076	0.0144	0.1677	0.9949
MeshGraphNet (Pfaff et al., 2020)	0.0214	0.0387	0.2252	0.9945
GNO (Li et al., 2020b)	0.0269	0.0405	0.2016	0.9938
GEO-FNO (Li et al., 2023b)	0.0361	0.0301	0.6161	0.9257
Galerkin (Cao, 2021)	0.0074	0.0159	0.2336	0.9951
GNOT (Hao et al., 2023)	0.0049	0.0152	0.1992	0.9942
GINO (Li et al., 2023c)	0.0297	0.0482	0.1821	0.9958
LNO (Wang & Wang, 2024)	0.0214	0.0268	0.1480	0.9744
Transolver (Wu et al., 2024)	0.0023	0.0085	0.1230	0.9978
Transolver++ (Luo et al., 2025a)	0.0068	0.0159	0.1880	0.9910
LinearNO (Hu et al., 2026)	0.0011	0.0077	0.0491	<u>0.9992</u>
PANO-P (ours)	**0.0005**	**0.0010**	<u>0.0236</u>	**0.9997**
PANO-D (ours)	<u>0.0009</u>	<u>0.0031</u>	**0.0225**	**0.9997**

Table 11: Comparison on AirfRANS under the standard setting. Vol. and Surf. report prediction errors for surrounding and surface physical fields. C_L reports the lift-coefficient prediction error, and ρ_L is the Spearman rank correlation between predicted and ground-truth lift coefficients. The best and second-best results are **bold** and <u>underlined</u>, respectively.

Method	OOD Reynolds				OOD Angles			
	Vol.↓	Surf.↓	C_L ↓	ρ_L ↑	Vol.↓	Surf.↓	C_L ↓	ρ_L ↑
MLP	0.0669	0.1153	0.6205	0.9578	0.1309	0.3311	0.4128	0.9572
PointNet (Qi et al., 2017)	0.0838	0.1403	0.3836	0.9806	0.2021	0.4649	0.4425	0.9784
Graph U-Net (Bronstein et al., 2016)	0.0538	0.1168	0.4664	0.9645	0.0979	0.2391	0.3756	0.9816
MeshGraphNet (Pfaff et al., 2020)	0.2789	0.2382	1.7718	0.7631	0.4902	1.1071	0.6525	0.8927
GNO (Li et al., 2020b)	0.0833	0.1562	0.4408	0.9878	0.1626	0.2359	0.3038	0.9884
Galerkin (Cao, 2021)	0.0330	0.0972	0.4615	0.9826	0.0577	0.2773	0.3814	0.9821
GNOT (Hao et al., 2023)	0.0305	0.0959	0.3268	0.9865	0.0471	0.3466	0.3497	0.9868
GINO (Li et al., 2023c)	0.0839	0.1825	0.4180	0.9645	0.1589	0.2469	0.2583	0.9923
LNO (Wang & Wang, 2024)	0.0825	0.1762	0.7440	0.9399	<u>0.0346</u>	**0.0790**	0.3732	0.9905
Transolver (Wu et al., 2024)	0.0122	0.0550	0.1622	0.9904	0.0480	0.2335	0.2438	0.9948
LinearNO (Hu et al., 2026)	<u>0.0112</u>	<u>0.0372</u>	0.2400	0.9951	0.0464	0.2500	0.0987	0.9963
PANO-P (ours)	0.0113	**0.0296**	<u>0.1314</u>	<u>0.9970</u>	**0.0261**	<u>0.1709</u>	<u>0.0671</u>	<u>0.9977</u>
PANO-D (ours)	**0.0106**	0.0462	**0.0834**	**0.9985**	0.0362	0.1823	**0.0658**	**0.9980**

Table 12: Complete AirfRANS results for Reynolds-number and angle-of-attack extrapolation. Vol. and Surf. report prediction errors for surrounding and surface physical fields. C_L reports the lift-coefficient prediction error, and ρ_L is the Spearman rank correlation between predicted and ground-truth lift coefficients. The best and second-best results are **bold** and <u>underlined</u>, respectively.

The core projection method of MoNo, CoTAP, is also different from CAP in this work. Specifically, CoTAP formulates the projection as a balanced optimal-transport problem and imposes marginal constraints on both sides of the transport plan, thereby enforcing balanced assignment mass for both the source and target representations. In contrast, CAP does not impose such full bidirectional balancing constraints. CAP normalizes the assignment along the latent-token dimension so that each latent token receives the same total assignment weight during encoding, and reuses the same assignment relations for encoding and decoding. This construction is sufficient to maintain latent-space stability in terms of feature scale and weight allocation.

Therefore, CAP is naturally compatible with the parallel architecture of PANO. Efficient construction of multiple parallel latent spaces requires the projection mechanism itself to remain lightweight. CoTAP introduces Sinkhorn iterations to enforce balanced transport. Replicating this procedure across multiple parallel branches incurs additional iterative overhead.

Method	Geometry		Stress		PEEQ	
	Metal	Rubber	Metal	Rubber	Metal	Rubber
GEO-FNO (Li et al., 2023b)	0.0109	0.0154	0.1113	0.1124	0.3771	0.3574
GINO (Li et al., 2023c)	0.0188	0.0219	0.1262	0.1438	0.4651	0.4885
GNO (Li et al., 2020b)	0.0426	0.0424	0.1351	0.1814	0.6413	0.6859
LSM (Wu et al., 2023)	0.0367	0.0357	0.1327	0.1335	0.5889	0.6026
Galerkin (Cao, 2021)	0.1007	0.1010	0.1725	0.2794	0.7986	0.8555
FactFormer (Li et al., 2023a)	0.0178	0.0166	0.1095	0.1035	0.3886	0.3022
OFormer (Li et al., 2022)	0.0119	0.0098	0.1103	0.0981	0.3269	0.2675
ONO (Xiao et al., 2023)	0.0433	0.0437	0.1377	0.1311	0.5140	0.4748
MeshGraphNet (Pfaff et al., 2020)	0.0391	0.0383	0.1252	0.1417	0.5805	0.5694
Transolver (Wu et al., 2024)	0.0079	0.0075	0.0839	0.0862	0.2938	0.2693
Unisoma (Tao et al., 2025)	0.0071	0.0063	0.0807	0.0773	0.2591	0.2114
PANO-P (ours)	**0.0052**	**0.0044**	**0.0743**	**0.0667**	**0.1827**	**0.1515**
PANO-D (ours)	<u>0.0067</u>	<u>0.0058</u>	<u>0.0756</u>	<u>0.0710</u>	<u>0.2248</u>	<u>0.1822</u>

Table 13: Comparison on Unilateral Stamping in relative L2 error (↓). Geometry, Stress, and PEEQ evaluate the final deformed geometry, stress, and equivalent plastic strain of metal and rubber, respectively. The best and second-best results are **bold** and <u>underlined</u>, respectively.

Method	Vol.↓	Surf.↓	C_D ↓	ρ_D ↑
MLP	0.0512	0.1304	0.0307	0.9496
PointNet (Qi et al., 2017)	0.0494	0.1104	0.0298	0.9583
Graph U-Net (Bronstein et al., 2016)	0.0471	0.1102	0.0226	0.9725
GNO (Li et al., 2020b)	0.0383	0.0815	0.0172	0.9834
GEO-FNO (Li et al., 2023b)	0.1670	0.2378	0.0664	0.8280
Galerkin (Cao, 2021)	0.0339	0.0878	0.0179	0.9764
GNOT (Hao et al., 2023)	0.0329	0.0798	0.0178	0.9833
GINO (Li et al., 2023c)	0.0386	0.0810	0.0184	0.9826
LNO (Wang & Wang, 2024)	0.0269	0.0870	0.0174	0.9781
Transolver (Wu et al., 2024)	0.0221	0.0785	<u>0.0117</u>	0.9914
Transolver++ (Luo et al., 2025a)	0.0226	0.0800	0.0132	0.9914
LinearNO (Hu et al., 2026)	<u>0.0194</u>	**0.0754**	**0.0106**	**0.9925**
PANO-P (ours)	0.0214	0.0817	0.0126	0.9874
PANO-D (ours)	**0.0187**	<u>0.0784</u>	<u>0.0117</u>	<u>0.9917</u>

Table 14: Comparison on ShapeNetCar. Vol. and Surf. report prediction errors for the surrounding velocity field and surface pressure field. C_D reports the drag-coefficient prediction error, and ρ_D is the Spearman rank correlation between predicted and ground-truth drag coefficients. The best and second-best results are **bold** and <u>underlined</u>, respectively.

As shown in Table 16, we further provide a comparison with MoNo. PANO achieves better performance than MoNo across these benchmarks.

J BENCHMARKS

J.1 STANDARD BENCHMARKS

Airfoil. Airfoil (Li et al., 2023b) considers Mach number prediction around varying airfoil geometries. Each sample is represented on a 221×51 structured mesh, and the model predicts the Mach number field from the two-dimensional mesh coordinates. The dataset contains 1,000 training samples and 200 testing samples.

Plasticity. Plasticity (Li et al., 2023b) considers time-dependent deformation of plastic materials under different dies. The physical domain is represented on a 101×31 structured mesh, and the

Method	Surf.↓	C_L ↓	R_L^2 ↑
PointNet (Qi et al., 2017)	0.169	0.095	0.982
Graph U-Net (Bronstein et al., 2016)	0.161	0.063	0.953
MeshGraphNet (Pfaff et al., 2020)	0.113	0.038	0.993
GNO (Li et al., 2020b)	0.129	0.031	0.991
Galerkin (Cao, 2021)	0.118	0.069	0.879
GEO-FNO (Li et al., 2023b)	0.395	0.243	0.903
GINO (Li et al., 2023c)	0.133	0.047	0.983
GNOT (Hao et al., 2023)	0.093	0.033	0.991
LNO (Wang & Wang, 2024)	0.137	0.091	0.992
3D-GeoCA (Deng et al., 2024)	0.097	0.022	0.993
Transolver (Wu et al., 2024)	0.092	0.037	0.994
Transolver++ (Luo et al., 2025a)	0.064	<u>0.014</u>	<u>0.999</u>
LinearNO (Hu et al., 2026)	0.065	0.052	0.995
PANO-P (ours)	<u>0.063</u>	0.016	**1.000 (0.9995)**
PANO-D (ours)	**0.056**	**0.011**	**1.000 (0.9996)**

Table 15: Comparison on AirCraft. Surf. reports the relative L2 error of surface physical fields, C_L reports the lift-coefficient prediction error, and R_L^2 is the coefficient of determination for lift predictions. Parentheses retain the additional precision reported for PANO. The best and second-best results are **bold** and <u>underlined</u>, respectively.

Method	Standard Benchmarks				Darcy			AirCraft		
	Airfoil↓	Plasticity↓	NS2D↓	Elasticity↓	85↓	141↓	211↓	Surf.↓	C_L ↓	R_L^2 ↑
MoNo (Yang et al., 2026)	0.0048	0.0006	0.0522	0.0033	0.0059	0.0049	0.0046	0.077	0.016	0.999
PANO-P (ours)	**0.0044**	0.0006	0.0720	0.0040	0.0055	0.0042	0.0041	0.063	0.016	1.000 (0.9995)
PANO-D (ours)	**0.0044**	**0.0004**	**0.0395**	**0.0029**	**0.0050**	**0.0039**	**0.0039**	**0.056**	**0.011**	**1.000 (0.9996)**

Table 16: Comparison with MoNo on standard benchmarks and AirCraft. For the standard benchmarks, we compare the relative L2 error (↓). For AirCraft, we report the complete evaluation metrics, including the relative L2 error of surface physical fields, denoted as Surf. (↓), the lift-coefficient prediction error C_L (↓), and the coefficient of determination for lift prediction R_L^2 (↑).

Benchmark	Training Set	Test Set
Airfoil	1000	200
Plasticity	900	80
NS2D	1000	200
Elasticity	1000	200
Darcy	1000	200
AirfRANS (Standard Setting)	720	200
Unilateral Stamping	1000	100
ShapeNetCar	789	100
Aircraft	140	10

Table 17: Data split of benchmarks.

Split	OOD Reynolds: Reynolds Number Range	OOD Angles: AoA Range
Training Set	$[3 \times 10^6,\ 5 \times 10^6]$	$[-2.5^\circ,\ 12.5^\circ]$
Test Set	$[2 \times 10^6,\ 3 \times 10^6] \cup [5 \times 10^6,\ 6 \times 10^6]$	$[-5^\circ,\ -2.5^\circ] \cup [12.5^\circ,\ 15^\circ]$

Table 18: Training and test ranges for the AirfRANS OOD Reynolds and OOD Angles evaluations.

model uses the input conditions and target time to predict deformation states independently at 20 time steps.

Benchmark	Epochs	Peak LR	Batch size	Optimizer	Scheduler
Airfoil	500	8×10^{-4}	4	AdamW	OneCycle
Plasticity	500	10^{-3}	8	AdamW	OneCycle
NS2D	500	10^{-3}	2	AdamW	OneCycle
Elasticity	500	10^{-3}	4	AdamW	OneCycle
Darcy85	500	10^{-3}	4	AdamW	OneCycle
Darcy141	500	10^{-3}	4	AdamW	OneCycle
Darcy211	500	10^{-3}	4 / 2	AdamW	OneCycle
AirfRANS (Standard)	1600	2×10^{-4}	4	AdamW	OneCycle
AirfRANS (OOD Reynolds)	1600	2×10^{-4}	4	AdamW	OneCycle
AirfRANS (OOD Angles)	1600	2×10^{-4}	4	AdamW	OneCycle
Unilateral Stamping	100	10^{-3}	1	AdamW	OneCycle
ShapeNetCar	400	8×10^{-4}	2	AdamW	OneCycle
Aircraft	200	10^{-3}	1	AdamW	OneCycle

Table 19: Training hyperparameters of PANO-P and PANO-D. Peak LR denotes the maximum learning rate of the OneCycle schedule. Paired values correspond to PANO-P / PANO-D.

NS2D. NS2D (Li et al., 2020a) considers the temporal evolution of two-dimensional flow fields under different initial conditions. On a 64×64 regular grid, the model takes observations from 10 preceding time steps and autoregressively predicts the next 10 steps.

Elasticity. Elasticity (Li et al., 2023b) considers stress distributions in varying material structures. Each sample contains 972 observation points, and the model maps their two-dimensional coordinates to the corresponding stress values.

Darcy. Darcy (Li et al., 2020a) considers steady flow through porous media with varying permeability distributions, requiring the model to predict the pressure field from spatial coordinates and permeability coefficients. We subsample the original 421×421 grid to 85×85, 141×141, and 211×211, forming Darcy85, Darcy141, and Darcy211 to evaluate predictive performance at different observation densities.

J.2 Advanced Benchmarks

AirfRANS. AirfRANS (Bonnet et al., 2022) consists of two-dimensional incompressible steady flows around varying airfoil geometries at different Reynolds numbers and angles of attack, generated using the Reynolds-averaged Navier–Stokes equations. Each case contains a variable number of observation points, with spatial coordinates, inlet velocity, distance to the airfoil, and surface normals as inputs, and velocity, pressure, and turbulent kinematic viscosity as prediction targets. We consider three settings: Standard Setting, OOD Reynolds, and OOD Angles, evaluating standard predictive performance and generalization to Reynolds numbers and angles of attack outside the training ranges. In addition to prediction errors in the surrounding flow field and on the airfoil surface, we assess aerodynamic prediction using lift-coefficient errors and Spearman rank correlations.

Unilateral Stamping. Unilateral Stamping (Tao et al., 2025) models the deformation of a hollow metal workpiece and an internal rubber insert during multi-point unilateral stamping, supporting long-time state prediction in complex multi-body systems. The system contains 16 moving rigid dies and one stationary rigid die that shape the metal workpiece, while the rubber insert helps preserve its cross-sectional integrity during deformation. Geometric parameters are sampled uniformly within prescribed ranges to generate diverse stamping configurations. Each sample provides the initial and final system states, with prediction targets comprising the final geometry, internal stress, and equivalent plastic strain (PEEQ) of both metal and rubber.

ShapeNetCar. ShapeNetCar (Umetani & Bickel, 2018) comprises aerodynamic simulations of 889 distinct car geometries traveling at 72 km/h, covering physical fields on the vehicle surface and in the surrounding domain. Each case is represented by 32,186 unstructured mesh points, with three-dimensional coordinates, a signed distance function, and normal vectors as inputs, and three-

dimensional velocity and pressure as prediction targets. Following the data split used by Transolver (Wu et al., 2024), we use 789 cases for training and 100 for testing.

AirCraft. AirCraft (Luo et al., 2025a) is a large-scale three-dimensional aerodynamic benchmark with approximately 330K surface observation points per sample, where the inputs consist of point coordinates, surface normals, and sample-level flow conditions including Mach number, angle of attack, and sideslip angle. The model predicts six pointwise physical quantities, including the pressure coefficient C_p, fluid density ρ, three velocity components (u, v, w), and pressure p.

J.3 Data Split

We follow existing methods Wu et al. (2024); Hu et al. (2026); Tao et al. (2025) for data splitting. In particular, the data split of each benchmark is as shown in Table 17. In addition, the setting of OOD Reynolds and OOD Angles on AirfRANS is shown in Table 18.

K Implementation Details

Baselines. We primarily reuse baseline results reported in prior work. Specifically, baseline results on the standard benchmarks, AirfRANS, and ShapeNetCar are taken from LinearNO (Hu et al., 2026), while those on AirCraft and Unilateral Stamping are taken from Transolver++ (Luo et al., 2025a) and Unisoma (Tao et al., 2025), respectively.

Implementation of PaLA. In practice, we implement and adopt multi-head PaLA instead of the standard single-head PaLA to further enhance cross-space interaction. Specifically, we split the matching and value representations along the feature dimension into multiple attention heads, each independently computing cross-space attention and aggregating information. The results from all heads are concatenated and passed through the output projection, allowing PaLA to integrate cross-space associations from multiple feature subspaces.

Training of PANO. All PANO experiments use a random seed of 0, the AdamW optimizer with a weight decay of 5×10^{-5}, and the OneCycle learning-rate scheduler. Table 19 summarizes the training configurations of both variants across all benchmarks. Both variants use the same settings unless separate values are provided in the order PANO-P / PANO-D.

For all benchmarks except AirfRANS, we perform physical field prediction at all observation points of each sample. For AirfRANS, we follow previous works Wu et al. (2024); Hu et al. (2026) by sampling 32,000 points from each AirfRANS case during training, and repeatedly performing sampling and prediction during inference until all points are covered. We further follow the AirfRANS sampling strategy of MoNo Yang et al. (2026), where each sampled case contains all surface points together with randomly sampled volume points, yielding 32,000 points in total.

Hardware and Software Environment. Except for the experiments on AirCraft, all other experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU. The experiments on AirCraft are conducted on a single NVIDIA H20 GPU with mixed precision to accelerate training. The core software environment consists of PyTorch 2.8.0 and CUDA 12.8.

Evaluation. For all experiments, we evaluate performance using the model saved at the last epoch, without any model selection strategy. We use the relative L2 error to evaluate prediction accuracy across all standard benchmarks. Specifically, for each test sample, the relative L2 error is defined as the L2 error between the predicted and ground-truth physical fields normalized by the L2 norm of the ground truth. The reported result is averaged over all testing samples.

For AirfRANS, ShapeNetCar, and AirCraft, in addition to the relative L2 error of the surrounding velocity field and surface pressure field, we further evaluate aerodynamic performance Wu et al. (2024); Hu et al. (2026). In particular, we introduce the lift coefficient for evaluation on AirfRANS and AirCraft. We introduce the drag coefficient for evaluation on ShapeNetCar. The detailed computation follows Transolver Wu et al. (2024) and Transolver++ Luo et al. (2025a). To evaluate the performance on Unilateral Stamping, we mainly follow UNISOMA Tao et al. (2025) and compute the relative L2 error for the deformed geometry, stress, and equivalent plastic strain of the metal and rubber components.

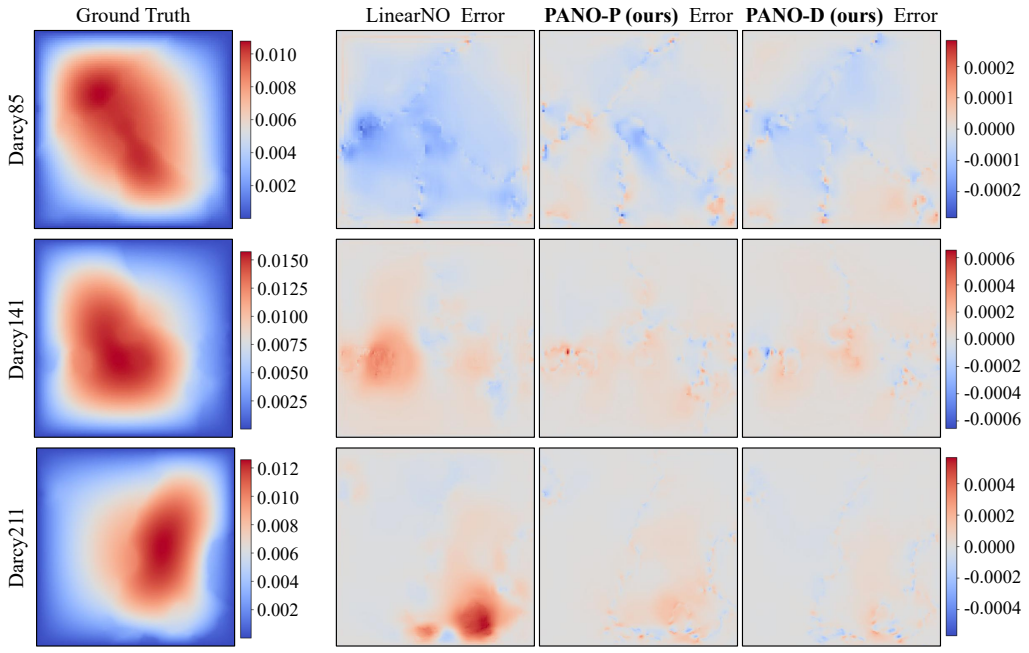


Figure 6: Visualization of prediction error on Darcy.

L SHOWCASES

Prediction Errors. Figures 6, 7, and 8 visualize prediction errors on Darcy, Unilateral Stamping, and ShapeNetCar, respectively.

Latent-Space Representations. We further visualize the latent-space assignments learned by PANO-P and PANO-D. Assignment maps are shown for 2D benchmarks, while latent-token spatial centers are shown for ShapeNetCar. Results for Darcy85, Darcy141, and Darcy211 are shown in Figures 9–10, Figures 11–12, and Figures 13–14, respectively. Additional visualizations are provided for Airfoil in Figures 15–16, Elasticity in Figures 17–18, NS2D in Figures 19–20, Plasticity in Figures 21–22, and ShapeNetCar in Figures 23–24.

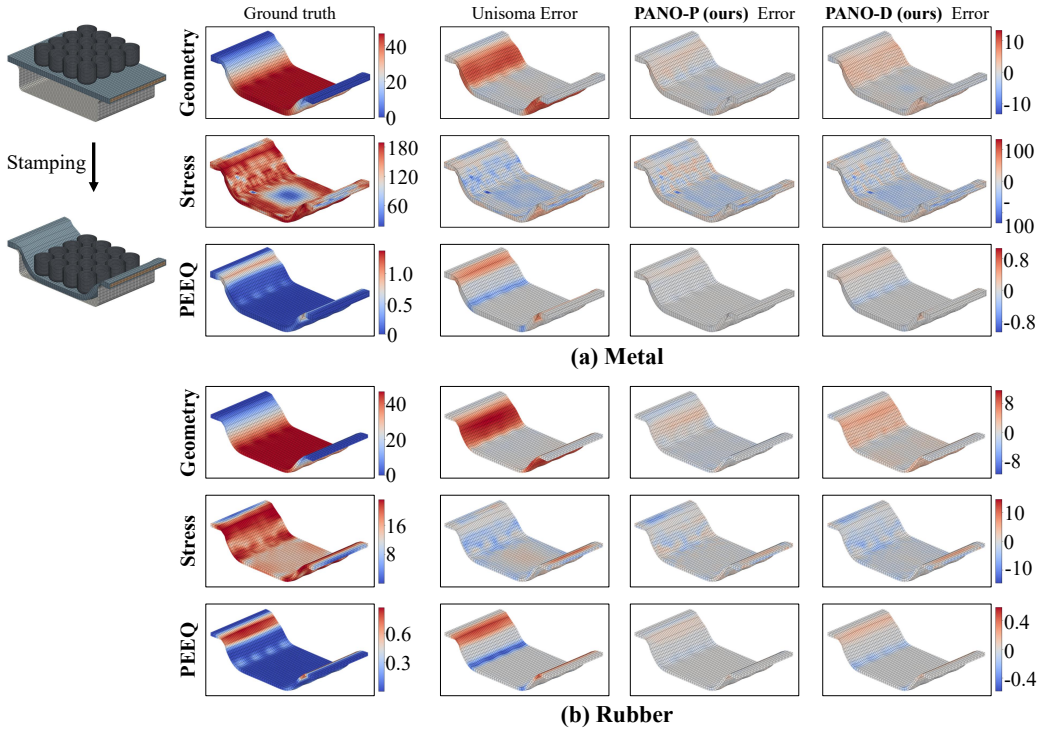


Figure 7: Visualization of prediction error on Unilateral Stamping.

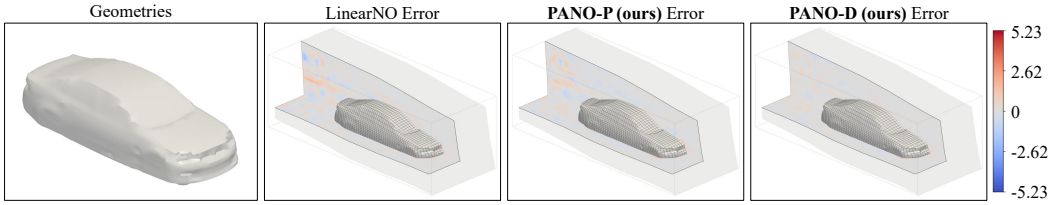


Figure 8: Visualization of prediction error on ShapeNetCar.

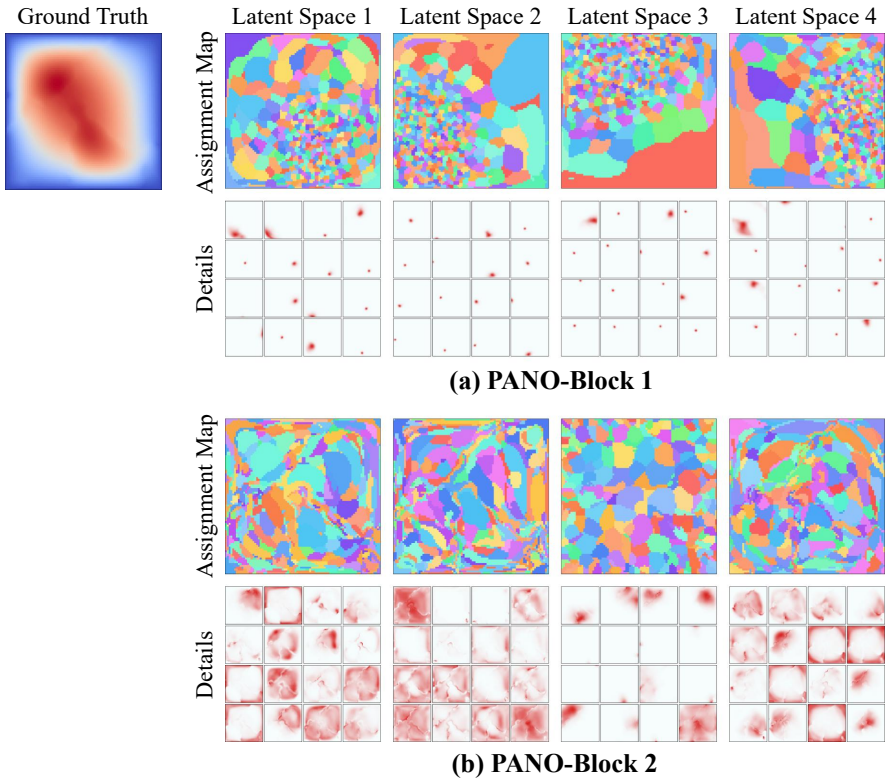


Figure 9: Visualization of PANO-P assignment maps on Darcy85.

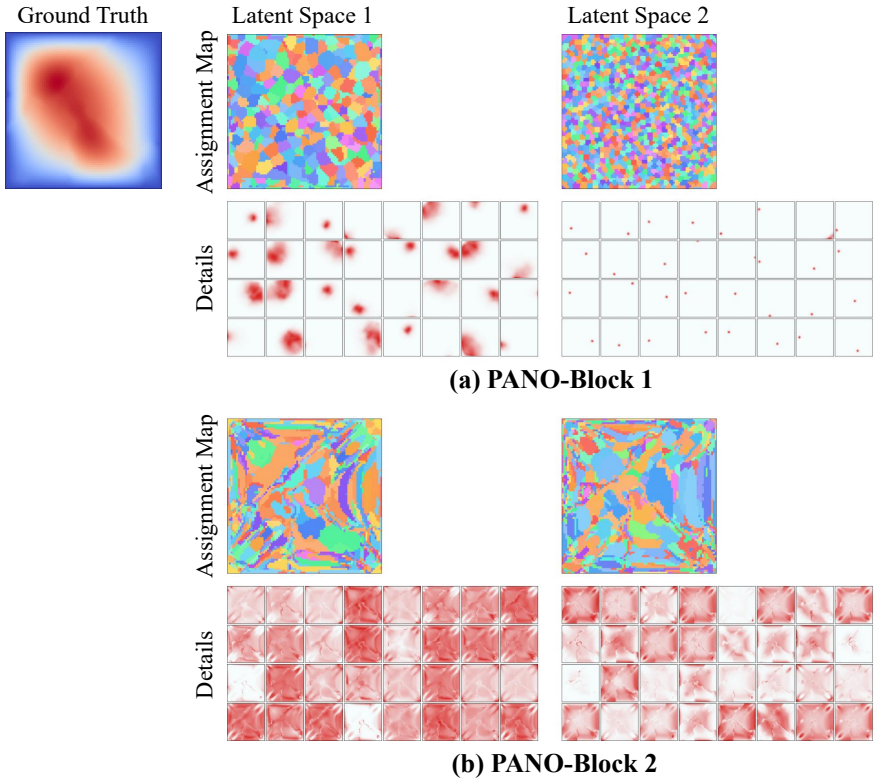


Figure 10: Visualization of PANO-D assignment maps on Darcy85.

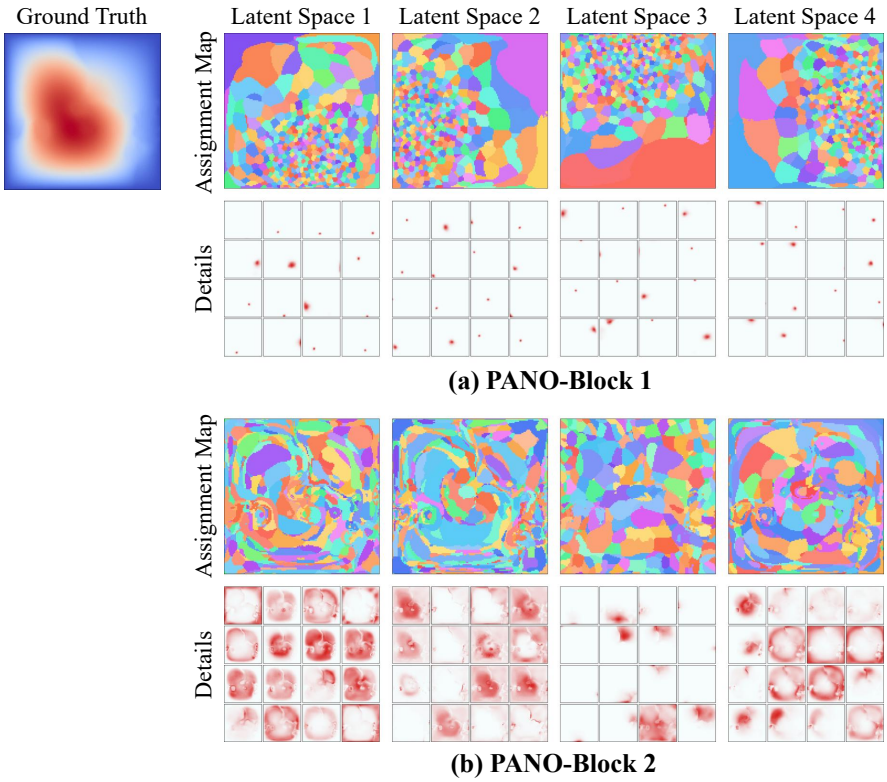


Figure 11: Visualization of PANO-P assignment maps on Darcy141.

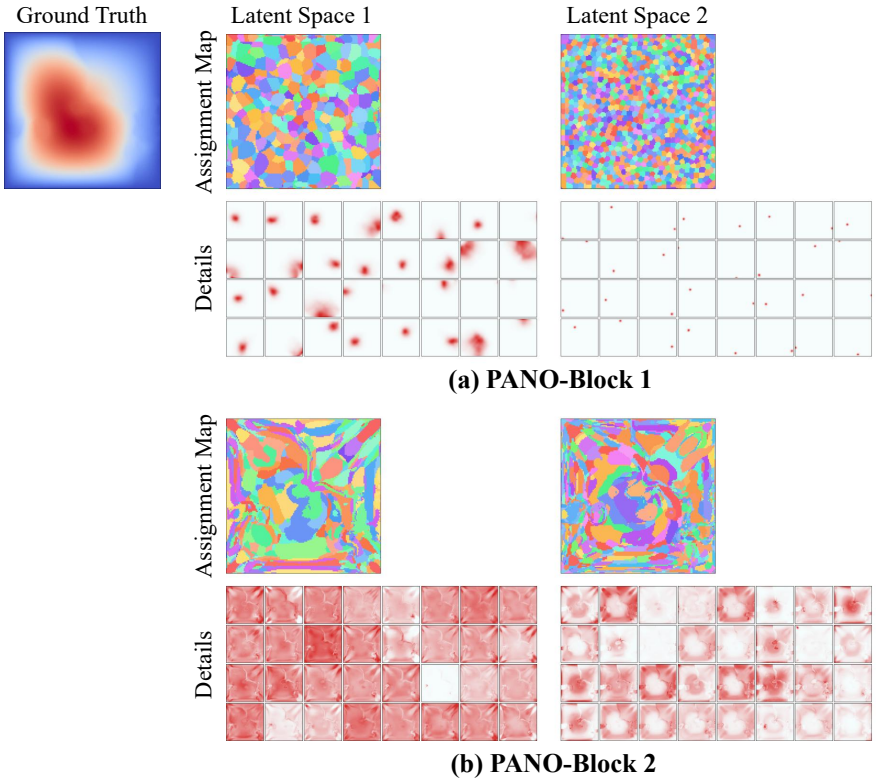


Figure 12: Visualization of PANO-D assignment maps on Darcy141.

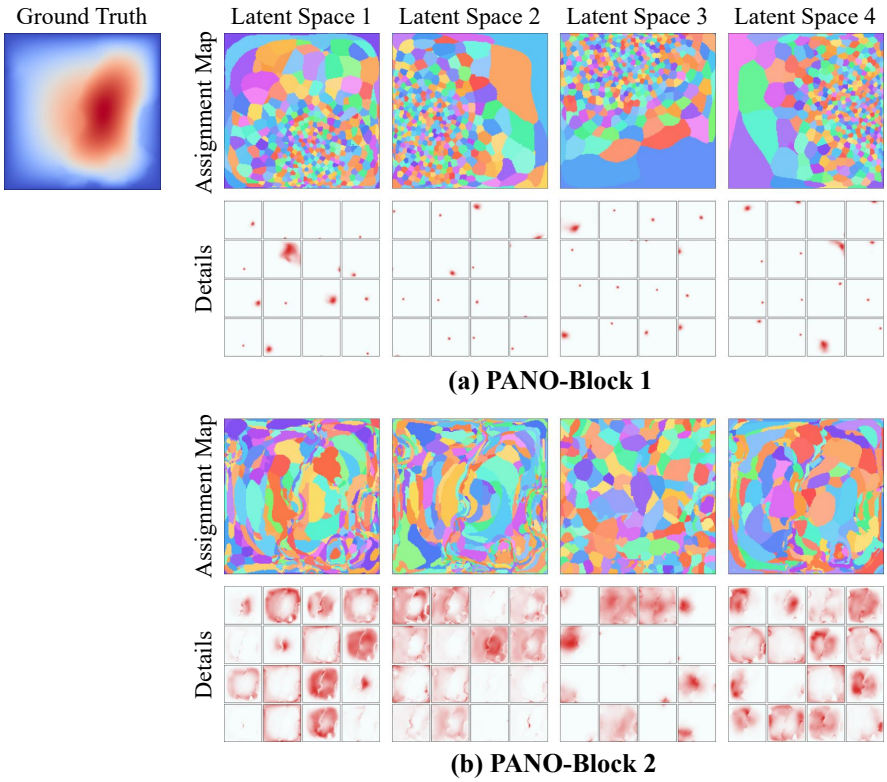


Figure 13: Visualization of PANO-P assignment maps on Darcy211.

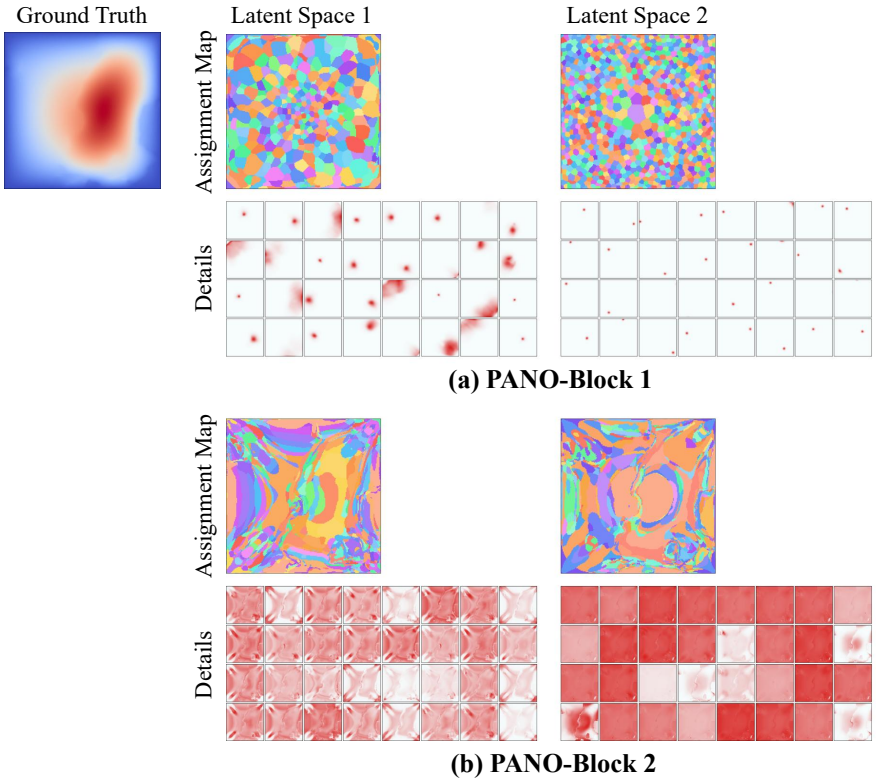


Figure 14: Visualization of PANO-D assignment maps on Darcy211.

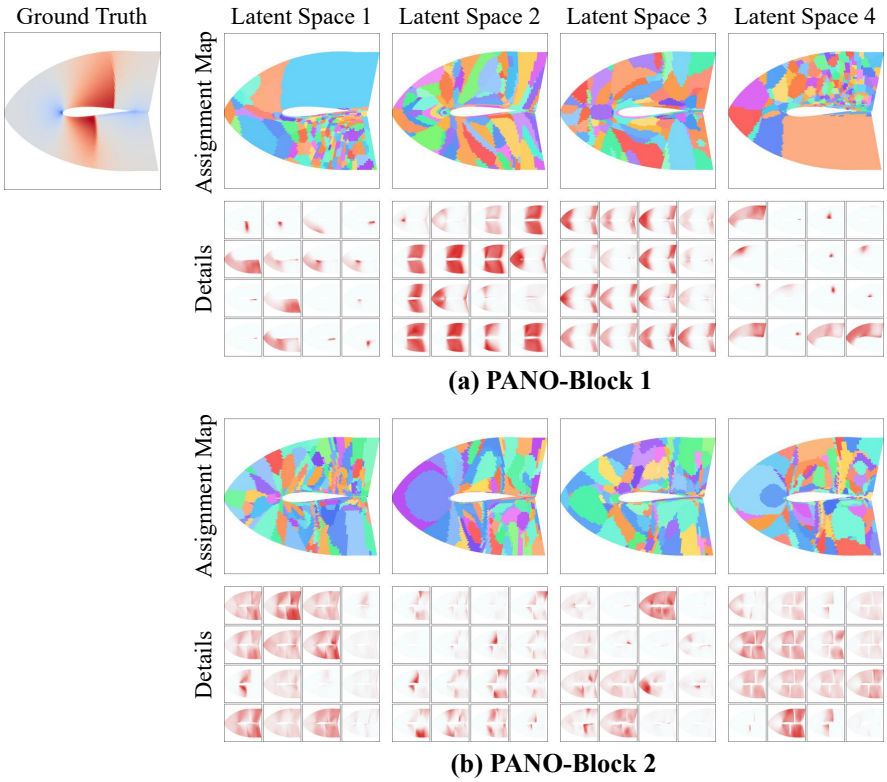


Figure 15: Visualization of PANO-P assignment maps on Airfoil.

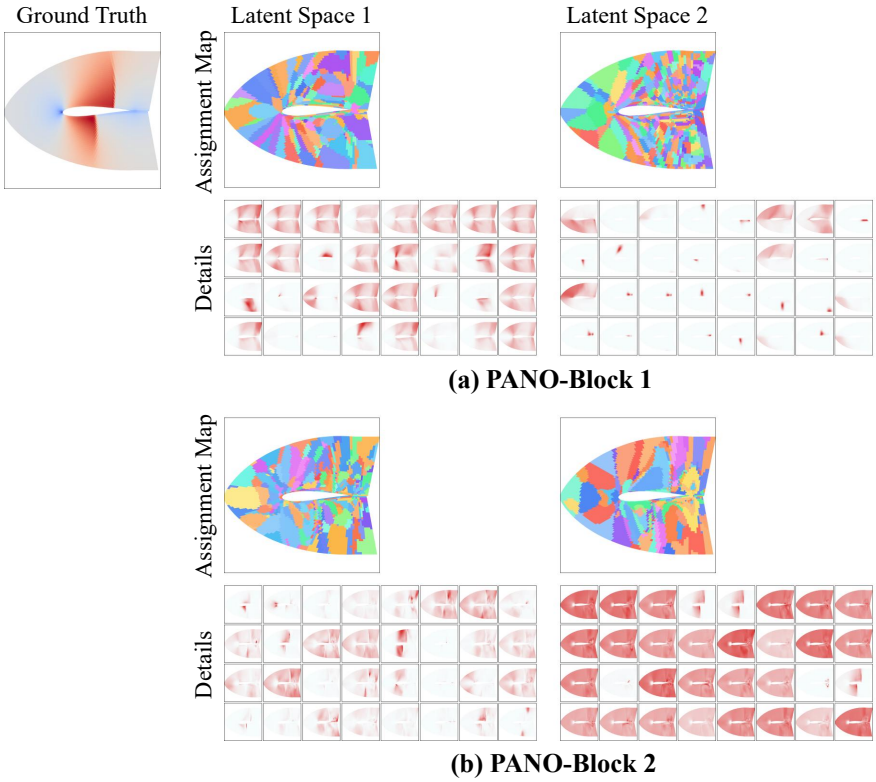


Figure 16: Visualization of PANO-D assignment maps on Airfoil.

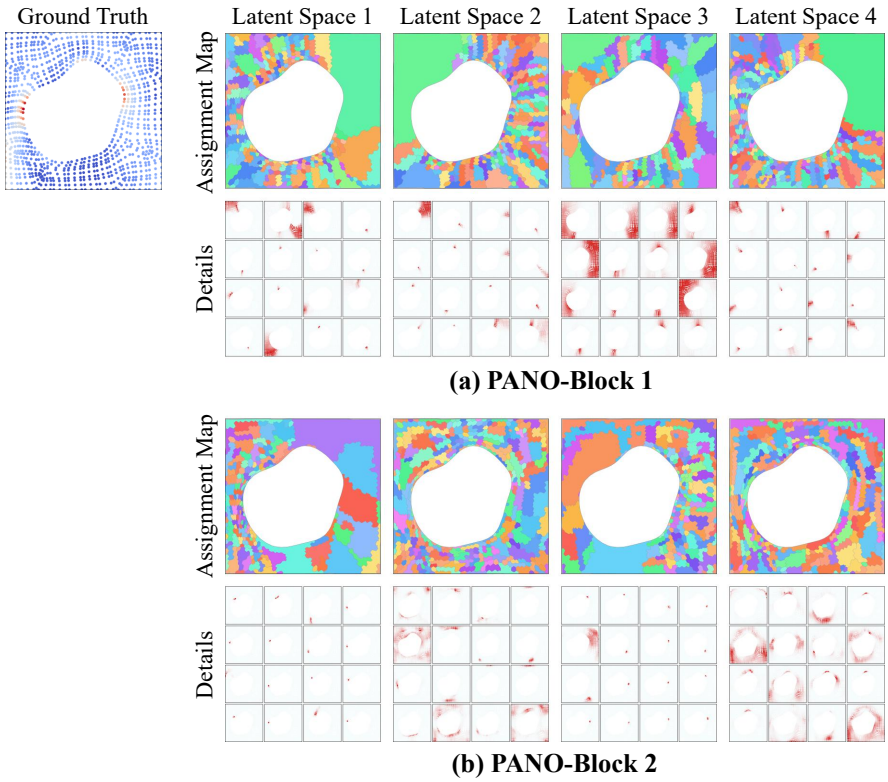


Figure 17: Visualization of PANO-P assignment maps on Elasticity.

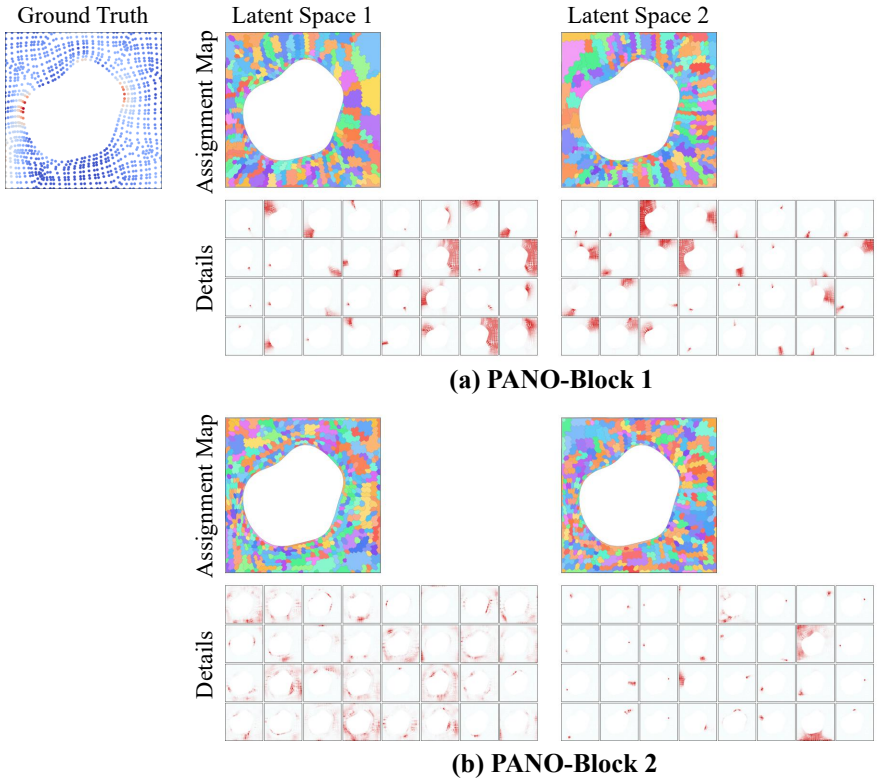


Figure 18: Visualization of PANO-D assignment maps on Elasticity.

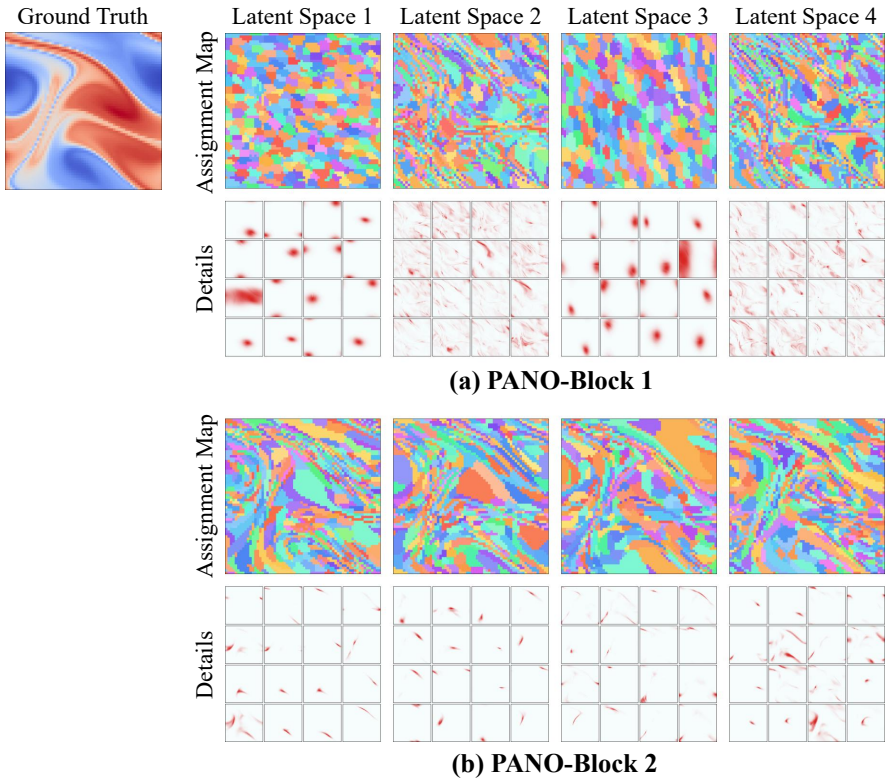


Figure 19: Visualization of PANO-P assignment maps on NS2D.

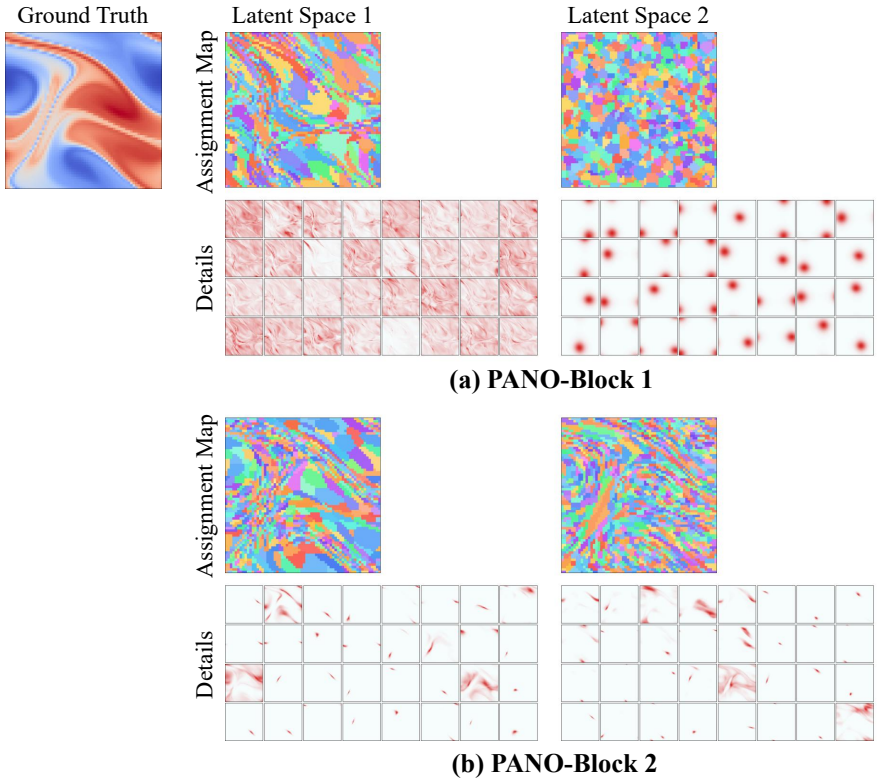


Figure 20: Visualization of PANO-D assignment maps on NS2D.

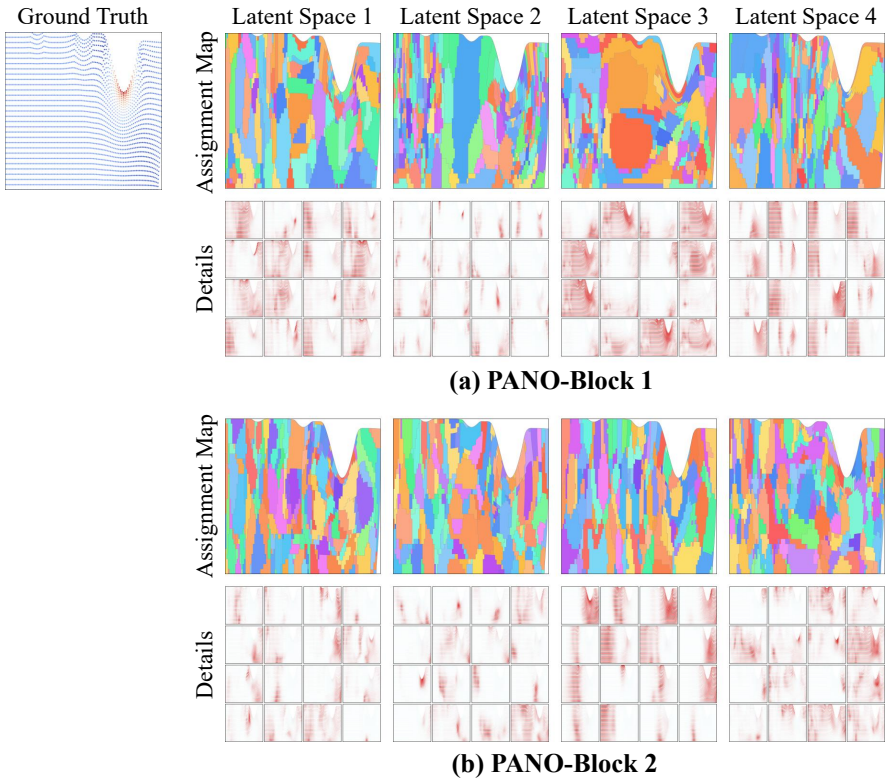


Figure 21: Visualization of PANO-P assignment maps on Plasticity.

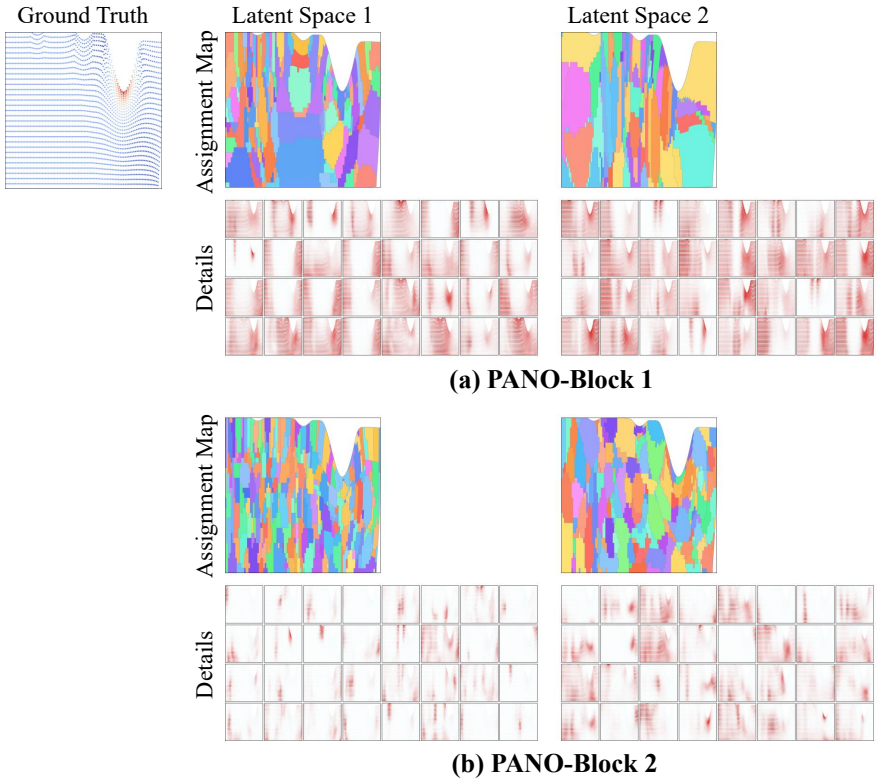
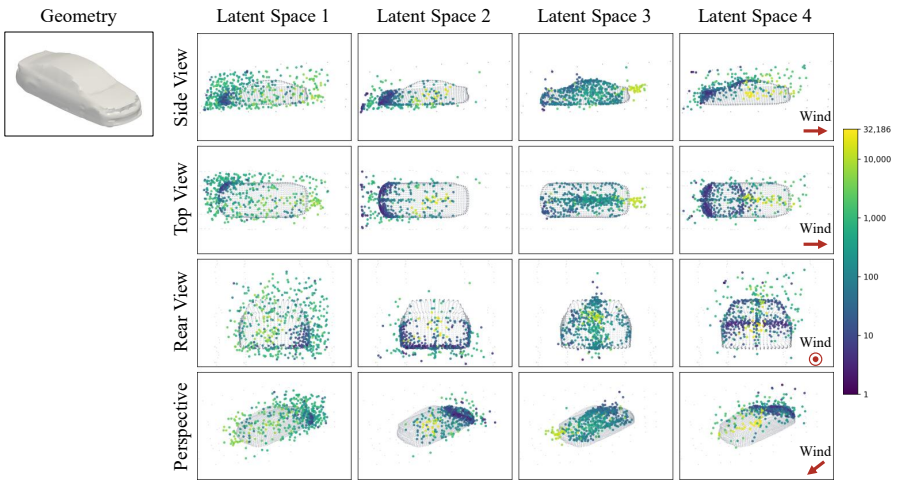
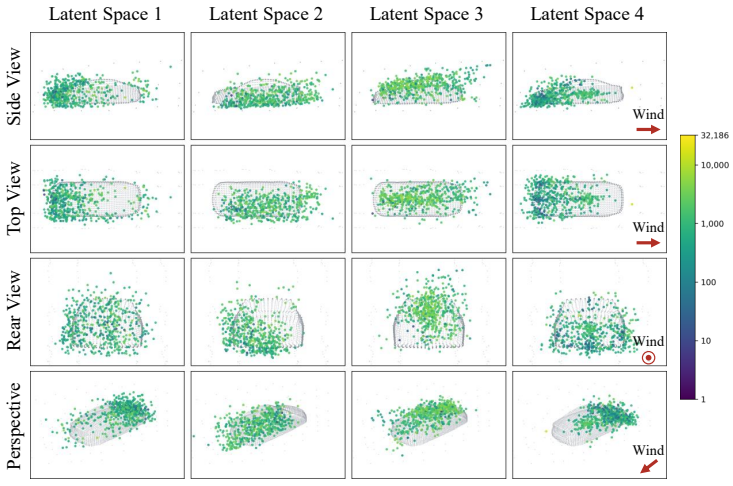


Figure 22: Visualization of PANO-D assignment maps on Plasticity.



(a) PANO-Block 1



(b) PANO-Block 2

Figure 23: Visualization of latent token centers learned by PANO-P on ShapeNetCar.

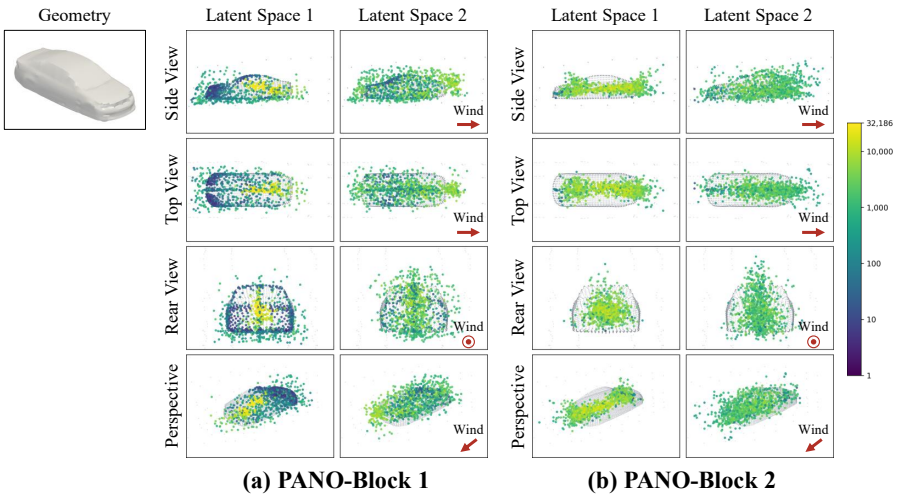


Figure 24: Visualization of latent token centers learned by PANO-D on ShapeNetCar.