SOLIDX: LEARNING PHYSICAL INTERACTIONS WITH STRUCTURED BASES FOR MULTI-SOLID SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Simulating contact-induced deformation in multi-solid systems is critical to many scientific and engineering applications, yet remains challenging for neural operators due to the complex and heterogeneous interactions among multiple physical bodies. Transformer-based neural operators have recently emerged as a powerful paradigm for physical modeling, typically compressing high-dimensional physical fields into a shared latent space to efficiently capture long-range physical dependencies. In multi-solid systems, however, such homogeneous latent representations tend to implicitly entangle distinct intra- and inter-solid couplings within the same set of latent tokens. Such entanglement can limit interaction-specific expressiveness, particularly in data-scarce scenarios. Inspired by structured decomposition principles in classical computational mechanics, we propose **SolidX**, an interaction-structured neural operator that explicitly organizes latent physical bases around distinct interaction patterns in multi-solid systems. By assigning dedicated basis groups to heterogeneous physical interactions, SolidX captures complementary physical behaviors more effectively while retaining linear complexity. Extensive experiments across five multi-solid benchmarks demonstrate that SolidX consistently outperforms strong baselines in prediction accuracy, convergence efficiency, and sample efficiency. Our results indicate that interaction-structured representations provide an effective inductive bias for operator learning, particularly in data-scarce scenarios where high-fidelity data are usually expensive to obtain.

1 INTRODUCTION

Multi-solid systems are ubiquitous in scientific and engineering applications, including manufacturing, robotic manipulation, and biomechanical modeling (Zhao et al., 2026; Li et al., 2019; Yu et al., 2024). Simulating contact-induced deformation in such systems is particularly challenging, as it requires resolving localized contact interactions together with globally coupled deformation responses. Moreover, the multiple participating solids may differ substantially in geometry, constitutive properties, and loading motions, further complicating the coupled physical response. Traditionally, numerical solvers are used to resolve these complex physical processes in multi-solid systems, but repeated simulations under varying configurations are computationally expensive (Tao et al., 2025).

With advances in deep learning, learning-based neural solvers have emerged as appealing surrogates for accelerating physical simulation (Sanchez-Gonzalez et al., 2020; Li et al., 2023c; Tao et al., 2025). Graph-based neural solvers provide an efficient mechanism for modeling local interactions, yet may struggle to capture long-range dependencies over long-horizon physical processes (Li et al., 2020; Pfaff et al., 2021; Linkerhägner et al., 2023). More recently, Transformer-based neural operators have emerged as a powerful paradigm for physical modeling, offering a more direct mechanism for capturing long-range and global dependencies (Wu et al., 2024; Hu et al., 2026; Xiao et al., 2026). Nonetheless, applying full attention to the large number of spatial points in realistic simulations incurs prohibitive computational and memory costs (Wu et al., 2024). Recent approaches therefore compress the original N spatial points into a much smaller set of M *latent slice tokens*, perform physical reasoning in this compact space, and map the resulting representations back to the spatial domain. This *slice–deslice* process[1] can further be interpreted as a form of *linear attention* (Hu et al., 2026), enabling efficient global modeling with linear complexity in the number of spatial points.

[1] From a basis-representation perspective, we also view this process as adaptive *analysis* and *synthesis*.

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.

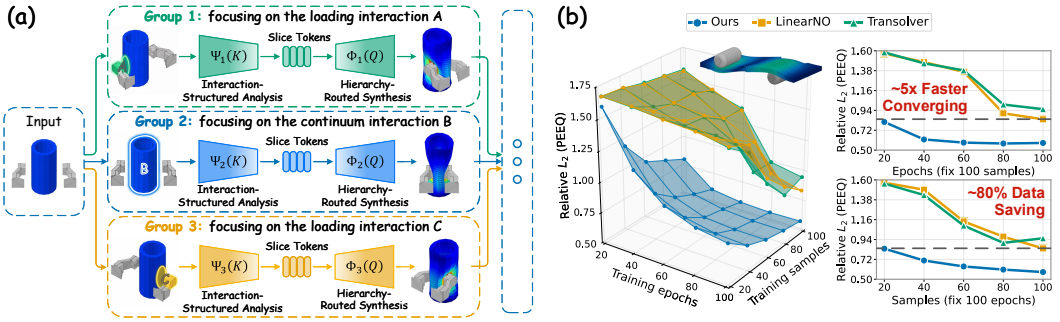


Figure 1: Overview and learning efficiency of SolidX. (a) SolidX explicitly organizes heterogeneous physical interactions into dedicated groups, injecting an interaction-structured inductive bias into the latent representations. (b) In the Bilateral Stamping task, SolidX achieves up to 5× faster convergence and 80% data savings on equivalent plastic strain (PEEQ). More results are provided in Appendix D.5.

Yet, the resulting latent slices or tokens are typically organized as a single homogeneous set shared across different physical interactions. Such a homogeneous organization can become restrictive in multi-solid systems, where different interaction patterns play fundamentally different physical roles. In particular, heterogeneous intra- and inter-solid couplings can become implicitly entangled within the same set of latent bases, ***forcing distinct slice tokens to compete for shared representational directions***. Such entanglement might limit the expressiveness of interaction modeling and reduce data efficiency. This issue is especially significant in data-scarce settings where high-fidelity data are expensive to obtain, and only limited samples are available (Wu et al., 2026; Tang et al., 2026).

Interestingly, classical computational mechanics has long used structured decomposition principles to organize complex physical systems around distinct components and coupling interfaces (Hurty, 1965; Craig & Bampton, 1968; Craig, 1995). Representative methods, such as ***dynamic substructuring*** (Hurty, 1965; Craig, 1995) and ***component-mode synthesis*** (Craig & Bampton, 1968; Krattiger et al., 2019), explicitly decompose multi-component systems into substructures, retaining compact internal modes and interface-related representations for efficient global assembly. This perspective naturally raises a research question: *can neural operators similarly organize their learned physical bases according to the interaction structure of multi-solid systems for more effective physical modeling?*

Inspired by this perspective, we propose ***SolidX***, an interaction-structured neural operator for multi-solid systems. As illustrated in Figure 1(a), SolidX organizes heterogeneous physical interactions into dedicated groups, each equipped with a specialized family of adaptive physical bases. Through *interaction-structured analysis*, these bases extract group-specific physical modes as compact slice tokens. Then, *hierarchy-routed synthesis* adaptively recombines these tokens over the spatial domain to recover the globally coupled response. This structured analysis–synthesis formulation provides an explicit physical inductive bias while retaining linear computational complexity with respect to the number of spatial points (i.e., as a form of *linear attention*). As shown in Figure 1(b), SolidX achieves up to 5× faster convergence and 80% data savings over representative baselines, demonstrating strong learning efficiency in data-scarce regimes. Our key contributions are summarized as follows:

- **Conceptually**, drawing inspiration from structured decomposition principles, we introduce an interaction-structured perspective on operator learning, where learned physical bases are explicitly organized according to distinct physical interaction patterns.
- **Technically**, we develop an interaction-structured analysis module and a hierarchy-routed synthesis module. The former extracts group-specific physical modes through dedicated adaptive bases, while the latter adaptively recombines them into globally coupled responses.
- **Theoretically**, we establish the adaptive-basis and interaction-structured neural-operator interpretations of SolidX. We further demonstrate that correctly specified interaction priors can help preserve the target mapping while reducing unnecessary degrees of freedom.
- **Experimentally**, we conduct extensive evaluations on five multi-solid benchmarks. The results show SolidX's superiority in prediction accuracy, convergence efficiency, and sample efficiency. Representation analyses further show that SolidX exhibits higher representational rank, lower basis redundancy, and clear structure-aligned basis specialization.

2 Related Work

Neural Solvers. Learning-based neural solvers have been widely explored for learning mappings between physical fields and accelerating physical simulation (Li et al., 2026; Tao et al., 2026). Graph-based methods model local physical interactions through message passing over spatial neighborhoods, making them naturally applicable to irregular geometries (Li et al., 2020; Feng et al., 2026; Chen et al., 2026). Yet, capturing long-range dependencies often requires repeated multi-hop propagation, while their step-wise modeling paradigm can also make direct long-horizon prediction challenging. Fourier- and spectral-based methods achieve efficient global modeling (Li et al., 2021; 2023b; Wu et al., 2023), and Transformer-based solvers further exploit attention to capture long-range physical dependencies over general spatial domains (Cao, 2021; Li et al., 2022; 2023a). Recent Transformer-based neural solvers increasingly adopt compact latent representations to avoid expensive pointwise attention. For instance, Transolver (Wu et al., 2024) compresses spatial points into a small set of slices, performs attention among these slices, and then maps them back to the spatial domain. Many subsequent works have adopted a similar slice-based latent representation (Luo et al., 2025; Zhou et al., 2026; Adams et al., 2025; Pei et al., 2026; Xiao et al., 2026). Among them, LinearNO (Hu et al., 2026) further interprets the *slice–deslice* process as a form of linear attention and shows that explicit attention among latent slices is not always necessary. *Despite these advances, existing methods typically organize the latent slices within a homogeneous representation, without explicitly accounting for heterogeneous interaction structures.* SolidX extends this line of work by explicitly organizing adaptive physical bases according to distinct interaction patterns in multi-solid systems.

Learning-Based Multi-Solid Simulation. Recent studies have increasingly explored learning-based solid simulation, with many efforts focusing on rigid-body dynamics (Rubanova et al., 2024; Wei & Fink, 2025; Dou et al., 2026). Several recent works have further extended learning-based simulation to deformable and contact-rich systems (Li et al., 2026; Yu et al., 2024; Chen et al., 2026; Tang et al., 2026). These approaches often adopt task-specific architectures or graph-based formulations to model physical interactions, while efficiently capturing heterogeneous interactions over long-horizon deformation remains challenging. The work most closely related to ours is Unisoma (Tao et al., 2025), which explicitly distinguishes individual solids and models their interactions through object-wise slice composition and attention. *However, its explicit structure remains primarily object-centric: interaction regions are not explicitly encoded into the latent basis construction, while cross-object interactions are still modeled implicitly through attention in the slice space.* Moreover, recent analyses suggest that additional slice-level attention may not always be necessary for effective physical modeling (Hu et al., 2026). In contrast, SolidX directly incorporates physical interaction structure into the basis construction, organizing dedicated adaptive bases around distinct interaction patterns and integrating the resulting physical modes through a unified analysis–synthesis formulation.

Structured Decomposition in Classical Computational Mechanics. Classical computational mechanics has long employed structured decomposition to decompose complex mechanical systems according to their constituent substructures and coupling interfaces (De Klerk et al., 2008; Rixen, 2004; Iollo et al., 2023). Representative approaches, such as dynamic substructuring and component-mode synthesis, retain compact internal modes together with interface-related representations for global system assembly (Hurty, 1965; Craig & Bampton, 1968). Subsequent developments further improve the reduction and coupling of substructures while preserving the distinction between different physical roles within the system (Craig, 1995; Krattiger et al., 2019). This separation reflects a broader principle: *physical behaviors with distinct roles can be represented by dedicated basis families rather than being represented within a single homogeneous reduced basis.* Inspired by this principle, our SolidX adapts structured decomposition to neural operator learning by organizing adaptive physical bases according to distinct interaction patterns in multi-solid systems. In this way, this principle provides the conceptual foundation for our interaction-structured physical basis representation.

3 Method

In this section, we present SolidX for learning multi-solid interactions. We first formulate the operator learning problem in multi-solid systems and review the latent analysis–synthesis formulation underlying recent latent neural operators in Section 3.1. We then motivate our interaction-structured representation and provide an overview of SolidX in Section 3.2. Section 3.3 introduces explicit physical structure modeling, followed by the interaction-structured analysis–synthesis formulation in Section 3.4. Finally, Section 3.5 provides the theoretical analyses of the proposed representation.

3.1 Problem Formulation and Preliminaries

Problem formulation. We aim to learn a neural operator for multi-solid systems that maps the input physical fields to target response fields, primarily deformation fields, optionally accompanied by stress, strain, or other fields. We consider a multi-solid system consisting of S interacting bodies $\mathcal{O} = \{\Omega_i\}_{i=1}^{S}$, including mainly deformable solids and rigid solids with loading motion. Each solid Ω_i is discretized into N_i spatial points, with N points in total, where the n-th point is represented by its coordinate $x_{i,n} \in \mathbb{R}^{d_x}$ and task-dependent physical attribute $a_{i,n} \in \mathbb{R}^{d_a}$. The attributes may encode material properties, body identity, physical states, and boundary or loading information. Given these point-wise inputs, our goal is to learn a direct long-horizon mapping from the initial multi-solid field to the target physical fields (typically final or steady-state) over the deformable solids.

Preliminaries. Transformer-based neural operators have recently emerged as a powerful paradigm for physical modeling, typically compressing high-dimensional physical fields into a shared latent space to efficiently capture long-range physical dependencies. Specifically, Transolver (Wu et al., 2024) adopts a *slice–attention–deslice* pipeline: the N spatial points are first aggregated into a smaller set of M physical slices, where $M \ll N$; self-attention is then performed among these slices; and the updated slice features are mapped back to the original spatial domain. LinearNO (Hu et al., 2026) further shows that the *slice* and *deslice* operations themselves form a linear-attention transformation, while the intermediate slice attention might not be essential. Their formulations can be written as:

$$\mathrm{Transolver}(H) = \phi(Q)\ \mathrm{Attn}\big(\psi(K)^\top V\big)\,, \tag{1}$$

$$\mathrm{LinearNO}(H) = \phi(Q)\,\big(\psi(K)^\top V\big)\,, \tag{2}$$

where $H \in \mathbb{R}^{N\times d_h}$ denotes the point-wise features with hidden dimension d_h. The query, key, and value representations are obtained through learnable linear projections, $Q = HW_Q$, $K = HW_K$, and $V = HW_V$. Transolver uses a shared projection for Q and K, whereas LinearNO learns the two projections independently. The mappings $\phi(\cdot)$ and $\psi(\cdot)$ denote Softmax normalization along the latent dimension M and the spatial dimension N, respectively. In both formulations, $\psi(K)^\top V \in \mathbb{R}^{M\times d_h}$ aggregates the N points into M latent representations and corresponds to the *slice* operation, while $\phi(Q)$ maps these latent representations back to the N points and corresponds to the *deslice* operation.

3.2 Motivation

As reviewed above, recent Transformer-based neural operators typically compress high-dimensional physical fields into a shared latent space to efficiently capture long-range physical dependencies. In multi-solid systems, however, such homogeneous latent representations may implicitly entangle distinct intra- and inter-solid couplings within the same set of latent bases. Drawing inspiration from the structured decomposition principles of dynamic substructuring and component-mode synthesis (Hurty, 1965; Craig & Bampton, 1968; Craig, 1995; Krattiger et al., 2019), which distinguish interface interactions from ordinary intra-component dynamics for efficient physical modeling in classical mechanics, we instead organize physical bases according to the underlying interaction structure.

Motivated by this, we introduce SolidX, which organizes the physical domain into groups with distinct interaction roles and assigns dedicated basis families to each group. To make our motivation concrete, we view the *slice–deslice* transformation through the classical analysis–synthesis framework of basis representations, a standard perspective in wavelet and filter-bank theory (Mallat, 1989; Vetterli & Herley, 1992). Under this view, the *slice* and *deslice* operations naturally correspond to adaptive *analysis* and *synthesis*, respectively. Analysis compresses high-dimensional spatial features into interaction-specific slice tokens through basis functions explicitly associated with interaction groups, while synthesis hierarchically recombines these tokens and maps them back to the spatial domain to recover the global response. In this way, heterogeneous interactions are explicitly separated at the basis level and jointly integrated into the global system response.

Realizing this design also raises three technical challenges. First, potential interaction regions must be identified before explicit modeling. Existing rollout simulators can update contact information from intermediate geometry at each step, e.g., using distance-threshold criteria (Zhao et al., 2026; Sharma & Fink, 2026). Nonetheless, such intermediate states are unavailable for long-horizon prediction. Second, geometric and structural cues may gradually fade as features propagate through deeper layers. Third, after extracting group-specific slice tokens, their information must be recombined across groups to recover the global response. These technical challenges motivate our candidate interface identification, persistent physical conditioning, and hierarchy-routed synthesis, respectively.

3.3 Explicit Physical Structure Modeling

Here, we first identify relevant inter-solid interfaces from the input configuration and organize them into physical groups, explicitly structuring the learned physical bases. Then, we further introduce persistent physical conditioning to preserve structural and geometric cues across network layers. The detailed designs of these two components are introduced below.

Candidate interface identification. For long-horizon prediction, future contact states are unavailable during inference. We therefore identify potential contact interfaces directly from the initial geometry and loading motion. For a potentially interacting rigid–deformable pair (Ω_r, Ω_i), let $x_{r,m}(t)$ denote the loading continuous trajectory of rigid point m over $t \in [0, 1]$, with $x_{r,m}(0)$ denoting its initial position. In practice, we sample the trajectory at $t \in \mathcal{T} \subset [0, 1]$ and perform radius queries on the deformable points using a KD-tree. The candidate interaction interface is defined as:

$$\Gamma_{r,i} = \left\{ x_{i,n} \in X_i \;\middle|\; \min_{m,\, t\in\mathcal{T}} \|x_{i,n} - x_{r,m}(t)\|_2 \le R \right\}, \tag{3}$$

where R denotes the radius threshold. Thus, $\Gamma_{r,i}$ captures the deformable region that may interact with the loading motion of Ω_r and serves as the candidate interface for subsequent grouping. The threshold R can be selected through simple visualization of the identified candidate interfaces on the training set. Our sensitivity analysis further shows that SolidX is robust to the choice of R, while Figure 5 visualizes the candidate interaction interfaces identified under different threshold values.

Physical group construction. Inspired by the structured decomposition principles in classical computational mechanics, we organize the physical domain according to distinct physical roles and interaction patterns. Based on the candidate interfaces identified above, we construct three types of physical groups: ***(1) loading interaction groups***, which focus on candidate interface regions to capture localized loading interactions between rigid and deformable bodies; ***(2) continuum interaction groups***, which cover the entire deformable body to capture body-wide continuum mechanics, where deformation in one region can influence other regions through the continuous material domain; and ***(3) boundary interaction groups***, which focus on boundary constraints that are typically imposed by rigid regions on deformable bodies. The resulting groups are collected into $\mathcal{G}$, with $G = |\mathcal{G}|$, and are used to organize the subsequent interaction-structured basis construction. Some representative examples are shown in Figure 2, with further details provided in Appendix A.

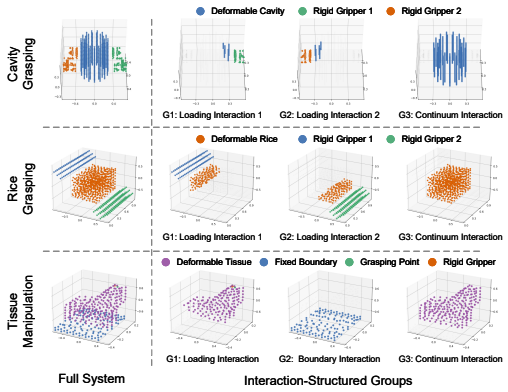


Figure 2: Physical group construction in SolidX.

Persistent physical conditioning. Recent neural operators observe that input-side geometric information may be gradually forgotten as features propagate through deeper layers (Adams et al., 2025). We therefore introduce a lightweight persistent physical conditioning strategy that repeatedly provides geometric and structural cues when constructing the analysis and synthesis bases. Notably, this conditioning is optional, and we conduct extensive ablation studies in Section 4.2. Empirical results show that our lightweight strategy achieves comparable or better performance than more elaborate strategies, while removing this module does not undermine the effectiveness of our SolidX. Implementation details and differences from existing strategies are provided in Appendix A.

3.4 Interaction-Structured Analysis–Synthesis

Here, we detail the core interaction-structured analysis–synthesis module of SolidX. Given the previously constructed physical groups $\mathcal{G}$, SolidX organizes a total of M physical basis functions according to distinct interaction patterns, assigning M_g dedicated basis functions to each group g. We denote the input feature matrix of each block by $H \in \mathbb{R}^{N\times d_h}$. When physical conditioning is enabled, the features used at different stages may differ, and we omit this distinction below for clarity. Notably, we retain the Q, K, and V notation used in the *linear-attention* formulations of current Transformer-based operators in Eqs. (1)–(2), and denote the complete analysis and synthesis basis functions by $\Psi(K)$ and $\Phi(Q)$, respectively, to maintain correspondence with this formulation.

Interaction-structured basis analysis. The analysis stage learns group-specific adaptive physical basis functions that determine how spatial information is analyzed into compact slice tokens. For each physical group g, we first project the input features H into group-specific basis scores K_g and apply group-constrained masked normalization over the spatial dimension,

$$K_g = HW_{K,g} \in \mathbb{R}^{N\times M_g}, \qquad \Psi_g = \mathrm{MaskedSoftmax}_N(K_g; m_g) \in \mathbb{R}^{N\times M_g}, \tag{4}$$

where $W_{K,g} \in \mathbb{R}^{d_h\times M_g}$ is a group-specific projection matrix and $m_g \in \{0,1\}^N$ is the membership mask of group g. Each column of Ψ_g defines an adaptive analysis basis function restricted to the corresponding physical region. The complete set of analysis basis functions is represented as $\Psi(K) = [\Psi_g]_{g=1}^G \in \mathbb{R}^{N\times M}$, where $M = \sum_{g=1}^G M_g$. We next use these functions to extract the corresponding slice tokens. Specifically, for each group g, the point-wise features H are first projected to obtain V_g, which are subsequently aggregated by Ψ_g to extract the slice tokens Z_g:

$$V_g = HW_{V,g} \in \mathbb{R}^{N\times d_h}, \qquad Z_g = \Psi_g^\top V_g \in \mathbb{R}^{M_g\times d_h}, \tag{5}$$

where $W_{V,g} \in \mathbb{R}^{d_h\times d_h}$ is a group-specific projection matrix. The resulting Z_g contains the compact slice tokens extracted from the spatial features of group g. In this way, SolidX maps the original N point-wise features onto M interaction-structured slice tokens. In contrast to conventional *slice* operations, SolidX introduces an explicit physical inductive bias into the analysis stage.

Hierarchy-routed basis synthesis. The synthesis stage learns adaptive physical basis functions that determine how the slice tokens produced by analysis are recombined over the spatial domain. Unlike analysis, synthesis must globally combine information from different physical groups, requiring both group-level relevance and within-group token contributions for each spatial point. We therefore construct the synthesis basis functions hierarchically. The input features are projected into a group-level query $Q^{(1)}$ and token-level queries $Q_g^{(2)}$ for each physical group g,

$$Q^{(1)} = HW_Q^{(1)} \in \mathbb{R}^{N\times G}, \qquad Q_g^{(2)} = HW_{Q,g}^{(2)} \in \mathbb{R}^{N\times M_g}, \tag{6}$$

where $W_Q^{(1)} \in \mathbb{R}^{d_h\times G}$ and $W_{Q,g}^{(2)} \in \mathbb{R}^{d_h\times M_g}$ are learnable projection matrices. The group-level query determines the relevance of physical groups to each spatial point, while the token-level query determines the contribution of tokens within each group. The resulting synthesis basis functions are:

$$\Phi_g = \mathrm{Softmax}_G(Q^{(1)})_{:,g} \odot \mathrm{Softmax}_{M_g}(Q_g^{(2)}) \in \mathbb{R}^{N\times M_g}, \tag{7}$$

where the subscripts G and M_g denote normalization over the group and within-group token dimensions, respectively, and $\odot$ combines the two routing weights element-wise. Each column of Φ_g defines an adaptive synthesis basis function combining group-level selection with within-group token routing. All synthesis basis functions are represented as $\Phi(Q) = [\Phi_g]_{g=1}^G \in \mathbb{R}^{N\times M}$. Then, the spatial features are reconstructed by routing the tokens of each group back to the spatial domain:

$$O = \sum_{g=1}^G \Phi_g Z_g \in \mathbb{R}^{N\times d_h}. \tag{8}$$

In this way, SolidX hierarchically maps the M interaction-structured slice tokens back to the original N physical points, allowing information extracted from different interaction groups to propagate across the spatial domain. As further validated by the representation analyses (RQ3) in Section 4.2, our proposed design effectively preserves the injected interaction structure. As our prediction targets lie on deformable solids, the rigid-body points keep their features fixed across layers to avoid unnecessary computation while preserving their physical influence. In practice, each analysis–synthesis module is followed by a point-wise feed-forward network, and L such blocks are stacked to form SolidX. For clarity, we summarize the detailed procedures of our SolidX in Algorithm 1.

3.5 Theoretical Analyses

Here, we theoretically characterize SolidX through four key properties. For clarity, we summarize the conclusions here, while the formal propositions and proofs are provided in Appendices B and C.

Remark 1 (The analysis and synthesis functions form adaptive bases). The learned analysis and synthesis function families, $\Psi(K)$ and $\Phi(Q)$, each span an input-adaptive subspace, with a maximal linearly independent subset forming a basis of the corresponding subspace. When either family has full column rank, all its learned functions are linearly independent and themselves form a basis. Empirical analyses and comparisons of these ranks are further provided in Section 4.2. We provide a formal statement and complete proof of this property in Proposition 1 of Appendix B.

Remark 2 (SolidX is an interaction-structured neural operator). Neural operators are often formulated as learnable mappings between function spaces through integral kernels (Kovachki et al., 2023; Wu et al., 2024). Following this perspective, the analysis–synthesis process in SolidX admits an input-adaptive integral-operator form, $o(x) = \int_\Omega \mathcal{K}(x,\xi)H(\xi)\,d\mu(\xi)$, where $H(\xi)$ denotes the source feature field, $o(x)$ the output at query location x, and $\mathcal{K}(x,\xi)$ the input-adaptive kernel. In SolidX, this kernel is explicitly organized according to the underlying physical interaction structure. We provide a formal statement and detailed proof of this property in Proposition 2 of Appendix B.

Remark 3 (Interaction priors reduce unnecessary degrees of freedom). By introducing a correctly specified interaction prior, SolidX organizes the learned bases according to the underlying physical interaction structure, thereby preserving the target mapping while reducing unnecessary degrees of freedom. We provide a detailed proof of this property in Proposition 3 of Appendix B.

Remark 4 (SolidX retains linear complexity in the number of spatial points). The computational complexity of each analysis–synthesis module in SolidX is $\mathcal{O}(Nd_h(d_hG + M))$, which is linear with respect to the number of spatial points N. We provide detailed analyses in Appendix C.

4 Experimental Evaluations

This section comprehensively evaluates SolidX from four perspectives, including overall predictive performance, learning efficiency, interaction-structured basis organization, and the contribution of individual components. Specifically, we investigate the following Research Questions (RQs):

RQ1: How does SolidX perform across diverse benchmarks compared with existing methods?
RQ2: How efficiently does SolidX learn from limited data and converge during training?
RQ3: How well do the learned physical bases reflect the underlying interaction structure?
RQ4: How does each key component contribute to the overall performance of SolidX?

4.1 Experimental Setup

Datasets. We evaluate SolidX on five 3D multi-solid datasets, including Plate Deforming (Pfaff et al., 2021), Cavity Grasping (Linkerhägner et al., 2023), Rice Grasping (Li et al., 2019), Tissue Manipulation (Linkerhägner et al., 2023), and Bilateral Stamping (Tao et al., 2025). Plate Deforming evaluates both geometry and stress, while Cavity Grasping, Rice Grasping, and Tissue Manipulation focus on geometry prediction. Bilateral Stamping contains two deformable solids, a metal plate and a rubber insert, and evaluates geometry, stress, and equivalent plastic strain (PEEQ) for both materials. Following industrial practice (Wu et al., 2026; Ansys Inc., 2026), we use approximately 100 training samples and 50 test samples for each benchmark, reflecting the limited and costly data available in real-world simulations. Detailed dataset descriptions and settings are provided in Appendix D.

Baselines. We compare SolidX with a broad range of learning-based solvers, including graph-based models, i.e., GraphSAGE (Hamilton et al., 2017), MGN (Pfaff et al., 2021), HOOD (Grigorev et al., 2023), and HCMT (Yu et al., 2024); typical neural operators, i.e., GNO (Li et al., 2020), Geo-FNO (Li et al., 2023b), LSM (Wu et al., 2023), and GINO (Li et al., 2023c); and recent Transformer-based neural operators, i.e., Transolver (Wu et al., 2024), Transolver++ (Luo et al., 2025), Unisoma (Tao et al., 2025), LinearNO (Hu et al., 2026), HNO (Pei et al., 2026), and FuncAttn (Xiao et al., 2026). For a fair comparison, we use the same training hyperparameters across all methods. For comparable Transformer-based baselines, we further match the number of latent slices, layers, and attention heads whenever applicable. Detailed baseline and implementation settings are provided in Appendix D.

4.2 Main Results and Analyses

Overall performance (RQ1). We compare SolidX with all baselines across all five multi-solid benchmarks with complex 3D deformation. The results are summarized in Table 1, with detailed prediction and error visualizations in Figs. 7–13 of Appendix D. From these results, we can observe:

(1) Consistent improvements across diverse settings. As shown in Table 1, SolidX achieves the best performance on all 11 outputs across geometry, stress, and PEEQ, yielding 9.70%–33.73% improvements over the second-best methods. For Bilateral Stamping, M and R denote the metal plate and rubber insert, respectively. These consistent gains across physical quantities, interaction settings, and materials demonstrate the effectiveness of interaction-structured physical bases.

Table 1: Performance comparison on five multi-solid datasets. Relative L2 is recorded. A smaller value indicates better performance. The best result is in **bold** and the second best is underlined. For Bilateral Stamping, the two columns under each metric correspond to the metal and rubber, respectively. The improvement indicates the relative reduction over the second-best result. We repeat experiments for SolidX and the second-best methods and report the standard deviations in Table 5.

	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
	Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
GraphSAGE (2017)	0.1002	0.6073	0.1111	0.9603	0.0847	0.0359	0.0352	0.0446	0.3039	0.7035	0.8607
MGN (2021)	0.1314	0.7252	0.1261	0.9815	0.2435	0.0319	0.0317	0.0426	0.2753	0.6454	0.6728
HOOD (2023)	0.0932	0.6179	0.1151	0.9569	0.1098	0.0373	0.0365	0.0443	0.3107	0.7338	0.8900
HCMT (2024)	0.0960	0.5660	0.1062	0.8072	0.0686	0.0351	0.0351	0.0444	0.3055	0.7104	0.8731
GNO (2020)	0.0944	0.6321	0.1133	0.8752	0.0875	0.0322	0.0319	0.0435	0.3011	0.7121	0.8680
Geo-FNO (2023)	0.1056	0.6585	0.0594	0.7187	0.0590	0.0287	0.0281	0.0454	0.3006	0.6930	0.8611
LSM (2023)	0.1392	0.7958	0.1400	0.9829	0.1675	0.0491	0.0367	0.0496	0.3115	0.7789	0.8847
GINO (2023)	0.1098	0.6761	0.0601	0.7425	0.0667	0.0318	0.0316	0.0436	0.2971	0.7028	0.8630
Transolver (2024)	0.0792	0.5977	0.0376	0.8437	0.0433	0.0138	0.0137	0.0342	0.1828	0.4266	0.5270
Transolver++ (2025)	0.0863	0.5153	0.0361	0.9059	0.2727	0.0129	0.0131	0.0347	0.1768	0.4220	0.4748
Unisoma (2025)	0.0986	0.5790	0.1015	0.8976	0.0453	0.0099	0.0094	0.0337	0.1515	0.4097	0.3707
LinearNO (2026)	0.0907	0.5239	0.0346	0.9310	0.0843	0.0131	0.0129	0.0343	0.1713	0.4056	0.4357
HNO (2026)	0.0942	0.5895	0.0350	0.8904	0.0577	0.0126	0.0123	0.0342	0.1675	0.4078	0.4234
FuncAttn (2026)	0.0639	0.4648	0.0385	0.8841	0.0462	0.0140	0.0137	0.0345	0.1833	0.4521	0.5277
SolidX (Ours)	**0.0443**	**0.3437**	**0.0273**	**0.5478**	**0.0287**	**0.0077**	**0.0068**	**0.0304**	**0.1308**	**0.3214**	**0.2580**
Improvement ↑	30.67%	26.06%	21.09%	23.78%	33.73%	22.07%	27.13%	9.70%	13.64%	20.76%	30.40%

(2) Better recovery of localized interaction responses. Visualizations in Figs. 7–13 show that SolidX better recovers localized high-response regions and fine-grained spatial variations than Transolver and LinearNO, whose predictions are often smoother. The error maps also show lower errors for SolidX, indicating more faithful modeling of heterogeneous intra- and inter-solid couplings.

(3) Comparable computation and memory costs. We compare the parameter count, computation cost, and GPU memory usage with representative baselines in Table 10. With comparable parameter counts, SolidX achieves the lowest GFLOPs on 4 datasets and GPU memory on 3 datasets. The reduced cost mainly comes from candidate interface identification, which filters out rigid regions irrelevant to potential contacts, and from keeping rigid features fixed across layers, avoiding unnecessary updates.

Learning efficiency (RQ2). Here, we evaluate the convergence and data efficiency of SolidX. Results under different epoch and sample budgets are shown in Figs. 14–21. These results show:

(1) Faster convergence. SolidX reaches comparable or better performance with substantially fewer training epochs than Transolver and LinearNO, achieving approximately $1.6\times$–$5\times$ faster convergence across datasets and physical quantities, demonstrating more efficient optimization across settings.

(2) Lower data requirements. SolidX maintains strong performance with substantially fewer training samples, corresponding to approximately 40%–80% data savings across all datasets and demonstrating improved data efficiency for data-scarce multi-solid applications.

Interaction-structured basis organization (RQ3). To examine the learned basis organization, we measure the rank of the *implicit linear-attention map* $\Phi(Q)\Psi^\top(K)$, which reflects the number of linearly independent directions preserved by the analysis–synthesis mapping and the redundancy of the learned latent representation (Hu et al., 2026). We further compare the similarities of the analysis and synthesis bases to examine their structural patterns. Results on Cavity Grasping are summarized in Fig. 3, while detailed results for the other datasets are provided in Figs. 22–33. We additionally visualize the learned basis assignments in Figs. 36–51. From these results, we can observe:

(1) Higher rank with less basis redundancy. We analyze the rank of $\Phi(Q)\Psi^\top(K)$ via singular value decomposition. As shown in Fig. 3(a), under the relative threshold of 10^{-6}, SolidX reaches an effective-rank ratio of nearly 100% across all examined layers, approaching full rank and substantially exceeding the baselines. Fig. 3(b) further shows the distributions of the relative singular values, where SolidX preserves a larger number of directions with non-negligible singular values. This advantage remains consistent across a wide range of relative thresholds in Fig. 3(c). Together, these results indicate less linear redundancy and more effective utilization of the available basis capacity in SolidX.

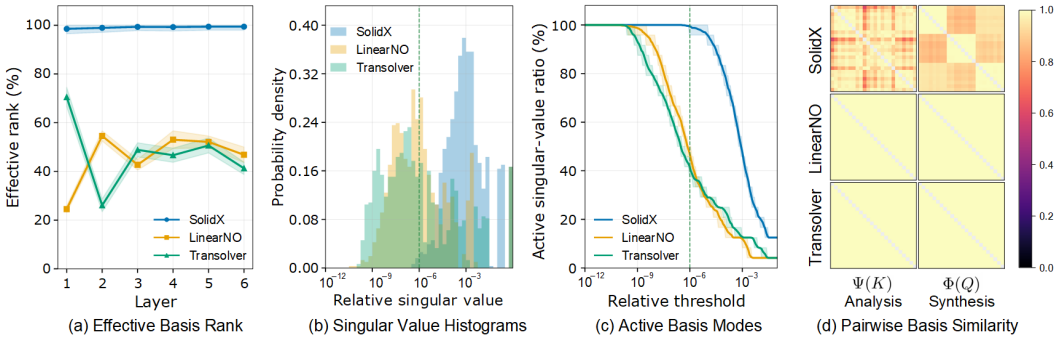


Figure 3: Basis analyses for our SolidX and baselines on Cavity Grasping.

(2) Interaction-structured basis specialization. As shown in Fig. 3(d), we compare the similarities induced by the learned analysis and synthesis bases. LinearNO and Transolver exhibit uniformly high similarities, indicating largely homogeneous basis representations and substantial redundancy. In contrast, SolidX shows clear differentiation on both the analysis and synthesis sides, together with a pronounced group-wise similarity structure: bases within the same physical group remain more similar, whereas those across different groups are more distinct, indicating interaction-structured specialization. The basis visualizations in Figs. 36–51 further confirm this structure-aligned specialization, with different groups focusing on distinct interaction and component-level regions. In contrast, the baseline methods tend to produce highly similar weights across different bases.

Ablation analyses (RQ4). We conduct extensive ablations to examine the main design choices of SolidX, including latent attention, physical conditioning, input physical cues, and the key structural components. The results are summarized in Tables 8–9. From these results, we can observe:

(1) Additional attention is not essential. Following Transolver (Wu et al., 2024), we introduce self-attention over the extracted slice tokens, either globally across all tokens or explicitly across physical groups. These variants yield no consistent improvements across datasets and metrics, indicating that additional attention among the extracted physical modes is not essential for SolidX.

(2) Simple physical conditioning is sufficient. We compare our additive physical conditioning with more elaborate gated and FiLM schemes, and find that our additive strategy achieves comparable or better performance. We also examine different injection pathways and find that conditioning both the query and key pathways yields the most balanced performance, supporting our conditioning design.

(3) Physical cues are beneficial for conditioning. We further ablate the object-identity and relative-geometric cues used in physical conditioning. Incorporating these cues generally improves performance, while their combination helps distinguish heterogeneous physical roles and interactions.

(4) Structural components provide complementary benefits. We progressively remove persistent physical conditioning, rigid-state preservation, hierarchy-routed synthesis, and explicit interaction grouping, with the final configuration corresponding to LinearNO (Hu et al., 2026). Performance generally degrades along this progression, with the largest drop occurring when the explicit interaction structure is removed. This shows that the components provide complementary benefits, while interaction-structured organization remains the key distinction from homogeneous neural operators.

5 Conclusion and Discussion

In this work, we introduced SolidX, an interaction-structured neural operator for multi-solid systems. Inspired by structured decomposition principles in classical computational mechanics, SolidX explicitly organizes adaptive physical bases according to distinct interaction patterns. Theoretical and experimental analyses further support the effectiveness of our interaction-structured basis organization. Our current study primarily focuses on rigid-to-deformable loading scenarios, where the loading motions of rigid bodies drive the deformation of deformable solids, a setting widely encountered in manipulation and engineering simulation. In the current setting, the loading motion and boundary conditions are provided as known inputs, consistent with standard simulation settings. Looking forward, we aim to extend SolidX toward more general interaction settings with dynamically evolving interaction structures, and ultimately beyond multi-solid mechanics toward multiphysics systems involving heterogeneous physical processes and couplings among different fields.

AI USE STATEMENT

Generative AI tools were used in the preparation of this work for mainly three purposes: (1) assisting with language polishing and improving the clarity and readability of the manuscript; (2) assisting with theoretical analysis by providing feedback on mathematical derivations and proof strategies; (3) assisting with the software implementation of the proposed method and related experimental code. All AI-assisted outputs were critically reviewed and revised by the authors. The mathematical claims, derivations, and proofs were independently verified by the authors, and AI-assisted code was reviewed, tested, and debugged before being used in the reported experiments. The authors take full responsibility for the final manuscript, theoretical claims, implementation, and experimental results.

ETHICS STATEMENT

In this paper, we propose a new neural solver that explicitly organizes latent physical bases around distinct interaction patterns in multi-solid systems. Our work primarily focuses on learning-based physical modeling for scientific and engineering applications. It does not involve human participants, personally identifiable information, or decision-making about individuals. The experiments are conducted on benchmark datasets describing physical systems. We are committed to ensuring that ethical considerations are taken into account throughout our research. While our work may have potential broader societal impacts through its applications in scientific and engineering domains, we do not identify any specific ethical concerns that require particular emphasis in this work.

REPRODUCIBILITY STATEMENT

To ensure the full reproducibility of our research, we have made extensive efforts to provide all the necessary details. We provide detailed procedures of our proposed approach in our paper, including comprehensive descriptions of the algorithmic steps and design choices. For our theoretical claims, the complete proofs and analyses are included in the Appendix. The experimental benchmarks, evaluation metrics, data processing procedures, dataset splits, model configurations, training hyperparameters, and baseline settings are documented in the experimental section and Appendix. If accepted at ICLR 2027, our source code and implementation will be made publicly available as open source, allowing the broader research community to reproduce our results and build upon our work. Should reviewers request access to the implementation for their assessment, we are also willing to provide an anonymized version of the source code during the rebuttal period.

REFERENCES

Corey Adams, Rishikesh Ranade, Ram Cherukuri, and Sanjay Choudhry. Geotransolver: Learning physics on irregular domains using multi-scale geometry aware physics attention transformer, 2025. URL https://arxiv.org/abs/2512.20399.

Ansys Inc. Ansys SimAI, 2026. URL https://www.ansys.com/products/simai. Accessed: 2026-01-06.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d0921d442ee91b896ad95059d13df618-Abstract.html.

Qianyi Chen, Tianrun Gao, Chenbo Jiang, and Tailin Wu. EqCollide: Equivariant and collision-aware deformable objects neural simulator. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, New York, NY, USA, 2026. Association for Computing Machinery. doi: 10.1145/3770855.3818862. URL https://doi.org/10.1145/3770855.3818862.

Roy R. Craig, Jr. Substructure methods in vibration. *Journal of Vibration and Acoustics*, 117(B):207–213, June 1995. doi: 10.1115/1.2838665. URL https://doi.org/10.1115/1.2838665.

Roy R. Craig, Jr. and Mervyn C. C. Bampton. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1313–1319, July 1968. doi: 10.2514/3.4741. URL https://doi.org/10.2514/3.4741.

Dennis De Klerk, Daniel J Rixen, and Sven N Voormeeren. General framework for dynamic substructuring: history, review and classification of techniques. *AIAA journal*, 46(5):1169–1181, 2008.

Zhiyang Dou, Minghao Guo, Haixu Wu, Doug Roble, Tuur Stuyck, and Wojciech Matusik. RigidFormer: Learning rigid dynamics using transformers, 2026. URL https://arxiv.org/abs/2605.09196.

Zhe Feng, Shilong Tao, Haonan Sun, Shaohan Chen, Zhanxing Zhu, and Yunhuai Liu. MAVEN: A mesh-aware volumetric encoding network for simulating 3D flexible deformation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Artur Grigorev, Michael J. Black, and Otmar Hilliges. HOOD: Hierarchical graphs for generalized modelling of clothing dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2023.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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. doi: 10.1609/aaai.v40i1.37003.

Walter C. Hurty. Dynamic analysis of structural systems using component modes. *AIAA Journal*, 3 (4):678–685, 1965. doi: 10.2514/3.2947.

Angelo Iollo, Giulia Sambataro, and Tommaso Taddei. A one-shot overlapping schwarz method for component-based model reduction: application to nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 404:115786, 2023.

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.

Dimitri Krattiger, Long Wu, Martin Zacharczuk, Martin Buck, Robert J. Kuether, Matthew S. Allen, Paolo Tiso, and Matthew R. W. Brake. Interface reduction for Hurty/Craig–Bampton substructured models: Review and improvements. *Mechanical Systems and Signal Processing*, 114:579–603, 2019. doi: 10.1016/j.ymssp.2018.05.031.

Yifei Li, Haixu Wu, Zeyi Xu, Tuur Stuyck, and Wojciech Matusik. Neural modular physics for elastic simulation. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. doi: 10.48550/arXiv.2512.15083. URL https://arxiv.org/abs/2512.15083.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJgbSn09Ym.

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

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for PDE surrogate modeling. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/590daf74f99ee85df3d8c007df9c8187-Abstract-Conference.html.

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*, 2020. URL https://arxiv.org/abs/2003.03485.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

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. URL https://jmlr.org/papers/v24/23-0064.html.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023c. doi: 10.52202/075280-1556. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/70518ea42831f02afc3a2828993935ad-Abstract-Conference.html.

Jonas Linkerhägner, Niklas Freymuth, Paul Maria Scheikl, Franziska Mathis-Ullrich, and Gerhard Neumann. Grounding graph network simulators using physical sensor observations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jsZsEd8VEY.

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41432–41449. PMLR, 2025. URL https://proceedings.mlr.press/v267/luo25o.html.

Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. doi: 10.1109/34.192463.

Jieyuan Pei, Zhuoxuan Li, Wei Li, Haobo Zhang, Jiawei Jiang, and Jianwei Zheng. Hyperbolic neural operator. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://icml.cc/virtual/2026/poster/65554.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Daniel J Rixen. A dual craig–bampton method for dynamic substructuring. *Journal of Computational and applied mathematics*, 168(1-2):383–391, 2004.

Yulia Rubanova, Tatiana Lopez-Guevara, Kelsey R Allen, William F Whitney, Kimberly Stachenfeld, and Tobias Pfaff. Learning rigid-body simulators over implicit shapes for large-scale scenes and vision. *Advances in Neural Information Processing Systems*, 37:125809–125838, 2024.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020. URL https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Vinay Sharma and Olga Fink. A physics-informed graph neural network conserving linear and angular momentum for dynamical systems. *Nature Communications*, 17:1045, 2026. doi: 10.1038/s41467-025-67802-5.

Jianheng Tang, Shilong Tao, Zhe Feng, Haonan Sun, Menglu Wang, Zhanxing Zhu, and Yunhuai Liu. Fildeep: Learning large deformations of elastic-plastic solids with multi-fidelity data. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2458–2469, 2026.

Shilong Tao, Zhe Feng, Haonan Sun, Zhanxing Zhu, and Yunhuai Liu. Unisoma: A unified transformer-based solver for multi-solid systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59132–59153. PMLR, 2025. URL https://proceedings.mlr.press/v267/tao25b.html.

Shilong Tao, Zhe Feng, Shaohan Chen, Weichen Zhang, Zhanxing Zhu, and Yunhuai Liu. Neural latent arbitrary lagrangian-eulerian grids for fluid-solid interaction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jKeOsMdMe5.

Martin Vetterli and Cormac Herley. Wavelets and filter banks: Theory and design. *IEEE Transactions on Signal Processing*, 40(9):2207–2232, 1992. doi: 10.1109/78.157221.

Amaury Wei and Olga Fink. Integrating physics and topology in neural networks for learning rigid body dynamics. *Nature Communications*, 16(1):6867, 2025.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional PDEs with latent spectral models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37417–37438. PMLR, 2023. URL https://proceedings.mlr.press/v202/wu23f.html.

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

Haixu Wu, Minghao Guo, Zongyi Li, Zhiyang Dou, Mingsheng Long, Kaiming He, and Wojciech Matusik. GeoPT: Scaling physics simulation via lifted geometric pre-training. *arXiv preprint arXiv:2602.20399*, 2026.

Jiefang Xiao, Maolin Gao, Simon Weber, Guandao Yang, and Daniel Cremers. Functional attention: From pairwise affinities to functional correspondences. *arXiv preprint arXiv:2605.31559*, 2026.

Youn-Yeol Yu, Jeongwhan Choi, Woojin Cho, Kookjin Lee, Nayong Kim, Kiseok Chang, Chang-Seung Woo, Ilho Kim, Seok-Woo Lee, Joon-Young Yang, Sooyoung Yoon, and Noseong Park. Learning flexible body collision dynamics with hierarchical contact mesh transformer. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.12467.

Yingxue Zhao, Qianyi Chen, Haoran Li, Haosu Zhou, Hamid Reza Attar, Tobias Pfaff, Tailin Wu, and Nan Li. Recurrent u-net-based graph neural network (rugnn) for accurate deformation predictions in sheet material forming. *Advanced Engineering Informatics*, 69:104021, 2026. doi: 10.1016/j.aei.2025.104021.

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. In *Forty-third International Conference on Machine Learning*, 2026.

A Detailed Implementation of SolidX

Algorithm 1 Detailed Procedure of SolidX

Require: Multi-solid coordinates X, physical attributes A
Ensure: Predicted physical fields $\widehat{Y}$

// Physical structure construction
Identify candidate interaction regions following Section 3.3;
Construct physical groups $\mathcal{G}$ and membership masks $\{m_g\}_{g\in\mathcal{G}}$;
Construct the deformable-point update mask $u \in \{0,1\}^N$, where $u_n = 1$ for deformable points and 0 otherwise;
// Input encoding
Encode (X, A) into the initial point-wise features H^0;
Construct the persistent physical context $C \in \mathbb{R}^{N\times d_h}$ from $(X, A, \mathcal{G})$;
for $\ell = 0, \ldots, L-1$ **do**
  Normalize the current features: $\bar{H}^\ell \leftarrow \mathrm{LN}(H^\ell)$;
  // Interaction-structured basis analysis
  for each physical group $g \in \mathcal{G}$ **do**
    Construct Ψ_g^ℓ from $\bar{H}^\ell + C$ and m_g using Eq. (4);
    Construct V_g^ℓ from $\bar{H}^\ell$ and extract the slice tokens Z_g^ℓ using Eq. (5);
  end for
  // Hierarchy-routed basis synthesis
  Construct $\{\Phi_g^\ell\}_{g\in\mathcal{G}}$ from $\bar{H}^\ell + C$ using Eqs. (6)–(7);
  Reconstruct the point-wise features: $O^\ell \leftarrow \sum_{g\in\mathcal{G}} \Phi_g^\ell Z_g^\ell$;
  // Deformable-state update
  Apply the residual update: $\widetilde{H}^\ell \leftarrow H^\ell + u \odot O^\ell$;
  Apply the point-wise feed-forward update: $H^{\ell+1} \leftarrow \widetilde{H}^\ell + u \odot \mathrm{FFN}_\ell\big(\mathrm{LN}(\widetilde{H}^\ell)\big)$;
end for
Decode H^L over the deformable solids to obtain $\widehat{Y}$;
return Predicted physical fields $\widehat{Y}$;

Here, we provide detailed implementation descriptions of SolidX. Specifically, we first summarize the complete forward procedure in Algorithm 1 and describe the overall network workflow. We then elaborate on the construction of physical groups and membership masks, the persistent physical conditioning mechanism, and the allocation of physical bases across different groups. We further discuss how the interaction-region identification can be adapted to scenarios with dynamically evolving interactions. The detailed implementation is presented below.

Overall procedure. For clarity, Algorithm 1 summarizes the complete forward procedure. Here, we use $\bar{H}^\ell$ to denote the normalized features within the ℓ-th block, while this distinction is omitted in the main text for notational simplicity. Given the multi-solid coordinates X and physical attributes A, SolidX first identifies potential interaction regions and organizes the physical domain into groups $\mathcal{G}$ with corresponding membership masks. The input is then encoded into H^0, while a physical context C is constructed once from the initial configuration and shared across all L SolidX blocks. Within each block, $\bar{H}^\ell + C$ is used to construct the adaptive analysis and synthesis bases, whereas the value features are obtained directly from $\bar{H}^\ell$. The resulting group-specific slice tokens are hierarchically synthesized into O^ℓ. The deformable-point mask u then gates both the residual and feed-forward updates, so that only deformable points are updated while rigid or loading-motion points remain unchanged. Finally, the representation H^L is decoded into the target physical fields.

Physical group construction. Following the formulation in the main text, we organize the domain into three types of groups: *loading interaction groups*, *continuum interaction groups*, and *boundary interaction groups*. For each potentially interacting rigid–deformable pair, the candidate interface region identified from the input configuration forms a *loading interaction group*, which captures the localized region where loading interactions may occur between the rigid and deformable bodies. For each deformable body, all spatial points belonging to the body form a *continuum interaction group*, which captures its body-wide continuum mechanics and allows deformation in one region to influence other regions through the continuous material domain. Rigid regions that impose boundary

constraints on deformable bodies are further organized into *boundary interaction groups*, explicitly representing the externally imposed constraints involved in the physical interaction.

For example, when a loading rigid tool interacts with a deformable plate, the candidate interface region on the plate and the rigid tool forms a loading interaction group, while the entire deformable plate forms a continuum interaction group. If rigid boundaries are present, they are further organized as boundary interaction groups. A spatial point may belong to multiple groups when it participates in multiple physical roles. The resulting groups are collected as $\mathcal{G}$. For each group g, we construct a mask $m_g \in \{0,1\}^N$, where $m_{g,n} = 1$ indicates that point n belongs to g. The collection $\{m_g\}_{g\in\mathcal{G}}$ therefore provides the explicit grouping used to organize the subsequent group-specific bases.

Persistent physical conditioning. To preserve input-side geometric and structural information across deep blocks, we construct a persistent physical context $C \in \mathbb{R}^{N\times d_h}$ once from $(X, A, \mathcal{G})$ and share it across all SolidX blocks. The context encodes physical cues such as coordinates, solid identity, group membership, loading motion, and relative geometric relations.

For the ℓ-th block, we first normalize the current features as $\bar{H}^\ell = \mathrm{LN}(H^\ell)$ and add C only when constructing the analysis and synthesis bases:

$$K_g^\ell = (\bar{H}^\ell + C)W_{K,g},\ Q^{(1),\ell} = (\bar{H}^\ell + C)W_Q^{(1)},\ Q_g^{(2),\ell} = (\bar{H}^\ell + C)W_{Q,g}^{(2)},\ V_g^\ell = \bar{H}^\ell W_{V,g}. \quad (9)$$

Thus, the persistent context affects the assignment of spatial information to the adaptive analysis and synthesis bases without directly modifying the value features aggregated into the slice tokens Z_g^ℓ.

Our conditioning differs from existing geometry-reinjection strategies in two aspects. First, rather than introducing an additional attention pathway (Adams et al., 2025), we find that simple additive injection before the query and key projections is sufficient to effectively preserve physical information, while introducing substantially less architectural overhead. Second, beyond the geometric information emphasized by existing methods, we find that other physical cues, such as solid identity, can also be gradually lost across deep layers, and persistently injecting these cues together with geometry further improves predictive performance. We further ablate different conditioning schemes, physical cues, and injection locations in Section 4.2, further supporting the effectiveness of our conditioning design.

Basis-function allocation. Given the G physical groups, the total budget of M basis functions needs to be distributed among different groups. In our implementation, we adopt a simple uniform allocation strategy, and our experiments show that this straightforward allocation already yields substantial performance improvements across diverse systems. More generally, SolidX naturally supports flexible allocation of basis functions across different physical groups, for example according to their spatial size, interaction complexity, or available physical prior knowledge.

Interface identification for evolving interactions. In the industrial forming and manipulation scenarios considered in this work, the dominant external loading is typically imposed by rigid tools with loading motions. Since these motions are available at inference time, their swept trajectories provide reliable geometric cues for identifying potential interaction regions before direct long-horizon prediction. Such interaction regions can then be used to organize the physical groups for subsequent basis construction. In other interaction settings, the bodies that drive the interaction may themselves undergo deformation or state-dependent motion. Their future displacements are therefore determined by the evolving system dynamics and cannot be accurately specified in advance. In this case, the complete interaction structure cannot be reliably constructed once from the initial state using the trajectory-based procedure adopted above. Autoregressive rollout provides a natural alternative, since the interactions can be updated from the current system state at each prediction step. Existing simulators have adopted dynamically constructed interaction graphs (Li et al., 2019; Sanchez-Gonzalez et al., 2020), world-space or proximity-based contact connections (Pfaff et al., 2021), and contact-aware mesh structures for flexible-body collisions (Yu et al., 2024).

These interaction-identification mechanisms can be directly integrated into SolidX in place of the trajectory-based construction. SolidX only requires the identified interaction regions and their point-wise memberships to form the corresponding physical groups. Once these memberships are obtained, the remaining pipeline is unchanged: the physical groups are used to organize the interaction-structured basis families, followed by the same basis analysis and hierarchical synthesis. This allows dynamically identified interactions during rollout to be incorporated into SolidX without changing its subsequent basis formulation.

B THEORETICAL ANALYSES

In this appendix, we provide detailed proofs for the three main theoretical properties established in Section 3.5. For convenience, throughout this appendix, we use Ψ and Φ to denote the analysis and synthesis bases $\Psi(K)$ and $\Phi(Q)$, respectively. Specifically, Appendix B.1 justifies the *interaction-structured analysis–synthesis basis* interpretation of our complete adaptive analysis and synthesis representations. Appendix B.2 proves that the resulting analysis–synthesis transformation can be formulated as an input-adaptive, interaction-structured neural operator. Finally, Appendix B.3 analyzes an idealized setting and shows that, under a correct interaction prior, explicit structural modeling reduces unnecessary degrees of freedom and thus provides a theoretical explanation for improved data efficiency. The detailed statements and proofs are presented below.

B.1 PROPOSITION 1: INTERACTION-STRUCTURED ANALYSIS–SYNTHESIS BASES

In this subsection, we provide a formal justification for the *interaction-structured analysis–synthesis basis* interpretation introduced in Section 3.5. Our argument proceeds in three parts. First, we show that the complete adaptive analysis and synthesis functions span finite-dimensional global subspaces, with non-redundant subsets forming bases of the corresponding subspaces. Second, we show that masked analysis restricts each analysis function to its corresponding physical group, establishing the interaction-structured property of the global representation. Finally, in the proof of Proposition 1, we characterize the complete analysis–synthesis representation and its induced interaction matrix.

For clarity, we first restate the notation used throughout this subsection. Let $\Psi \in \mathbb{R}^{N\times M}$ and $\Phi \in \mathbb{R}^{N\times M}$ denote the complete adaptive analysis and synthesis function families learned by SolidX, respectively, where M is the total number of physical bases. The columns of Ψ determine how spatial features are analyzed into slice tokens, while the columns of Φ determine how these slice tokens are synthesized back to the spatial domain. We define the corresponding global learned subspaces as $\mathcal{U}^{\mathrm{ana}} = \mathrm{span}(\Psi)$ and $\mathcal{U}^{\mathrm{syn}} = \mathrm{span}(\Phi)$.

The physical interaction structure further organizes these complete function families into G groups. Specifically, $\Psi = [\Psi_1, \ldots, \Psi_G]$ and $\Phi = [\Phi_1, \ldots, \Phi_G]$, where group g contains M_g analysis and synthesis functions, $\Psi_g = [\psi_{g,1}, \ldots, \psi_{g,M_g}]$ and $\Phi_g = [\phi_{g,1}, \ldots, \phi_{g,M_g}]$, with $M = \sum_{g=1}^{G} M_g$. Thus, $\psi_{g,k}$ and $\phi_{g,k}$ form a joint analysis–synthesis basis through the same physical-group index g and basis index k. To characterize the physical support of the analysis bases, let $I_g = \{n : m_{g,n} = 1\}$ denote the set of spatial points assigned to group g, where $m_{g,n}$ is the corresponding binary membership mask. We further define $\mathcal{V}_g = \{f \in \mathbb{R}^N : f_n = 0 \text{ for } n \notin I_g\}$ as the space of functions supported on that physical region. The masked analysis construction ensures that each $\psi_{g,k}$ is associated with the corresponding physical support I_g, while the complete family Ψ remains a global interaction-structured representation.

Lemma 1.1 (Adaptive Analysis and Synthesis Subspaces). The complete analysis and synthesis basis families Ψ and Φ span the global latent subspaces $\mathcal{U}^{\mathrm{ana}} = \mathrm{span}(\Psi)$ and $\mathcal{U}^{\mathrm{syn}} = \mathrm{span}(\Phi)$, whose dimensions are $\mathrm{rank}(\Psi)$ and $\mathrm{rank}(\Phi)$, respectively, with both ranks bounded by M. Moreover, each function family contains a maximal linearly independent subset that forms a basis of its corresponding subspace. In particular, when Ψ or Φ has full column rank, its complete set of M functions forms a basis of the corresponding learned subspace.

Proof. By definition, $\mathcal{U}^{\mathrm{ana}}$ and $\mathcal{U}^{\mathrm{syn}}$ are the column spaces of Ψ and Φ, respectively. Therefore, their dimensions are exactly the corresponding column ranks:

$$\dim(\mathcal{U}^{\mathrm{ana}}) = \mathrm{rank}(\Psi), \qquad \dim(\mathcal{U}^{\mathrm{syn}}) = \mathrm{rank}(\Phi). \tag{10}$$

Since both matrices contain M columns, $\mathrm{rank}(\Psi), \mathrm{rank}(\Phi) \le M$.

By the definition of matrix rank, one can select exactly $\mathrm{rank}(\Psi)$ linearly independent columns of Ψ whose span equals $\mathcal{U}^{\mathrm{ana}}$, and exactly $\mathrm{rank}(\Phi)$ linearly independent columns of Φ whose span equals $\mathcal{U}^{\mathrm{syn}}$. These subsets therefore form bases of the corresponding learned subspaces.

When Ψ has full column rank, all M columns are linearly independent and hence the complete analysis family forms a basis of $\mathcal{U}^{\mathrm{ana}}$. The same argument applies to Φ when it has full column rank.

□

Lemma 1.2 (Interaction-Structured Analysis Support). For every physical group g and every adaptive analysis function $\psi_{g,k}$, the masked analysis guarantees that $\psi_{g,k}$ vanishes outside the support I_g. Consequently, each analysis function belongs to the corresponding physical support space $\mathcal{V}_g$.

Proof. Recall that the adaptive analysis function for group g is defined by the masked normalization

$$\psi_{g,n,k} = \frac{m_{g,n} \exp(s_{g,n,k})}{\sum_{j=1}^{N} m_{g,j} \exp(s_{g,j,k})}. \tag{11}$$

For any point $n \notin I_g$, we have $m_{g,n} = 0$, and therefore $\psi_{g,n,k} = 0$ for every k. Hence, each $\psi_{g,k}$ is supported entirely within I_g and thus $\psi_{g,k} \in \mathcal{V}_g$. Since $\mathcal{V}_g$ is a linear subspace, any linear combination of the analysis functions within Ψ_g also belongs to $\mathcal{V}_g$. Therefore,

$$\mathrm{span}\{\psi_{g,1}, \ldots, \psi_{g,M_g}\} \subseteq \mathcal{V}_g. \tag{12}$$

Thus, although Ψ forms a global analysis representation, its constituent functions retain the physical support associated with their corresponding physical groups. □

Proposition 1 (Interaction-Structured Analysis–Synthesis Bases). Let $\Psi = [\Psi_1, \ldots, \Psi_G] \in \mathbb{R}^{N \times M}$ and $\Phi = [\Phi_1, \ldots, \Phi_G] \in \mathbb{R}^{N \times M}$ denote the complete adaptive analysis and synthesis function families, respectively. They span two input-adaptive global latent subspaces, $\mathcal{U}_{\mathrm{ana}} = \mathrm{span}(\Psi)$ and $\mathcal{U}_{\mathrm{syn}} = \mathrm{span}(\Phi)$, with their maximal linearly independent subsets forming corresponding bases. The analysis family is interaction-structured in that each $\psi_{g,k}$ is restricted to the physical region of group g, while $\psi_{g,k}$ corresponds to $\phi_{g,k}$ through the same group and basis index. Their composition induces the global interaction matrix $A = \Phi\Psi^\top$. When both Ψ and Φ have full column rank, all M analysis and synthesis functions themselves form bases of their respective learned subspaces, so that all available latent directions are retained without linear redundancy and $\mathrm{rank}(A) = M$.

Proof. By Lemma 1.1, the complete adaptive analysis and synthesis families span the global learned subspaces $\mathcal{U}^{\mathrm{ana}}$ and $\mathcal{U}^{\mathrm{syn}}$, respectively, and maximal linearly independent subsets of the two families form corresponding bases. By Lemma 1.2, every analysis function $\psi_{g,k}$ is supported only on the physical region associated with group g, while the shared group and basis index (g, k) associates it with the corresponding synthesis function $\phi_{g,k}$. Together, these properties establish the interaction-structured analysis–synthesis interpretation of the complete representation.

Now suppose that both Ψ and Φ have full column rank. Since each contains M columns,

$$\mathrm{rank}(\Psi) = \mathrm{rank}(\Phi) = M. \tag{13}$$

By Lemma 1.1, the complete sets of M analysis and synthesis functions therefore form bases of $\mathcal{U}^{\mathrm{ana}}$ and $\mathcal{U}^{\mathrm{syn}}$, respectively.

It remains to consider the induced interaction matrix $A = \Phi\Psi^\top$. Since Φ has full column rank and $\Psi^\top$ has full row rank, the Sylvester rank inequality gives

$$\mathrm{rank}(\Phi\Psi^\top) \geq \mathrm{rank}(\Phi) + \mathrm{rank}(\Psi^\top) - M = M. \tag{14}$$

On the other hand, $\mathrm{rank}(\Phi\Psi^\top) \leq M$. Therefore,

$$\mathrm{rank}(A) = M. \tag{15}$$

Hence, in the full-rank case, the complete analysis–synthesis representation realizes all M available latent directions without linear redundancy. □

Proposition 1 establishes the analysis–synthesis basis interpretation of the complete interaction-structured representation. The group-specific support structure within this global family further yields a useful rank-preservation property. When different physical groups occupy disjoint analysis supports, their physical modes cannot become linearly redundant across groups. Consequently, the independent modes contributed by the group-wise analysis families can accumulate in the global interaction representation. This provides a direct connection between the proposed interaction-structured basis construction and the interaction-rank behavior examined in Section 4.2.

Corollary 1.1 (Rank Preservation under Interaction Prior). Let $\Psi = [\Psi_1, \ldots, \Psi_G] \in \mathbb{R}^{N \times M}$ and $\Phi = [\Phi_1, \ldots, \Phi_G] \in \mathbb{R}^{N \times M}$, where $M = \sum_{g=1}^{G} M_g$, and define the global interaction matrix

$A = \Phi\Psi^\top$. If the physical groups have pairwise disjoint analysis supports, $I_g \cap I_h = \emptyset$ for all $g \neq h$, then the independent directions contributed by the group-specific analysis families are additive in the complete analysis representation,

$$\operatorname{rank}(\Psi) = \sum_{g=1}^{G} \operatorname{rank}(\Psi_g). \tag{16}$$

Furthermore, if the global synthesis family Φ has full column rank, the interaction matrix preserves all independent directions extracted by the analysis family, i.e., $\operatorname{rank}(A) = \operatorname{rank}(\Psi)$. Therefore, when every group-wise analysis family is also full column rank, the global interaction matrix reaches its maximal latent rank $\operatorname{rank}(A) = M$.

Proof. By Lemma 1.2, every column of Ψ_g is supported only on the physical region I_g. When the group supports are pairwise disjoint, any analysis function from group g has zero overlap with every analysis function from group $h \neq g$. Hence, the column spaces of different group-wise analysis families are mutually orthogonal. Their direct sum therefore has dimension equal to the sum of the individual dimensions, which gives $\operatorname{rank}(\Psi) = \sum_g \operatorname{rank}(\Psi_g)$.

We next consider the interaction matrix $A = \Phi\Psi^\top$. Since $\Phi \in \mathbb{R}^{N\times M}$ has full column rank, it defines an injective linear mapping on the M-dimensional latent space. Left multiplication by such a mapping cannot remove any independent row direction of $\Psi^\top$. Equivalently, the Sylvester rank inequality gives $\operatorname{rank}(\Phi\Psi^\top) \geq \operatorname{rank}(\Phi) + \operatorname{rank}(\Psi^\top) - M = \operatorname{rank}(\Psi)$. Together with the standard upper bound $\operatorname{rank}(\Phi\Psi^\top) \leq \operatorname{rank}(\Psi)$, we obtain $\operatorname{rank}(A) = \operatorname{rank}(\Psi)$.

Finally, if every group-wise analysis family has full column rank, then $\operatorname{rank}(\Psi_g) = M_g$ for all g. Rank additivity therefore gives $\operatorname{rank}(\Psi) = \sum_g M_g = M$, and hence $\operatorname{rank}(A) = M$. □

Remark 1.1 (Interaction Rank and Interaction-Structured Representation). Corollary 1.1 characterizes one idealized extreme in which the physical groups have mutually disjoint analysis supports, so that the independent directions contributed by different groups are exactly additive in the complete analysis family and can be preserved in the global interaction matrix through synthesis. At the opposite extreme, complete overlap removes the support-induced structural separation and approaches the homogeneous shared-support organization commonly adopted by existing Transformer-based neural operators. Practical SolidX configurations generally lie between these two extremes: although groups may partially overlap, explicit grouping still preserves structural separation across different interaction pathways, which can help maintain diverse interaction directions.

This property is particularly useful for heterogeneous multi-solid systems. If all physical interactions are learned jointly within a single homogeneous basis family, different interaction patterns must compete for the same latent directions, which may introduce redundancy or cause distinct modes to collapse into shared representations. In contrast, SolidX decomposes the learning problem according to the physical interaction structure, so that each group-specific basis family primarily models a smaller and more homogeneous subset of the overall physical behavior. The resulting independent directions can then be combined globally rather than being forced to emerge simultaneously from a single shared representation. This structured decomposition therefore reduces cross-interaction interference and makes it easier for the model to retain a larger number of effective interaction modes.

This interpretation is consistent with the empirical analysis in Section 4.2. Following the rank-based evaluation of LinearNO, we measure the average rank of the learned interaction matrix $\phi(Q)\psi^\top(K)$ across network layers and observe that SolidX maintains a substantially higher interaction rank after introducing the explicit physical interaction prior. The same tendency remains evident in the full model with partially overlapping groups, suggesting that interaction-structured basis organization enables more effective utilization of the available interaction modes in practice.

B.2 Proposition 2: Interaction-Structured Neural Operator

Here, we provide the detailed proof of Proposition 2, showing that the analysis–synthesis transformation of SolidX admits a neural-operator interpretation. We first rewrite the discrete transformation in the same matrix form as Section 3. We then introduce its continuum counterpart and show that the discrete group-wise analysis converges, under standard sampling assumptions, to a continuous integral transformation. Finally, we derive the resulting interaction-structured kernel and characterize its group-wise source support.

Continuum setting. For notational simplicity, we consider a single SolidX analysis–synthesis block. Let $\bar{H} : \Omega \to \mathbb{R}^{d_h \times 1}$ denote the normalized feature field presented to the value branch, and let $C : \Omega \to \mathbb{R}^{d_h \times 1}$ denote the persistent physical context. Their combination defines

$$R(x) = \bar{H}(x) + C(x), \tag{17}$$

which is used to construct the adaptive analysis and synthesis bases. For each physical group g, let $m_g : \Omega \to \{0, 1\}$ denote its membership indicator and define $\Omega_g = \{x \in \Omega : m_g(x) = 1\}$. We assume $\mu(\Omega_g) > 0$ for every retained group. The continuum analysis-score vector is

$$K_g(x) = W_{K,g}^\top R(x) \in \mathbb{R}^{M_g}, \tag{18}$$

and its corresponding analysis-basis vector is

$$\boldsymbol{\psi}_g(\xi) = \frac{m_g(\xi) \exp(K_g(\xi))}{\displaystyle\int_\Omega m_g(\eta) \exp(K_g(\eta))\, d\mu(\eta)} \in \mathbb{R}^{M_g}, \tag{19}$$

where the exponential, division, and integral in the denominator are applied component-wise over the M_g bases. Similarly, let $\boldsymbol{\phi}_g(x) \in \mathbb{R}^{M_g}$ denote the hierarchy-routed synthesis-basis vector corresponding to group g, whose entries are generated by the group-level and within-group routing defined in Section 3. The group-specific value transformation is

$$V_g(\xi) = W_{V,g}^\top \bar{H}(\xi). \tag{20}$$

Throughout the following block-wise analysis, $\bar{H}$ and C are treated as fixed bounded measurable fields. Consequently, $\boldsymbol{\psi}_g$ and $\boldsymbol{\phi}_g$ are also fixed measurable functions conditional on the input fields.

Proposition 2 (Interaction-Structured Neural Operator). For every fixed query location $x \in \Omega$, the discrete SolidX analysis–synthesis transformation converges almost surely, as $N \to \infty$, to an input-adaptive integral neural operator

$$o(x) = \int_\Omega \mathcal{K}(x, \xi; \bar{H}, C) \bar{H}(\xi)\, d\mu(\xi), \tag{21}$$

where the matrix-valued kernel admits the interaction-structured decomposition

$$\mathcal{K}(x, \xi; \bar{H}, C) = \sum_{g=1}^{G} \left[\boldsymbol{\phi}_g(x)^\top \boldsymbol{\psi}_g(\xi)\right] W_{V,g}^\top. \tag{22}$$

Thus, each physical group contributes a low-rank adaptive spatial interaction term whose source-side support is restricted to the corresponding physical region, while synthesis remains globally defined over the query domain.

Proof. We prove Proposition 2 in three steps. First, we rewrite the discrete SolidX analysis–synthesis transformation in matrix form and show that each physical group induces a group-wise spatial interaction matrix $\Phi_g^{(N)}(\Psi_g^{(N)})^\top$. Second, we establish the continuum limit of the group-wise basis analysis by showing that the discrete slice tokens $Z_g^{(N)} = (\Psi_g^{(N)})^\top V_g^{(N)}$ converge almost surely to their corresponding integral representations under standard sampling assumptions. Third, we substitute these continuous slice tokens into the hierarchical synthesis process and derive the resulting input-adaptive integral kernel. The detailed proof is given below.

Step 1: Matrix form of the discrete SolidX transformation. Let $\bar{H}_N \in \mathbb{R}^{N \times d_h}$ denote the normalized point-wise feature matrix on a finite discretization, whose i-th row is $\bar{H}(x_i)^\top$. Following the notation of Section 3, for each physical group g, SolidX constructs

$$\Psi_g^{(N)} \in \mathbb{R}^{N \times M_g}, \qquad V_g^{(N)} = \bar{H}_N W_{V,g} \in \mathbb{R}^{N \times d_h}. \tag{23}$$

The corresponding group-wise slice tokens are

$$Z_g^{(N)} = \left(\Psi_g^{(N)}\right)^\top V_g^{(N)} \in \mathbb{R}^{M_g \times d_h}. \tag{24}$$

The hierarchy-routed synthesis basis $\Phi_g^{(N)} \in \mathbb{R}^{N \times M_g}$ then maps these slice tokens back to the spatial domain. Hence, the complete analysis–synthesis transformation is

$$O_N = \sum_{g=1}^{G} \Phi_g^{(N)} Z_g^{(N)} = \sum_{g=1}^{G} \Phi_g^{(N)} \left(\Psi_g^{(N)}\right)^\top \bar{H}_N W_{V,g}. \tag{25}$$

Define the group-wise spatial interaction matrix

$$A_g^{(N)} = \Phi_g^{(N)} \left(\Psi_g^{(N)}\right)^\top \in \mathbb{R}^{N \times N}. \tag{26}$$

Then, Eq. (25) becomes

$$O_N = \sum_{g=1}^{G} A_g^{(N)} \bar{H}_N W_{V,g}. \tag{27}$$

Moreover, with $\Psi = [\Psi_1, \ldots, \Psi_G]$ and $\Phi = [\Phi_1, \ldots, \Phi_G]$, the global interaction matrix introduced in Proposition 1 satisfies

$$A = \Phi \Psi^\top = \sum_{g=1}^{G} \Phi_g \Psi_g^\top. \tag{28}$$

Thus, the interaction matrix analyzed in Proposition 1 is exactly the sum of the group-wise spatial interaction matrices appearing in the SolidX forward transformation.

Step 2: Continuum limit of the matrix analysis. Let $X_N = \{x_i\}_{i=1}^N$ be a spatial discretization with $x_i \overset{\text{i.i.d.}}{\sim} \mu$. For group g, the k-th column of $\Psi_g^{(N)}$ is the discrete evaluation of the corresponding adaptive analysis basis. Therefore, the k-th row of $Z_g^{(N)}$ can be written as

$$\left[Z_g^{(N)}\right]_{k,:} = \frac{\frac{1}{N}\sum_{j=1}^{N} m_g(x_j) \exp([K_g(x_j)]_k) \bar{H}(x_j)^\top W_{V,g}}{\frac{1}{N}\sum_{j=1}^{N} m_g(x_j) \exp([K_g(x_j)]_k)}. \tag{29}$$

Because $\bar{H}$, m_g, and K_g are fixed bounded measurable functions, the strong law of large numbers gives, for every $k = 1, \ldots, M_g$,

$$\left[Z_g^{(N)}\right]_{k,:} \xrightarrow{\text{a.s.}} \int_\Omega [\boldsymbol{\psi}_g(\xi)]_k \bar{H}(\xi)^\top W_{V,g}\, d\mu(\xi). \tag{30}$$

Since M_g is finite, the convergence holds jointly for all rows of $Z_g^{(N)}$. Hence,

$$Z_g^{(N)} \xrightarrow{\text{a.s.}} Z_g = \int_\Omega \boldsymbol{\psi}_g(\xi) \bar{H}(\xi)^\top W_{V,g}\, d\mu(\xi), \tag{31}$$

where $\boldsymbol{\psi}_g(\xi) \in \mathbb{R}^{M_g \times 1}$ and $\bar{H}(\xi)^\top W_{V,g} \in \mathbb{R}^{1 \times d_h}$, so the integrand is an $M_g \times d_h$ outer product.

Thus, the discrete matrix multiplication $\Psi_g^\top V_g$ is a Monte Carlo approximation of the corresponding continuous interaction-structured basis analysis.

Step 3: Continuous hierarchical synthesis. For a fixed query location x, the corresponding row of the synthesis basis matrix converges to $\boldsymbol{\phi}_g(x)^\top$. Using Eq. (31), the group-wise contribution at x therefore satisfies

$$\begin{aligned} o_g(x) &= Z_g^\top \boldsymbol{\phi}_g(x) \\ &= \left(\int_\Omega \boldsymbol{\psi}_g(\xi) \bar{H}(\xi)^\top W_{V,g}\, d\mu(\xi)\right)^\top \boldsymbol{\phi}_g(x) \\ &= \int_\Omega W_{V,g}^\top \bar{H}(\xi) \left[\boldsymbol{\psi}_g(\xi)^\top \boldsymbol{\phi}_g(x)\right] d\mu(\xi). \end{aligned} \tag{32}$$

Summing the contributions of all physical groups gives

$$o(x) = \sum_{g=1}^{G} o_g(x) = \int_{\Omega} \left\{ \sum_{g=1}^{G} \left[\boldsymbol{\phi}_g(x)^\top \boldsymbol{\psi}_g(\xi) \right] W_{V,g}^\top \right\} \bar{H}(\xi) d\mu(\xi). \tag{33}$$

The term in braces is exactly the kernel defined in Eq. (22). Therefore,

$$o(x) = \int_{\Omega} \mathcal{K}(x, \xi; \bar{H}, C) \bar{H}(\xi)\, d\mu(\xi). \tag{34}$$

Hence, conditional on the input feature field and persistent physical context, the discrete matrix-form SolidX transformation converges to an input-adaptive integral neural operator. □

Corollary 2.1 (Interaction-Structured Kernel Support). For each physical group g, the corresponding group-wise kernel component is

$$\mathcal{K}_g(x, \xi) = \left[\boldsymbol{\phi}_g(x)^\top \boldsymbol{\psi}_g(\xi) \right] W_{V,g}^\top. \tag{35}$$

Its source-side support is restricted to the physical region associated with group g, namely $\mathrm{supp}_\xi(\mathcal{K}_g) \subseteq \Omega_g$. Thus, although the synthesis basis is defined over the full query domain, the source information contributing through $\mathcal{K}_g$ can only originate from Ω_g.

Proof. By definition, $\Omega_g = \{\xi \in \Omega : m_g(\xi) = 1\}$. For any source location $\xi \notin \Omega_g$, we have $m_g(\xi) = 0$. According to Eq. (19), the membership indicator directly multiplies every component of the analysis-basis vector, and therefore

$$\boldsymbol{\psi}_g(\xi) = \mathbf{0}, \qquad \forall\, \xi \notin \Omega_g. \tag{36}$$

Substituting this result into the group-wise kernel gives

$$\mathcal{K}_g(x, \xi) = \left[\boldsymbol{\phi}_g(x)^\top \boldsymbol{\psi}_g(\xi) \right] W_{V,g}^\top = \mathbf{0} \tag{37}$$

for every query location x whenever $\xi \notin \Omega_g$. Hence, no source information outside the corresponding physical region can contribute through $\mathcal{K}_g$, which establishes $\mathrm{supp}_\xi(\mathcal{K}_g) \subseteq \Omega_g$. □

Multi-head Extension. The matrix-form analysis extends directly to the multi-head implementation of SolidX. For head a, the corresponding transformation is

$$O_N^{(a)} = \sum_{g=1}^{G} \Phi_g^{(a)} \left(\Psi_g^{(a)} \right)^\top \bar{H}_N W_{V,g}^{(a)}. \tag{38}$$

Each head therefore admits the same interaction-structured neural-operator interpretation established above. Since the physical-group masks are shared across heads, the source-support property remains unchanged. The complete multi-head representation is obtained by concatenating the head-wise outputs along the feature dimension.

Remark 2.1 (Block-wise Continuum Interpretation). Proposition 2 characterizes the continuum limit of a SolidX analysis–synthesis block conditional on its input feature field and physical context. In particular, the discrete matrix multiplication $\Psi_g^\top V_g$ acts as a Monte Carlo approximation of a continuous interaction-structured basis analysis, while $\Phi_g Z_g$ provides the corresponding global synthesis. The proposition therefore establishes the neural-operator interpretation of the SolidX transformation at the block level. For a complete stacked architecture, a corresponding layer-wise continuum interpretation additionally requires the intermediate feature fields to admit consistent continuum representations across successive blocks.

Remark 2.2 (Input Adaptivity and Nonlinearity). The kernel representation above is derived conditional on fixed $\bar{H}$ and C. In SolidX, however, both the analysis bases $\boldsymbol{\psi}_g$ and synthesis bases $\boldsymbol{\phi}_g$ are generated from the input feature and physical context fields. Consequently, the kernel $\mathcal{K}(x, \xi; \bar{H}, C)$ itself changes with the input. Thus, although the aggregation is linear once the adaptive analysis and synthesis bases are fixed, the overall SolidX transformation defines an input-adaptive nonlinear neural operator rather than a fixed linear integral operator.

Corollary 2.2 (Adaptive Global Synthesis). Although each group-wise kernel $\mathcal{K}_g$ is restricted to Ω_g on the source side, no analogous support restriction is imposed on the query coordinate x. The physical mask constrains the analysis basis $\boldsymbol{\psi}_g(\xi)$, whereas the hierarchy-routed synthesis basis $\boldsymbol{\phi}_g(x)$ is defined over the full query domain. Therefore, slice tokens extracted from a particular group can contribute to query locations outside that group during reconstruction. Consequently, SolidX separates different physical pathways during basis analysis while allowing its slice tokens to be adaptively combined across the query domain.

B.3 Proposition 3: Reduced Complexity under Interaction Prior

In this subsection, we analyze in an idealized setting how interaction priors can reduce the representational complexity of the analysis–synthesis mapping. Specifically, we consider a single analysis–synthesis block for a fixed input and first introduce the spatial mapping and interaction-group notation used in our analysis. Proposition 3 then compares the shared and interaction-structured representations under the same total basis budget and establishes the general complexity reduction result. Corollary 3.1 further gives an exact characterization for non-overlapping groups, while Corollary 3.2 extends the result to partially overlapping groups under the uniform basis allocation used in our implementation. The detailed analysis is given below.

Setup and Notation. For a fixed input to one analysis–synthesis block, let $T^\star \in \mathbb{R}^{N\times N}$ denote the target spatial mapping from source points to query points, where $T^\star_{ij}$ represents the contribution of source point j to query point i. We omit the feature dimension and focus on the spatial transformation. Let $\mathcal{I}_g \subseteq \{1,\dots,N\}$ denote the source-point indices associated with physical group g, with $n_g = |\mathcal{I}_g|$. The groups may overlap and cover the source domain, $\bigcup_{g=1}^{G} \mathcal{I}_g = \{1,\dots,N\}$. For each group, let $T_g \in \mathbb{R}^{N\times N}$ denote its contribution to the spatial mapping, with $\mathrm{supp}_{\mathrm{src}}(T_g) \subseteq \mathcal{I}_g$. Thus, each T_g is restricted to its group on the source side, while its query side remains global. We assign M_g physical bases to group g, with $M = \sum_{g=1}^{G} M_g$, where $M_g \le n_g$ and $M \le N$. Each group-wise contribution therefore satisfies $\mathrm{rank}(T_g) \le M_g$. We call an interaction prior *compatible* with the target mapping if $T^\star$ admits a decomposition $T^\star = \sum_{g=1}^{G} T^\star_g$ such that $\mathrm{supp}_{\mathrm{src}}(T^\star_g) \subseteq \mathcal{I}_g$ and $\mathrm{rank}(T^\star_g) \le M_g$ for every g.

Lemma 3.1 (Dimension of a Low-Rank Matrix Family). The family of $m \times n$ matrices with rank at most r has maximal dimension $r(m+n-r)$, where $r \le \min(m,n)$. Formally,

$$\mathcal{M}^{(r)}_{m,n} = \left\{ T \in \mathbb{R}^{m\times n} : \mathrm{rank}(T) \le r \right\}, \qquad \dim_{\max}\left(\mathcal{M}^{(r)}_{m,n}\right) = r(m+n-r). \tag{39}$$

Proof. The maximal-dimensional stratum of $\mathcal{M}^{(r)}_{m,n}$ consists of matrices with rank exactly r. Any such matrix can be factorized as $T = UV^\top$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ both have full column rank. Together, U and V contain $mr + nr$ scalar parameters.

This factorization is not unique. For any invertible matrix $R \in \mathbb{R}^{r\times r}$,

$$UV^\top = (UR)(VR^{-\top})^\top, \tag{40}$$

so an r^2-dimensional change of basis leaves T unchanged. The rank-r stratum therefore has dimension $mr + nr - r^2 = r(m+n-r)$. All lower-rank strata have smaller dimension, and hence

$$\dim_{\max}\left(\mathcal{M}^{(r)}_{m,n}\right) = r(m+n-r). \tag{41}$$

□

Proposition 3 (Reduced Representational Complexity under Interaction Prior). Under the same total basis budget $M = \sum_{g=1}^{G} M_g$, we compare the shared hypothesis class and the interaction-structured hypothesis class,

$$\begin{aligned} \mathcal{H}_{\mathrm{shared}} &= \left\{ T \in \mathbb{R}^{N\times N} : \mathrm{rank}(T) \le M \right\}, \\ \mathcal{H}_{\mathrm{struct}} &= \left\{ \sum_{g=1}^{G} T_g : \mathrm{rank}(T_g) \le M_g,\ \mathrm{supp}_{\mathrm{src}}(T_g) \subseteq \mathcal{I}_g \right\}. \end{aligned} \tag{42}$$

If the interaction prior is compatible with the target mapping, then $T^\star \in \mathcal{H}_{\text{struct}} \subseteq \mathcal{H}_{\text{shared}}$. Hence, the loading interaction structure preserves the target mapping while restricting the admissible transformation family. Moreover, while the shared class has maximal dimension $d_{\text{shared}} = M(2N - M)$, the structured class satisfies

$$\dim_{\max}(\mathcal{H}_{\text{struct}}) \leq \sum_{g=1}^{G} M_g(N + n_g - M_g). \tag{43}$$

Therefore, the interaction-structured representation has strictly fewer representational degrees of freedom whenever the right-hand side above is smaller than $M(2N - M)$.

Proof. We prove the proposition in three steps. First, we establish target realizability and the containment of the interaction-structured hypothesis class in the shared class. Second, we derive an upper bound on the dimension of the structured class. Finally, we compare this bound with the dimension of the shared class.

Step 1: Target realizability and hypothesis-class containment. By the compatibility assumption, there exists a decomposition $T^\star = \sum_{g=1}^{G} T_g^\star$ such that $\text{supp}_{\text{src}}(T_g^\star) \subseteq \mathcal{I}_g$ and $\text{rank}(T_g^\star) \leq M_g$ for every group g. Therefore, $T^\star \in \mathcal{H}_{\text{struct}}$.

For any $T = \sum_{g=1}^{G} T_g \in \mathcal{H}_{\text{struct}}$, the subadditivity of matrix rank gives

$$\text{rank}(T) \leq \sum_{g=1}^{G} \text{rank}(T_g) \leq \sum_{g=1}^{G} M_g = M. \tag{44}$$

Hence, $T \in \mathcal{H}_{\text{shared}}$, and therefore

$$T^\star \in \mathcal{H}_{\text{struct}} \subseteq \mathcal{H}_{\text{shared}}. \tag{45}$$

Importantly, this containment does not require the physical groups to be non-overlapping.

Step 2: Dimension of the interaction-structured class. For each physical group g, define

$$\mathcal{S}_g = \left\{ T_g \in \mathbb{R}^{N \times N} : \text{rank}(T_g) \leq M_g,\ \text{supp}_{\text{src}}(T_g) \subseteq \mathcal{I}_g \right\}. \tag{46}$$

Since all columns outside $\mathcal{I}_g$ are identically zero, every $T_g \in \mathcal{S}_g$ is completely determined by its $N \times n_g$ submatrix on the source coordinates in $\mathcal{I}_g$. Thus, $\mathcal{S}_g$ is naturally identified with the family of $N \times n_g$ matrices of rank at most M_g. By Lemma 3.1,

$$\dim_{\max}(\mathcal{S}_g) = M_g(N + n_g - M_g). \tag{47}$$

The structured hypothesis class is obtained by summing one element from each group-wise family:

$$\mathcal{H}_{\text{struct}} = \left\{ \sum_{g=1}^{G} T_g : T_g \in \mathcal{S}_g \right\}. \tag{48}$$

Equivalently, it is the image of the product family $\mathcal{S}_1 \times \cdots \times \mathcal{S}_G$ under the linear summation map $(T_1, \ldots, T_G) \mapsto \sum_g T_g$. Since a linear map cannot increase the dimension of its domain,

$$\dim_{\max}(\mathcal{H}_{\text{struct}}) \leq \sum_{g=1}^{G} \dim_{\max}(\mathcal{S}_g) = \sum_{g=1}^{G} M_g(N + n_g - M_g). \tag{49}$$

This is an upper bound in the overlapping case because different group-wise components may induce redundant representations on their shared source coordinates.

For the shared hypothesis class, applying Lemma 3.1 with $m = n = N$ and $r = M$ gives

$$d_{\text{shared}} = \dim_{\max}(\mathcal{H}_{\text{shared}}) = M(2N - M). \tag{50}$$

Step 3: Reduction in representational complexity. Combining the above results, we obtain

$$\dim_{\max}(\mathcal{H}_{\text{struct}}) \leq \sum_{g=1}^{G} M_g(N + n_g - M_g). \tag{51}$$

Therefore, whenever

$$\sum_{g=1}^{G} M_g(N + n_g - M_g) < M(2N - M), \tag{52}$$

we have

$$\dim_{\max}(\mathcal{H}_{\text{struct}}) < \dim_{\max}(\mathcal{H}_{\text{shared}})\,. \tag{53}$$

Hence, under a compatible interaction prior, the structured representation preserves the target mapping while admitting a strictly lower-dimensional hypothesis class whenever the stated condition holds. □

Corollary 3.1 (Non-Overlapping Groups). If the physical groups form a non-overlapping partition of the source domain, i.e., $\mathcal{I}_g \cap \mathcal{I}_h = \emptyset$ for $g \neq h$ and $\sum_g n_g = N$, the dimension bound in Eq. (43) becomes exact:

$$d_{\text{struct}} = \sum_{g=1}^{G} M_g(N + n_g - M_g), \qquad d_{\text{shared}} - d_{\text{struct}} = \sum_{g=1}^{G} \sum_{\substack{h=1 \\ h\neq g}}^{G} M_g(n_h - M_h) \geq 0. \tag{54}$$

The inequality is strict under non-degenerate multi-group settings.

Proof. When the groups are non-overlapping, their source supports are pairwise disjoint. Therefore, the decomposition $T = \sum_{g=1}^{G} T_g$ is unique with respect to the group-wise source coordinates: each nonzero column of T_g belongs only to $\mathcal{I}_g$ and cannot be represented by another group.

Consequently, the summation map from $\mathcal{S}_1 \times \cdots \times \mathcal{S}_G$ to $\mathcal{H}_{\text{struct}}$ is injective, and the dimensions of the group-wise families add exactly. By Lemma 3.1,

$$d_{\text{struct}} = \sum_{g=1}^{G} M_g(N + n_g - M_g). \tag{55}$$

Using $M = \sum_{g=1}^{G} M_g$ and $N = \sum_{g=1}^{G} n_g$, the difference from the shared class becomes

$$\begin{aligned} d_{\text{shared}} - d_{\text{struct}} &= M(2N - M) - \sum_{g=1}^{G} M_g(N + n_g - M_g) \\ &= MN - M^2 - \sum_{g=1}^{G} M_g n_g + \sum_{g=1}^{G} M_g^2 \\ &= \sum_{g=1}^{G} \sum_{\substack{h=1 \\ h\neq g}}^{G} M_g(n_h - M_h). \end{aligned} \tag{56}$$

Since $M_h \leq n_h$ for every group h, all terms in the final summation are non-negative. Hence,

$$d_{\text{struct}} \leq d_{\text{shared}}. \tag{57}$$

The inequality is strict whenever there exists a pair $g \neq h$ such that $M_g > 0$ and $M_h < n_h$, which corresponds to the non-degenerate multi-group setting stated in the corollary. □

Corollary 3.2 (Partial Overlap under Uniform Basis Allocation). For the uniform allocation used in our implementation, $M_g = M/G$, let $\Delta = \sum_{g=1}^{G} n_g - N$ measure the total overlap beyond a non-overlapping partition. Proposition 3 then gives

$$d_{\text{shared}} - \dim_{\max}(\mathcal{H}_{\text{struct}}) \geq \frac{M}{G}\left[(G-1)(N-M) - \Delta\right]. \tag{58}$$

Therefore, a strict reduction is guaranteed whenever $\Delta < (G-1)(N-M)$, showing that the interaction-structured representation does not require mutually exclusive physical groups and remains effective under partial overlap.

Proof. Under the uniform allocation $M_g = M/G$, the dimension bound in Proposition 3 becomes

$$\begin{aligned}\dim_{\max}(\mathcal{H}_{\text{struct}}) &\le \sum_{g=1}^{G} \frac{M}{G}\left(N + n_g - \frac{M}{G}\right) \\ &= MN + \frac{M}{G}\sum_{g=1}^{G} n_g - \frac{M^2}{G}.\end{aligned} \tag{59}$$

By the definition $\Delta = \sum_{g=1}^{G} n_g - N$, we have $\sum_{g=1}^{G} n_g = N + \Delta$. Substituting this relation together with $d_{\text{shared}} = M(2N - M)$ gives

$$\begin{aligned}d_{\text{shared}} - \dim_{\max}(\mathcal{H}_{\text{struct}}) &\ge M(2N - M) - \left(MN + \frac{M}{G}(N + \Delta) - \frac{M^2}{G}\right) \\ &= \frac{M}{G}\left[(G - 1)(N - M) - \Delta\right].\end{aligned} \tag{60}$$

Therefore, whenever

$$\Delta < (G - 1)(N - M), \tag{61}$$

the right-hand side is strictly positive, which implies

$$\dim_{\max}(\mathcal{H}_{\text{struct}}) < d_{\text{shared}}. \tag{62}$$

Thus, the reduction in representational complexity remains strict for partially overlapping physical groups whenever the total overlap satisfies the stated condition. □

Remark 3.1 (Effect of Interaction-Group Overlap). Proposition 3 and Corollaries 3.1–3.2 together characterize how the interaction structure controls representational complexity. The non-overlapping case imposes the strongest source-side structural constraint and admits an exact dimension characterization, while partial overlap introduces additional flexibility by allowing some source points to participate in multiple physical groups. Importantly, such overlap does not remove the benefit of interaction-structured organization: when $\Delta < (G - 1)(N - M)$, the structured hypothesis class remains strictly lower-dimensional than the shared class. This result provides a capacity-based interpretation of the interaction prior in SolidX. Under the same total basis budget, the physical groups restrict how the basis capacity is organized on the analysis side, while global synthesis still allows the extracted slice tokens to contribute across the full query domain.

C Computational Complexity

In this section, we provide a detailed analysis of the computational complexity of SolidX. Specifically, we analyze the interaction-structured basis analysis and hierarchy-routed basis synthesis, and then derive the complexity of the complete network. After simplification, the overall complexity is $\mathcal{O}(LNd_h(d_hG + M + F))$, which is ***linear*** with respect to the number of spatial points N when the model hyperparameters are fixed. The detailed analysis is provided below.

Complexity of Interaction-Structured Analysis–Synthesis: We first analyze the core interaction-structured module in each SolidX block, which consists of interaction-structured basis analysis followed by hierarchy-routed basis synthesis.

• *Interaction-Structured Basis Analysis*: The analysis stage compresses the point-wise representation into group-specific slice tokens. Specifically, the input contains N spatial points, each represented by a d_h-dimensional hidden feature, giving an overall point-wise feature matrix of size $N \times d_h$. For each physical group g, the key projection maps these point-wise features to $N \times M_g$ analysis scores, where M_g denotes the number of physical bases assigned to the group. This projection requires $\mathcal{O}(Nd_hM_g)$ operations, and $\mathcal{O}(Nd_hM)$ across all groups. The group-specific value projection maps the $N \times d_h$ features to another $N \times d_h$ representation, resulting in $\mathcal{O}(Nd_h^2)$ complexity per group and $\mathcal{O}(Nd_h^2G)$ across all groups. The subsequent masked normalization operates on the $N \times M_g$ analysis scores and requires $\mathcal{O}(NM)$ operations in total. The normalized analysis weights then aggregate the N point-wise value features into M_g slice tokens, each with feature dimension d_h. Accordingly,

multiplying the $M_g \times N$ analysis weights with the $N \times d_h$ value features requires $\mathcal{O}(Nd_hM_g)$ operations for each group and $\mathcal{O}(Nd_hM)$ across all groups. Therefore, after omitting constant factors and the lower-order normalization term, the overall complexity of interaction-structured basis analysis is $\mathcal{O}(Nd_h^2G + Nd_hM)$.

• *Hierarchy-Routed Basis Synthesis*: The synthesis stage reverses the above transformation by mapping the group-wise slice tokens back to the point-wise domain. That is, each group starts from M_g slice tokens of dimension d_h and reconstructs features for the N spatial points. The group-level routing maps the $N \times d_h$ point-wise features to $N \times G$ routing scores, requiring $\mathcal{O}(Nd_hG)$ operations. For each physical group, the basis-level routing further maps the point-wise features to $N \times M_g$ basis scores. Across all groups, this step requires $\mathcal{O}(Nd_hM)$ operations. The hierarchical normalization is performed on the group-level and basis-level routing scores, with an overall complexity of $\mathcal{O}(N(G + M))$. The resulting synthesis weights determine how the M_g slice tokens of each group are distributed back to the N spatial points. Specifically, the $N \times M_g$ synthesis weights are multiplied with the $M_g \times d_h$ slice tokens, requiring $\mathcal{O}(Nd_hM_g)$ operations for each group and $\mathcal{O}(Nd_hM)$ across all groups. Therefore, after omitting lower-order normalization terms and constant factors, the overall complexity of hierarchy-routed basis synthesis is $\mathcal{O}(Nd_hG + Nd_hM)$.

Combining the interaction-structured basis analysis and hierarchy-routed basis synthesis, the computational complexity of the core interaction-structured module in one SolidX block is

$$\mathcal{O}\big(N\big[d_h^2G + d_h(G + M)\big]\big)\,. \tag{63}$$

Importantly, SolidX does not construct a point-to-point $N \times N$ interaction matrix. For fixed G, M, and d_h, its complexity therefore grows linearly with the number of spatial points N.

Complexity of the Complete SolidX Network: We next consider the remaining operations in a complete SolidX block. Besides the interaction-structured analysis–synthesis module, each block contains layer normalization, physical-context injection, feature masking, residual updates, and a point-wise feed-forward network. The normalization, context injection, masking, and residual operations all act directly on the $N \times d_h$ point-wise features and therefore each require $\mathcal{O}(Nd_h)$ computation. Together, these point-wise operations remain $\mathcal{O}(Nd_h)$ up to constant factors.

The feed-forward network maps each d_h-dimensional point-wise feature to an intermediate feature of dimension F, where F denotes the intermediate width of the feed-forward network, and then projects it back to d_h dimensions. Its computational complexity is therefore $\mathcal{O}(Nd_hF)$ after omitting constant factors. Combining these operations with the interaction-structured module in Eq. (63), one complete SolidX block has complexity $\mathcal{O}(N[d_h^2G + d_h(G + M + F)])$.

For a network containing L SolidX blocks, the repeated backbone consequently requires $\mathcal{O}(LN[d_h^2G + d_h(G + M + F)])$ operations. The input encoder and physical-context encoder are executed once before the repeated blocks, while the output decoder is applied once after the final block. Since these components mainly consist of point-wise projections or small multilayer perceptrons, their computational costs are also linear in N and do not change the overall asymptotic dependence on the spatial discretization size.

The above analysis characterizes the complexity of a forward pass. For brevity, candidate interface identification can be treated here as a separate preprocessing step, whose complexity is independent of our SolidX network and is therefore not further derived in this analysis. Backpropagation through the same operations has the same asymptotic order up to a constant factor, while the total training cost additionally depends on the number of optimization steps.

Simplification of Complexity Results: Based on the above analysis, the computational complexity of the complete L-block SolidX network is $\mathcal{O}(LN[d_h^2G + d_h(G + M + F)])$. After omitting constant factors and lower-order terms, the Nd_hG contribution can be absorbed into the dominant Nd_h^2G term. The overall complexity can therefore be simplified as

$$\mathcal{O}(LNd_h(d_hG + M + F))\,. \tag{64}$$

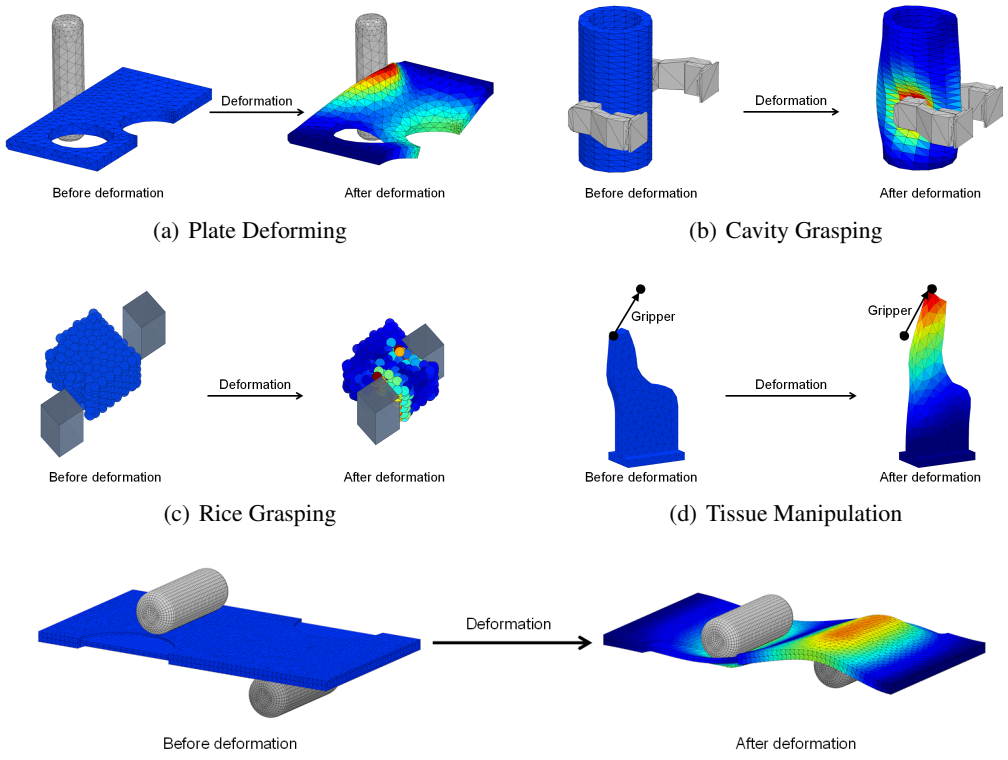
In practical architectures, L, G, M, d_h, and F are fixed model hyperparameters that do not grow with the number of spatial points N. Therefore, the computational complexity of SolidX is $\mathcal{O}(N)$ with respect to the spatial discretization size. This shows that introducing interaction-structured basis organization preserves the ***linear*** scalability of the underlying analysis–synthesis framework.

D EXPERIMENTAL DETAILS AND ADDITIONAL RESULTS

D.1 BENCHMARK DATASETS

In our experiments, we construct five unified benchmarks based on representative three-dimensional multi-solid datasets, including Plate Deforming (Pfaff et al., 2021), Cavity Grasping (Linkerhägner et al., 2023), Rice Grasping (Li et al., 2019), Tissue Manipulation (Linkerhägner et al., 2023), and Bilateral Stamping (Tao et al., 2025). Following industrial practice (Wu et al., 2026; Ansys Inc., 2026), we use approximately 100 training samples and 50 test samples for each benchmark, reflecting the limited data in real-world simulations. The main characteristics and visualizations of the datasets are shown in Table 2 and Figure 4, respectively. Specifically, Table 2 summarizes their material types, solid compositions, physical-group configurations, simulation frameworks, and prediction targets, while Figure 4 provides representative visualizations of the corresponding multi-solid systems.

For all datasets, we adopt a unified *long-horizon prediction* setting. Given the initial configuration, physical attributes, and loading rigid-body motion, the model directly predicts the target physical fields at the target state without autoregressive rollout. The loading rigid components serve as physical and geometric context and are not prediction targets, while predictions and evaluation are performed over the deformable solids. Depending on the benchmark, the target fields include deformed geometry, internal stress, and equivalent plastic strain (PEEQ). We next describe each benchmark in detail.



(a) Plate Deforming (b) Cavity Grasping

(c) Rice Grasping (d) Tissue Manipulation

(e) Bilateral Stamping

Figure 4: Visualization of the five benchmark datasets. Each example shows the initial multi-solid configuration and its corresponding target deformed state under the long-horizon prediction setting.

Plate Deforming (Pfaff et al., 2021). This dataset considers a hyperelastic deformable plate pressed by a rigid body, as shown in Figure 4(a). The original physical trajectories are generated using COMSOL Multiphysics on irregular tetrahedral meshes. The simulation contains 400 time steps, with the largest rigid-body displacement occurring at approximately step 340. In addition to the deformed geometry, the internal stress field of the plate is also predicted. Each processed sample contains approximately 1,271 discretized spatial points.

Table 2: Summary of the five benchmark datasets with different physical and interaction complexities. Under #Content, #DS, #RS, and #IG denote the numbers of deformable solids, rigid solids, and interaction groups, respectively. Simulator denotes the numerical simulation framework used to generate the original dataset.

Dataset	#Material	#Content			Simulator	Output Physics
		#DS	#RS	#IG		
Plate Deforming	Hyperelasticity	1	1	2	COMSOL	Geometry, Stress
Cavity Grasping	Elasticity	1	2	3	SOFA	Geometry
Rice Grasping	Elasto-Plasticity	1	2	3	NVIDIA FleX	Geometry
Tissue Manipulation	Elasticity	1	2	3	SOFA	Geometry
Bilateral Stamping	Elasto-Plasticity	2	2	6	ABAQUS	Geometry, Stress, PEEQ

Cavity Grasping (Linkerhägner et al., 2023). This dataset considers an elastic cone-shaped cavity manipulated by two independently moving rigid jaws, as shown in Figure 4(b). The trajectories are generated using the SOFA simulation framework. The cavity radius varies within $[50, 87.5]$, while its Poisson's ratio is selected from $\{-0.9,\ 0.0,\ 0.49\}$, introducing considerable variations in material response. The simulation contains 105 time steps, and each processed sample consists of approximately 1,386 spatial points. The prediction target is the deformed cavity geometry.

Rice Grasping (Li et al., 2019). This dataset considers an elasto-plastic deformable body manipulated by two rigid cuboid grippers, as shown in Figure 4(c). The physical trajectories are simulated using NVIDIA FleX through its PyFleX interface. The grippers are initialized with varying positions and orientations and move toward each other to compress the deformable material. The processed sequence contains 41 simulation steps and approximately 1,106 spatial points per sample. The prediction target is the final deformed geometry.

Tissue Manipulation (Linkerhägner et al., 2023). This dataset considers an elastic soft-tissue manipulation scenario involving a rigid gripper, as shown in Figure 4(d). A fixed boundary is imposed on the tissue, corresponding to the boundary region shown in the visualization. The dataset is generated using the SOFA simulation framework. To vary the material stiffness, the Young's modulus is sampled from $\{10{,}000,\ 30{,}000,\ 80{,}000\}$. The simulation contains 105 time steps, and each sample contains approximately 363 spatial points. The prediction target is the final tissue geometry.

Bilateral Stamping (Tao et al., 2025). This dataset considers an industrially motivated metal-forming scenario involving a deformable aluminum workpiece, an internal deformable rubber component, and two rigid stamping dies, as shown in Figure 4(e). Both deformable components exhibit elasto-plastic behavior. The dataset is generated using ABAQUS. The aluminum workpiece uses a Young's modulus of $69{,}000\ \mathrm{MPa}$ and a Poisson's ratio of 0.37, whereas the rubber component uses $1{,}306.26\ \mathrm{MPa}$ and 0.27, respectively. The die positions, loading motions, and geometric configurations vary across samples, resulting in diverse multi-solid contact configurations. Each sample contains approximately 13,714 spatial points, and the prediction targets include geometry, internal stress, and PEEQ.

Overall, the five benchmarks progressively increase in material, geometric, and interaction complexity, providing a diverse testbed for evaluating structured multi-solid physical modeling.

D.2 Experimental Settings

Here, we provide detailed experimental settings used throughout this work. We first introduce the training protocol, including the dataset splits and optimization settings, followed by the model configurations and hyperparameters used for all compared methods. We then describe the evaluation metric adopted for performance assessment. The detailed settings are presented below.

Training Protocol. To ensure a fair comparison, all compared methods follow the same training protocol, including the same long-horizon prediction setting, identical dataset splits, and unified optimization settings. The detailed dataset and training configurations are summarized in Table 3. Specifically, following industrial practice (Wu et al., 2026; Ansys Inc., 2026), where high-fidelity simulation data are typically limited and costly to obtain, we adopt a data-scarce setting with approximately 100 training samples and 50 validation and test samples for each benchmark. Plate Deforming,

Table 3: Training and dataset configurations for the five benchmark datasets. #Train, #Val, and #Test denote the numbers of training, validation, and test samples, respectively, while #Mesh denotes the mean number of discretized spatial points in each sample.

Dataset	Dataset Configuration				Training Configuration				
	#Train	#Val	#Test	#Mesh	Loss	Epochs	Optimizer	LR	Batch
Plate Deforming	100	50	50	1271	Rel L_2	100	AdamW	3×10^{-4}	1
Cavity Grasping	105	51	51	1386	Rel L_2	100	AdamW	3×10^{-4}	1
Rice Grasping	100	50	50	1106	Rel L_2	100	AdamW	3×10^{-4}	1
Tissue Manipulation	105	51	51	362	Rel L_2	100	AdamW	3×10^{-4}	1
Bilateral Stamping	100	50	50	13714	Rel L_2	100	AdamW	3×10^{-4}	1

Rice Grasping, and Bilateral Stamping use $100/50/50$ samples for training/validation/testing, respectively, while Cavity Grasping and Tissue Manipulation use $105/51/51$ samples. All methods are trained for 100 epochs using AdamW with an initial learning rate of 3×10^{-4} under the same optimization settings. We use single-sample training with a batch size of 1, since each sample represents a complete three-dimensional multi-solid system and different samples may contain different numbers of spatial points. This allows each system to be processed at its original spatial discretization without additional padding or packing overhead. To reduce concerns about run-to-run randomness, we repeat experiments for representative methods using different random seeds and report the results in Table 5, where SolidX shows consistent performance improvements across runs.

Baselines. We compare SolidX with a broad range of learning-based physical solvers, covering graph-based models, conventional neural operators, and recent Transformer-based neural operators. For graph-based models, we include GraphSAGE (Hamilton et al., 2017), MGN (Pfaff et al., 2021), HOOD (Grigorev et al., 2023), and HCMT (Yu et al., 2024). For neural operator baselines, we consider GNO (Li et al., 2020), Geo-FNO (Li et al., 2023b), LSM (Wu et al., 2023), and GINO (Li et al., 2023c). We further compare against recent Transformer-based neural operators, including Transolver (Wu et al., 2024), Transolver++ (Luo et al., 2025), Unisoma (Tao et al., 2025), LinearNO (Hu et al., 2026), HNO (Pei et al., 2026), and FuncAttn (Xiao et al., 2026). We implement these baselines using their official or widely adopted implementations whenever available and configure them for the same long-horizon prediction setting. To ensure a controlled comparison, all methods follow the same training protocol, including identical dataset splits, training budgets, and optimization settings. For comparable Transformer-based baselines, we additionally match the number of latent slices, layers, and attention heads whenever applicable. We also keep the parameter count of SolidX comparable to those of all baseline methods. The major architecture hyperparameters and parameter counts used in our experiments are summarized in Table 4. For additional implementation details, we refer readers to the corresponding original papers and repositories.

Evaluation Metrics. We evaluate SolidX from both predictive performance and computational efficiency. For predictive performance, we use the relative L2 error as the primary metric. Given the ground-truth physical field $\mathbf{y}$ and the corresponding prediction $\hat{\mathbf{y}}$, the relative error is defined as

$$\mathcal{E}_{\mathrm{rel}} = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y}\|_2}. \tag{65}$$

The error is evaluated over the predicted deformable-solid points, while the rigid components with loading motion are not included as prediction targets. Lower values indicate more accurate predictions. For benchmarks containing multiple physical quantities, including geometry, stress, and PEEQ, the corresponding relative errors are reported separately. For Bilateral Stamping, the errors of the metal plate and rubber insert, denoted by "M" and "R", respectively, are reported separately. For computational efficiency, we report the number of GFLOPs to quantify the computational cost of each model. We additionally report GPU memory usage and the number of model parameters to provide a more comprehensive comparison of computational and model complexity.

Table 4: Key hyperparameters and parameter counts of models.

Model	Hyperparameter	Deforming Plate	Cavity Grasping	Rice Grasping	Tissue Manipulation	Bilateral Stamping
GraphSAGE	Hiddens	256	256	256	256	256
	Layers	8	8	8	12	12
	Parameter (M)	0.99	0.99	0.99	1.52	1.52
MGN	Hiddens	128	128	128	128	128
	Layers	15	15	15	15	15
	Parameter (M)	2.33	2.33	2.33	2.88	2.88
HOOD	Hiddens	128	128	128	128	128
	Layers	4	4	4	4	4
	Parameter (M)	1.19	1.19	1.19	1.19	1.19
HCMT	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Parameter (M)	1.00	1.00	1.00	1.00	1.00
GNO	Hiddens	256	256	256	256	256
	Kernel Width	256	256	256	256	256
	Layers	6	6	6	6	6
	Parameter (M)	1.00	1.00	1.00	1.00	1.00
Geo-FNO	Modes	[8,8,8]	[8,8,8]	[8,8,8]	[8,8,8]	[8,8,8]
	Width	24	24	24	36	36
	Parameter (M)	1.19	1.19	1.19	2.66	2.66
LSM	Hiddens	256	256	256	256	256
	Basis Number	12	12	12	12	12
	Tokens	16	24	24	24	48
	Parameter (M)	2.23	2.24	2.24	2.24	2.25
GINO	Modes	[8,8,8]	[8,8,8]	[8,8,8]	[8,8,8]	[8,8,8]
	Layers	4	4	4	4	4
	Hidden Channels	8	8	8	12	12
	Parameter (M)	0.79	0.79	0.79	1.45	1.45
Transolver	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.13	1.13	1.13	1.13	1.14
Transolver++	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.04	1.04	1.04	1.04	1.04
Unisoma	Hiddens	128	128	128	128	128
	Processors	2	2	2	2	2
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	0.94	0.98	0.98	0.88	2.97
LinearNO	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.03	1.03	1.03	1.03	1.04
HNO	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.12	1.13	1.13	1.13	1.14
FuncAttn	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.13	1.13	1.13	1.13	1.13
SolidX	Hiddens	128	128	128	128	128
	Layers	6	6	6	6	6
	Heads	8	8	8	8	8
	Slices	16	24	24	24	48
	Parameter (M)	1.07	1.07	1.07	1.07	1.08

D.3 Sensitivity and Ablation Analyses

Here, we provide detailed sensitivity and ablation analyses, which further demonstrate the robustness of SolidX to its structural configurations and validate the effectiveness of its key design choices. First, we investigate the sensitivity of SolidX to two important structural hyperparameters, including the number of physical bases and the candidate-interface radius. Second, we conduct controlled ablation studies to examine the contributions of the main architectural components of SolidX.

Sensitivity Analysis. Here, we conduct detailed sensitivity analyses on two key structural configurations of SolidX. First, we investigate the effect of the candidate-interface identification radius R, which controls the spatial extent of the interaction regions identified from the loading rigid-body trajectories. The corresponding results are presented in Figure 5 and Table 7. We further analyze the effect of the number of physical bases used by SolidX, with the results shown in Table 6 and Figure 6. For each dataset, the default configurations are determined from the geometric scale and interaction structure of the training data and are fixed without test-set tuning.

(1) Robustness performance without extensive tuning. The default SolidX-specific configurations are selected according to the geometric scale and interaction structure observed in the training data and are kept fixed throughout evaluation, without exhaustive benchmark-specific tuning. As shown in Tables 6–7, SolidX maintains consistently strong performance across a broad range of basis numbers and candidate-interface radii, with only moderate performance variations as these structural hyperparameters change. Moreover, the default configurations are not necessarily the individually optimal choices, yet still yield competitive results across all benchmarks. This robustness suggests that the effectiveness of SolidX does not depend on carefully tuned structural settings, but instead mainly stems from the proposed interaction-structured representation itself.

(2) Robustness to the candidate-interface radius. The results show that SolidX remains robust over a broad range of the candidate-interface radius R, with moderate deviations from the default setting causing only minor changes in prediction performance. This indicates that the interaction modeling does not require a precisely selected radius. When R becomes sufficiently large, the candidate interaction region simply expands to cover the entire deformable body, making the corresponding interaction group less localized without invalidating the interaction modeling.

(3) Robustness to the number of physical bases. SolidX remains stable across different numbers of physical bases, with only limited performance variations over a broad range of settings. More importantly, SolidX consistently outperforms the corresponding baseline under all evaluated basis configurations, indicating that its advantage does not depend on a carefully selected basis number.

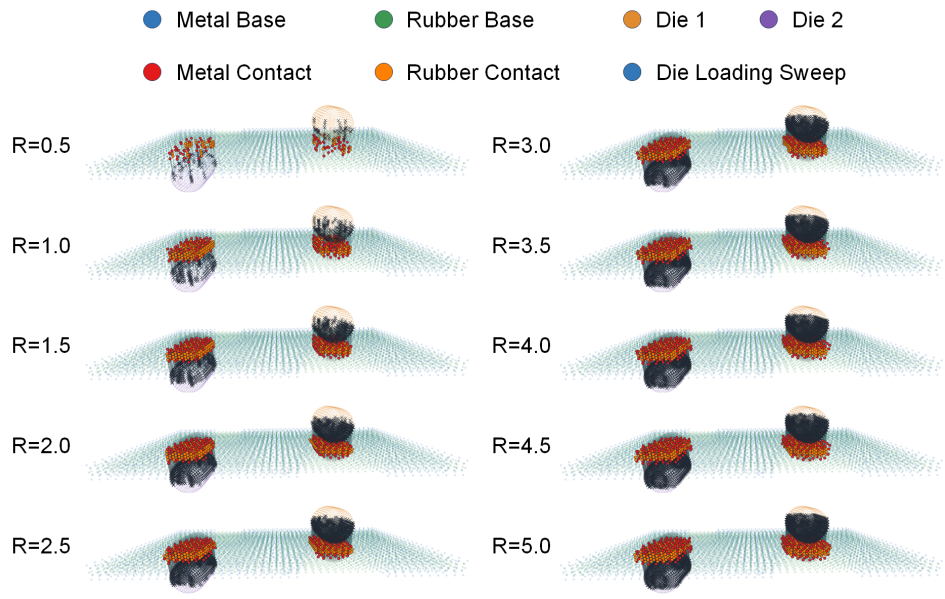


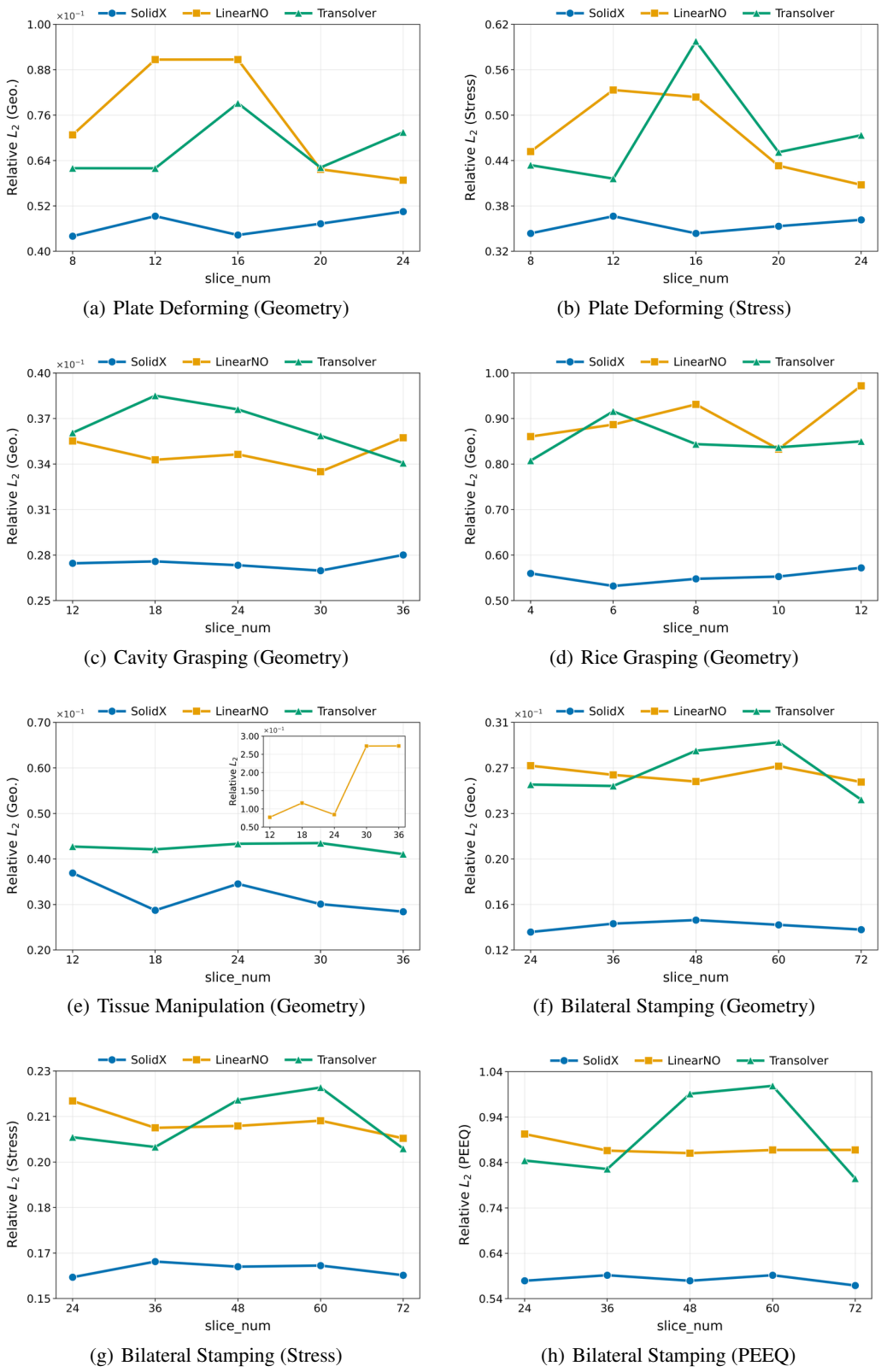
Figure 5: Visualization of candidate interaction-region identification under different thresholds R.

Table 5: Multi-seed performance comparison for representative methods over three independent runs with different random seeds. Relative L2 error is reported. For each method, "Mean" and "Std." denote the mean performance and standard deviation across the three runs, respectively. For the mean results, the best result is in **bold** and the second best is underlined. For Bilateral Stamping, the two columns under each metric correspond to the metal and rubber, respectively.

Method	Stat.	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
		Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
Geo-FNO	Mean	0.1086	0.6526	0.0587	0.7207	0.0563	0.0284	0.0283	0.0458	0.3015	0.6902	0.8626
	Std.	0.0092	0.0051	0.0016	0.0048	0.0024	0.0025	0.0026	0.0006	0.0029	0.0150	0.0100
Transolver	Mean	0.0739	0.5069	0.0363	0.8899	0.0406	0.0141	0.0141	0.0345	0.1836	0.4331	0.5165
	Std.	0.0145	0.0936	0.0019	0.0555	0.0024	0.0004	0.0004	0.0005	0.0050	0.0181	0.0397
Unisoma	Mean	0.1011	0.5910	0.0543	0.8520	0.0467	0.0103	0.0094	0.0339	0.1523	0.4142	0.3573
	Std.	0.0035	0.0344	0.0409	0.0520	0.0015	0.0007	0.0005	0.0004	0.0033	0.0200	0.0116
LinearNO	Mean	0.0898	0.5539	0.0347	0.9326	0.1681	0.0133	0.0134	0.0344	0.1753	0.4190	0.4680
	Std.	0.0034	0.0487	0.0015	0.0199	0.0965	0.0008	0.0006	0.0003	0.0101	0.0204	0.0528
FuncAttn	Mean	0.0734	0.4834	0.0387	0.9367	0.0489	0.0148	0.0149	0.0351	0.1876	0.4641	0.5473
	Std.	0.0096	0.0171	0.0002	0.0475	0.0026	0.0021	0.0022	0.0012	0.0183	0.0496	0.0866
SolidX	Mean	**0.0460**	**0.3500**	**0.0274**	**0.5566**	**0.0296**	**0.0074**	**0.0069**	**0.0309**	**0.1313**	**0.3204**	**0.2602**
	Std.	0.0030	0.0108	0.0001	0.0250	0.0021	0.0003	0.0002	0.0006	0.0023	0.0009	0.0038

Table 6: Sensitivity analysis of the number of bases for SolidX, LinearNO, and Transolver. The basis scale denotes the number of physical bases relative to the default setting, where the default numbers are 16, 24, 24, 24, and 48 for Plate Deforming, Cavity Grasping, Rice Grasping, Tissue Manipulation, and Bilateral Stamping, respectively. The best result is in **bold** and the second best is underlined.

Method	Slice Scale	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
		Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
SolidX	0.50 ×	**0.0440**	**0.3437**	0.0275	0.6175	0.0369	**0.0071**	**0.0064**	0.0298	**0.1277**	0.3133	0.2661
	0.75 ×	0.0493	0.3664	0.0276	0.5930	0.0345	0.0073	0.0069	0.0308	0.1322	0.3246	0.2671
	1.00 ×	0.0443	**0.3437**	0.0273	**0.5478**	0.0287	0.0077	0.0068	0.0304	0.1308	0.3214	**0.2580**
	1.25 ×	0.0473	0.3532	**0.0270**	0.6295	0.0301	0.0072	0.0069	0.0317	0.1299	0.3246	0.2671
	1.50 ×	0.0505	0.3616	0.0280	0.6167	**0.0284**	0.0073	**0.0064**	**0.0297**	0.1285	**0.3101**	0.2589
LinearNO	0.50 ×	0.0708	0.4518	0.0355	0.8604	**0.0770**	0.0135	0.0139	0.0348	0.1846	0.4159	0.4867
	0.75 ×	0.0907	0.5332	0.0343	0.8867	0.1162	0.0133	0.0133	0.0346	0.1754	0.4149	0.4513
	1.00 ×	0.0907	0.5239	0.0346	0.9310	0.0843	0.0131	**0.0129**	**0.0343**	**0.1713**	**0.4056**	**0.4357**
	1.25 ×	0.0617	0.4331	**0.0335**	**0.8327**	0.2726	0.0137	0.0137	0.0350	0.1775	0.4097	0.4578
	1.50 ×	**0.0588**	**0.4078**	0.0357	0.9718	0.2729	**0.0129**	0.0131	0.0348	0.1715	0.4188	0.4489
Transolver	0.50 ×	0.0620	0.4339	0.0361	**0.8073**	0.0427	0.0129	0.0129	0.0347	0.1720	0.3965	0.4478
	0.75 ×	**0.0619**	**0.4160**	0.0385	0.9157	0.0421	0.0132	0.0125	0.0341	0.1692	0.4009	0.4246
	1.00 ×	0.0792	0.5977	0.0376	0.8437	0.0433	0.0138	0.0137	0.0342	0.1828	0.4266	0.5270
	1.25 ×	0.0621	0.4510	0.0359	0.8367	0.0435	0.0146	0.0148	0.0350	0.1891	0.4569	0.5519
	1.50 ×	0.0715	0.4734	**0.0341**	0.8498	**0.0411**	**0.0122**	**0.0123**	**0.0340**	**0.1687**	**0.3882**	**0.4164**



(a) Plate Deforming (Geometry)

(b) Plate Deforming (Stress)

(c) Cavity Grasping (Geometry)

(d) Rice Grasping (Geometry)

(e) Tissue Manipulation (Geometry)

(f) Bilateral Stamping (Geometry)

(g) Bilateral Stamping (Stress)

(h) Bilateral Stamping (PEEQ)

Figure 6: Performance comparison under different numbers of latent slices.

Table 7: Sensitivity analysis of the candidate-interface radius used for explicit interaction modeling in SolidX. The radius scale denotes the candidate-interface radius relative to the default setting, accounting for differences in spatial scale and discretization resolution across benchmarks. The default radius settings are set to 0.01, 0.04, 0.12, 1.5, and 2.0 for Plate Deforming, Cavity Grasping, Rice Grasping, Tissue Manipulation, and Bilateral Stamping, respectively. The best result is in **bold** and the second best is underlined.

Radius Scale	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
	Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
0.20×	0.0511	0.3915	0.0390	0.6109	0.0323	0.0093	0.0086	0.0327	0.1468	0.3526	0.3423
0.40×	0.0472	0.3657	0.0303	0.5415	0.0309	0.0080	0.0075	0.0315	0.1359	0.3284	0.2749
0.60×	0.0457	0.3841	0.0289	**0.5086**	0.0299	0.0078	0.0071	0.0304	0.1326	0.3236	0.2630
0.80×	0.0496	0.3664	0.0274	0.5605	0.0290	0.0076	0.0067	0.0304	0.1333	0.3173	0.2600
0.90×	0.0436	0.3490	**0.0273**	0.5478	**0.0273**	0.0078	0.0067	0.0309	0.1316	0.3190	0.2598
1.00×	0.0443	0.3437	**0.0273**	0.5478	0.0287	0.0077	0.0068	0.0304	**0.1308**	0.3214	0.2580
1.10×	0.0451	0.3570	0.0276	0.5444	0.0318	0.0078	0.0069	0.0309	0.1318	0.3220	0.2565
1.20×	0.0463	**0.3425**	0.0281	0.5578	0.0325	0.0078	0.0069	0.0307	0.1318	0.3214	0.2579
1.40×	0.0446	0.3500	0.0281	0.5682	0.0360	0.0078	0.0069	0.0308	0.1326	0.3209	0.2616
1.60×	0.0470	0.3540	0.0280	0.5546	0.0348	0.0076	0.0069	0.0309	0.1313	0.3204	0.2613
1.80×	0.0444	0.3429	0.0287	0.5921	0.0348	0.0076	0.0068	0.0307	0.1315	0.3174	0.2563
2.00×	0.0467	0.3470	0.0303	0.6007	0.0348	0.0076	0.0067	0.0302	0.1319	0.3164	**0.2549**
3.00×	**0.0431**	0.3550	0.0329	0.6061	0.0348	0.0074	**0.0065**	0.0305	0.1325	0.3119	0.2581
4.00×	0.0462	0.3634	0.0321	0.6242	0.0348	0.0075	0.0067	0.0305	0.1315	0.3170	0.2589
5.00×	0.0472	0.3515	0.0335	0.6464	0.0348	**0.0073**	0.0067	**0.0300**	0.1324	**0.3114**	0.2643

Ablation Analysis. We conduct component-level ablation studies across all five datasets to further examine the main architectural choices of SolidX. The evaluated variants cover additional attention designs, physical conditioning schemes, physical and geometric cues, and the progressive removal of key structural components. Here, PPC, RSP, and HRS denote *Persistent Physical Conditioning*, *Rigid-State Preservation*, and *Hierarchy-Routed Synthesis*, respectively. The detailed results are reported in Table 8. From these results, we make the following observations.

(1) Additional attention is not essential. Adding Transolver-style self-attention or explicit inter-group attention can improve several outputs on Bilateral Stamping. But these additional attention modules provide no improvement on the remaining four datasets and generally lead to higher errors. Moreover, they introduce additional computational overhead. These results indicate that extra attention over the extracted physical modes provides limited benefit and is unnecessary for SolidX.

(2) Simple additive physical conditioning is sufficient. The gated conditioning variant performs close to the default design and slightly improves a few individual outputs, but provides no consistent advantage. FiLM conditioning generally leads to larger errors across multiple datasets and physical quantities. Overall, the simple additive conditioning used in SolidX provides a more balanced choice without introducing additional modulation mechanisms.

(3) Injecting object identity and relative-geometric cues is beneficial. Removing either object identity or relative-distance information generally degrades performance across datasets, although a few individual outputs may show slight improvements. In particular, removing object identity leads to noticeable degradation on Tissue Manipulation and several Bilateral Stamping quantities. These results show that injecting both types of physical cues is useful for distinguishing heterogeneous physical roles and interaction relationships across different multi-solid systems.

(4) The structural components provide complementary benefits. We progressively remove PPC, RSP, and HRS to examine their individual contributions. Overall performance generally degrades as more components are removed, with clear deterioration on several datasets such as Rice Grasping and Tissue Manipulation. These results indicate that PPC, RSP, and HRS each contribute useful and complementary effects to the complete SolidX framework.

(5) Interaction-structured organization provides the most substantial improvement. After further removing the interaction-structured grouping, the model reduces to the homogeneous LinearNO (Hu et al., 2026). This step causes the largest overall performance drop in the progressive ablation, substantially increasing errors across all five datasets. The pronounced degradation highlights the interaction-structured perspective as the central design of SolidX and shows the clear benefit of explicitly organizing physical bases according to heterogeneous interaction structures.

We further examine where to inject the persistent physical context within the analysis–synthesis process, with results reported in Table 9. Specifically, conditioning K affects the construction of the analysis bases, conditioning Q affects the hierarchy-routed synthesis bases, while conditioning V directly modifies the value features. Our default SolidX injects the physical context into Q and K and leaves the value pathway unchanged. From these results, we make the following observations.

(1) Joint conditioning of Q and K provides a balanced choice. The default $Q + K$ configuration achieves strong performance across the five datasets and different physical quantities. Conditioning only Q or K can improve several individual outputs, but these gains are not maintained across all tasks. Similarly, removing persistent conditioning may benefit a few quantities while degrading others. Overall, jointly conditioning Q and K provides a favorable balance across different settings.

(2) The value pathway is better kept free from persistent conditioning. Configurations involving V generally yield larger errors on multiple datasets than the default $Q+K$ design. This suggests that the physical context is more effective for controlling basis construction and routing than for modifying the aggregated value features, supporting our choice to condition only the query and key pathways.

Table 8: Component-level ablation study on five datasets. Relative L2 is reported, with lower values indicating better performance. For Bilateral Stamping, the left and right columns correspond to the metal and rubber, respectively. The best result is in **bold** and the second best is underlined.

Variant	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
	Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
Full SolidX	**0.0443**	0.3437	0.0273	0.5478	0.0287	0.0077	0.0068	0.0304	0.1308	0.3214	**0.2580**
Transolver Attn.	0.0528	0.3543	0.0275	0.5564	0.0305	**0.0067**	**0.0062**	**0.0292**	**0.1264**	**0.3123**	0.2640
Explicit Inter-Group Attn.	0.0477	0.3574	0.0273	0.5590	0.0315	0.0071	0.0063	0.0302	0.1309	0.3142	0.2638
Gated Geo. Injection	0.0452	0.3571	0.0274	0.5475	**0.0280**	0.0077	0.0068	0.0304	0.1307	0.3217	**0.2580**
FiLM Geo. Injection	0.0505	0.3866	0.0307	0.6036	0.0301	0.0080	0.0071	0.0311	0.1349	0.3294	0.2787
w/o Object ID	0.0469	0.3568	**0.0272**	**0.5434**	0.0353	0.0080	0.0071	0.0319	0.1336	0.3321	0.2612
w/o Relative Distance	0.0468	0.3614	0.0280	0.5543	0.0347	0.0078	0.0066	0.0312	0.1302	0.3232	0.2590
w/o PPC	0.0444	**0.3338**	0.0273	0.5449	0.0416	0.0081	0.0070	0.0319	0.1343	0.3289	0.2682
w/o PPC & RSP	0.0459	0.3646	0.0299	0.5755	0.0448	0.0079	0.0073	0.0309	0.1353	0.3241	0.2729
w/o PPC & RSP & HRS	0.0522	0.3584	0.0302	0.6133	0.0482	0.0080	0.0073	0.0319	0.1388	0.3310	0.2686
LinearNO	0.0907	0.5239	0.0346	0.9310	0.0843	0.0131	0.0129	0.0343	0.1713	0.4056	0.4357

Table 9: Ablation study of persistent physical conditioning on five datasets. Relative L2 is reported, with lower values indicating better performance. For Bilateral Stamping, the left and right columns correspond to the metal and rubber. The best result is in **bold** and the second best is underlined.

Variant	Plate Deforming		Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping					
	Geo.	Stress	Geo.	Geo.	Geo.	Geo. (M/R)		Stress (M/R)		PEEQ (M/R)	
Q+K (Default)	0.0443	0.3437	0.0273	0.5478	**0.0287**	0.0077	0.0068	0.0304	0.1308	0.3214	**0.2580**
Q	0.0488	0.3564	**0.0268**	0.5563	0.0379	0.0077	0.0068	0.0313	0.1323	0.3217	0.2739
K	**0.0411**	0.3516	0.0284	0.5867	0.0352	**0.0071**	**0.0063**	**0.0295**	**0.1293**	**0.3124**	0.2593
V	0.0451	0.3579	0.0275	0.6085	0.0379	0.0077	0.0072	0.0314	0.1368	0.3257	0.2810
K+V	0.0434	0.3526	0.0277	0.5945	0.0379	0.0075	0.0067	0.0319	0.1348	0.3309	0.2787
Q+V	0.0441	0.3650	0.0271	0.6223	0.0335	0.0076	0.0072	0.0318	0.1377	0.3300	0.2807
Q+K+V	0.0441	0.3599	**0.0268**	0.6024	0.0343	0.0077	0.0070	0.0311	0.1345	0.3273	0.2726
w/o PPC	0.0444	**0.3338**	0.0273	**0.5449**	0.0416	0.0081	0.0070	0.0319	0.1343	0.3289	0.2682

D.4 SHOWCASES

Here, we provide detailed prediction showcases of SolidX and the two most relevant baselines, Transolver and LinearNO, across all datasets, as shown in Figures 7–13. For each case, the left panel compares the predicted physical fields with the ground truth, while the right panel visualizes the corresponding prediction errors. Overall, SolidX produces predictions that are consistently closer to the ground truth and exhibits substantially smaller local errors than the two baselines. The improvement is particularly clear in regions with strong and highly localized physical responses, where SolidX more accurately recovers both the spatial extent and response magnitude. In contrast, Transolver and LinearNO tend to smooth or misestimate these local responses, with LinearNO generally exhibiting larger deviations.

Interestingly, these high-response regions frequently coincide with or lie near the physical interaction interfaces between different solids. Such regions are particularly challenging because the physical fields can vary rapidly within a small spatial neighborhood as a result of contact and cross-solid interactions. SolidX explicitly models these interaction regions and organizes dedicated physical bases for the corresponding physical groups. Consequently, part of the basis capacity can focus specifically on capturing the localized responses around interaction interfaces, rather than requiring a single homogeneous set of bases to simultaneously represent all spatial regions. This provides a natural explanation for the substantially reduced errors of SolidX in these physically critical regions.

The error maps on the right of these showcases make the advantage of SolidX more evident. SolidX maintains small prediction errors not only around localized high-response and interaction regions, but also over broader parts of the deformable bodies where the physical response propagates away from the interaction interfaces. This suggests that the proposed grouping strategy can capture both types of behavior at the same time. In particular, *loading interaction groups* focus on localized responses around potential contact regions, *boundary interaction groups* capture the effects of constrained boundaries, while *continuum interaction groups* model body-wide deformation responses propagated through the deformable domain. As a result, SolidX can model both localized responses near the interaction interfaces and the broader deformation patterns distributed across each component.

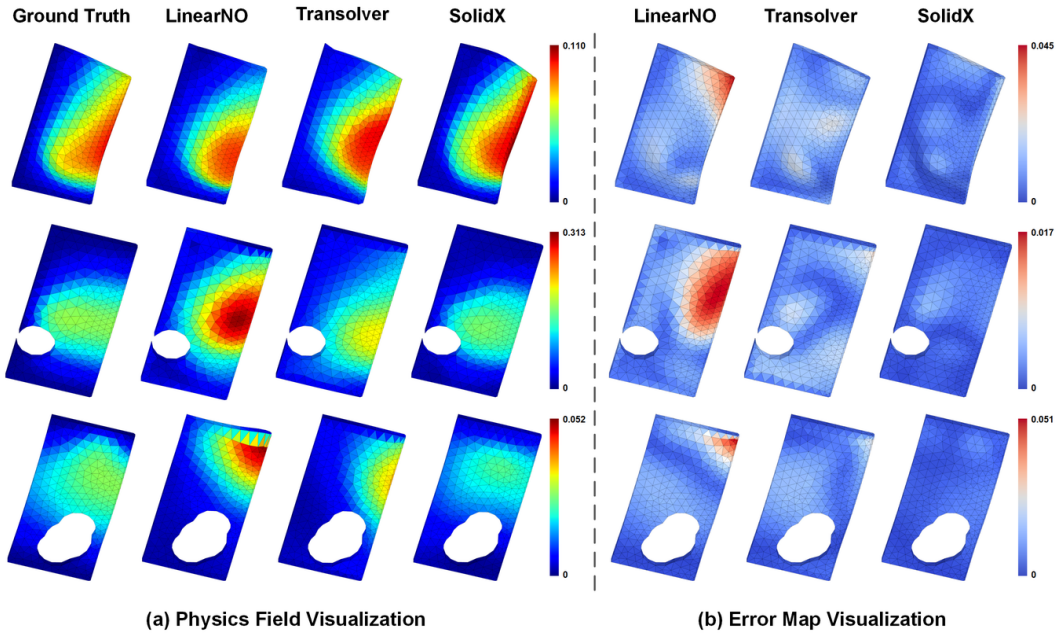


Figure 7: Showcases of Plate Deforming (Geometry). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

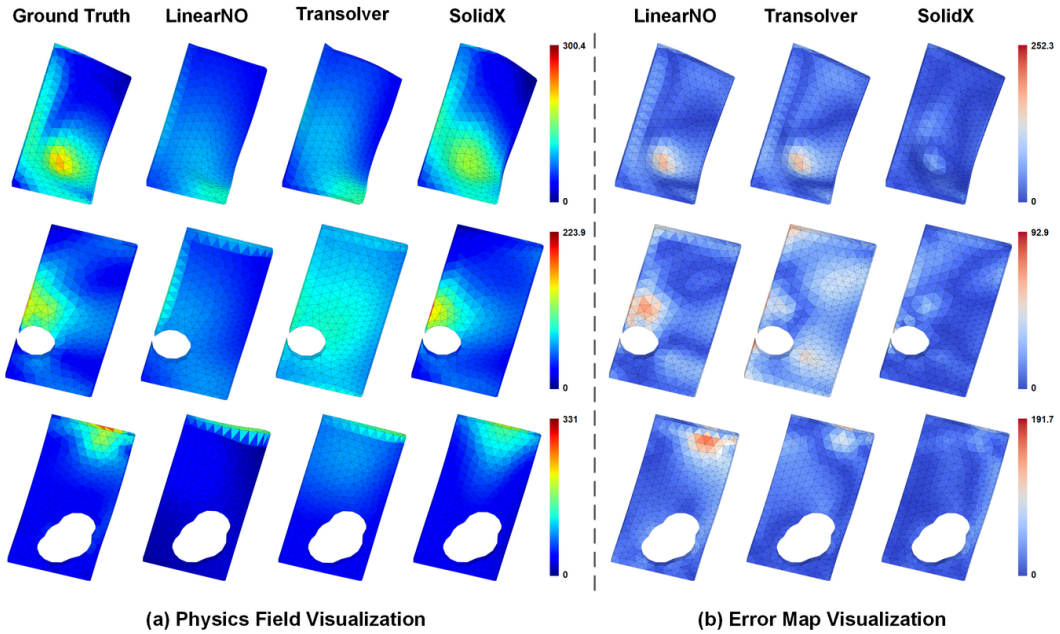


Figure 8: Showcases of Plate Deforming (Stress). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

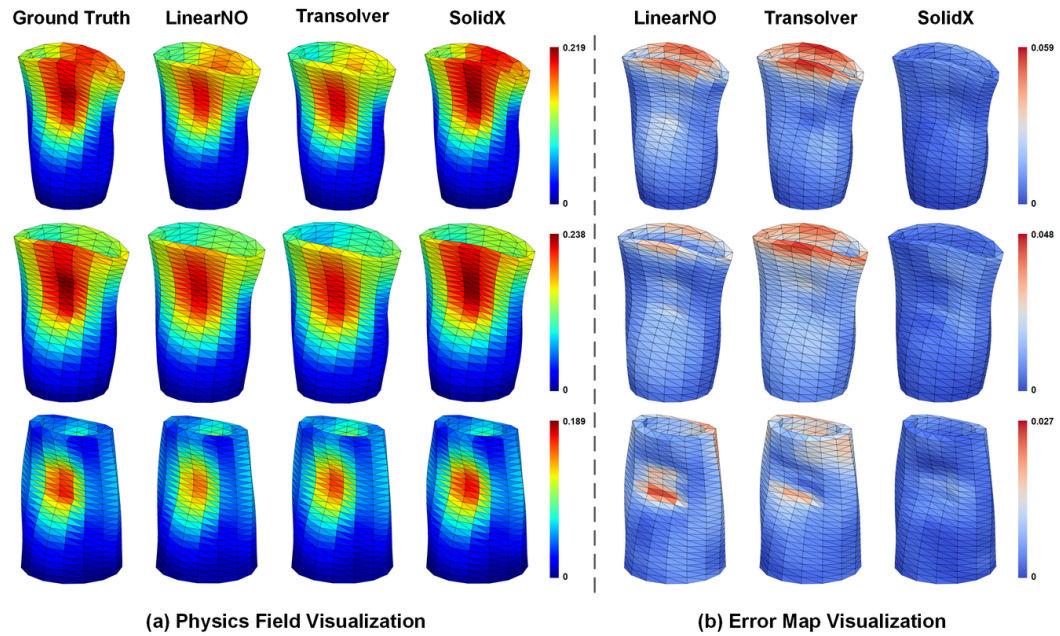


Figure 9: Showcases of Cavity Grasping (Geometry). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

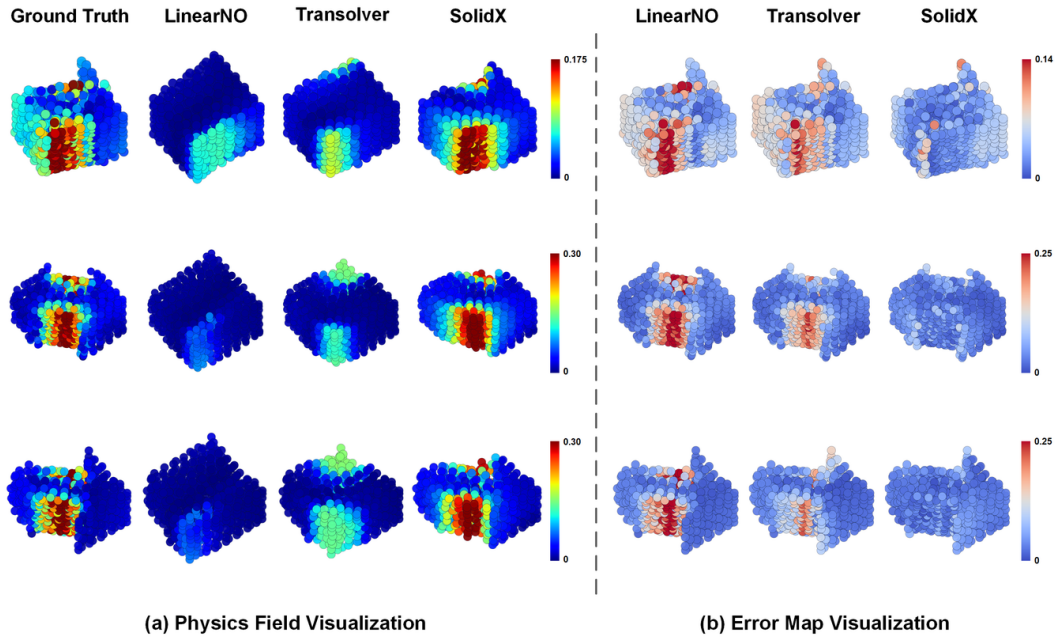


Figure 10: Showcases of Rice Grasping (Geometry). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

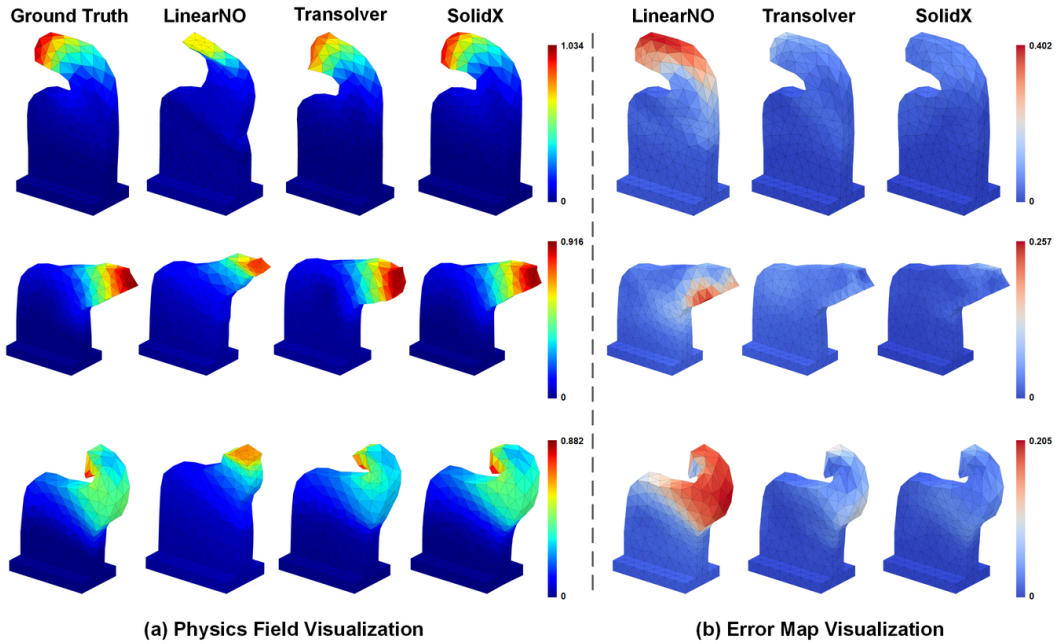


Figure 11: Showcases of Tissue Manipulation (Geometry). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

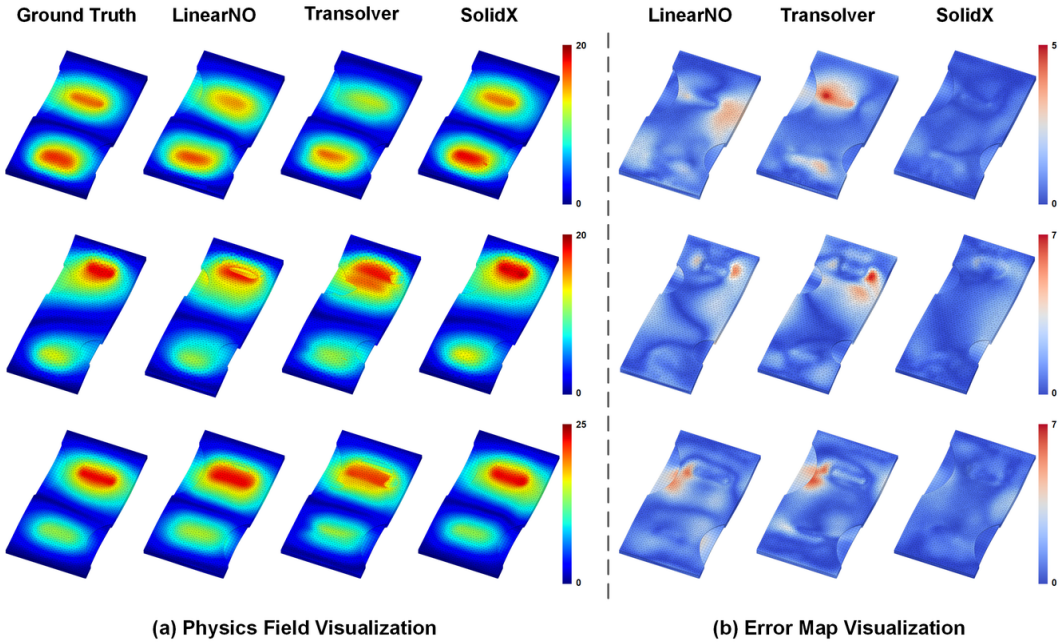


Figure 12: Showcases of Bilateral Stamping (Geometry). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

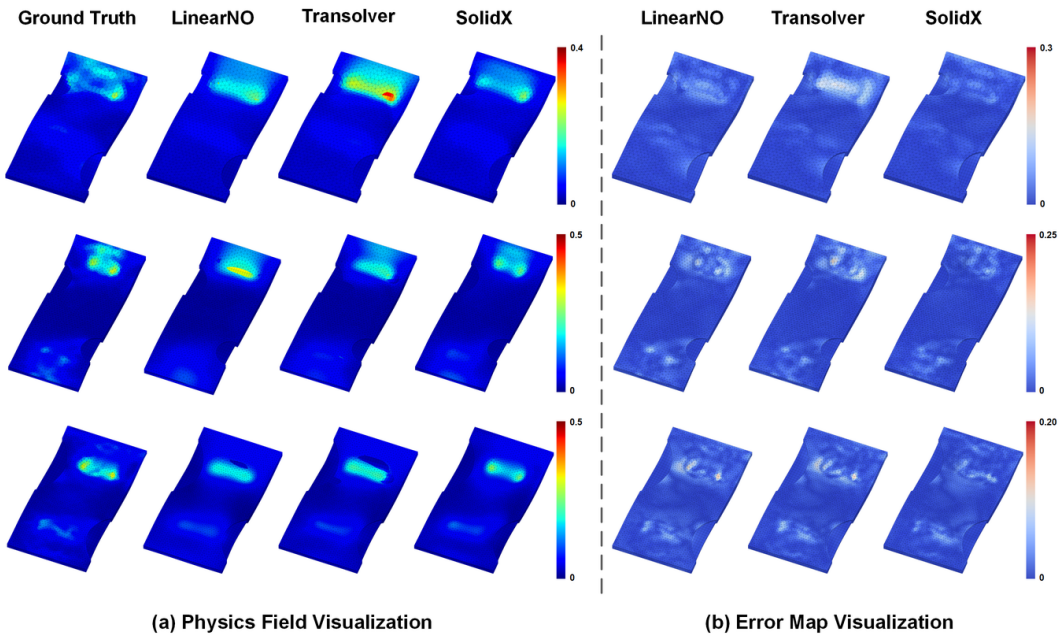


Figure 13: Showcases of Bilateral Stamping (PEEQ). Across three representative samples, the left panel compares the ground truth with predictions from LinearNO, Transolver, and our SolidX, while the right panel shows the corresponding absolute error maps.

D.5 Additional Efficiency Results

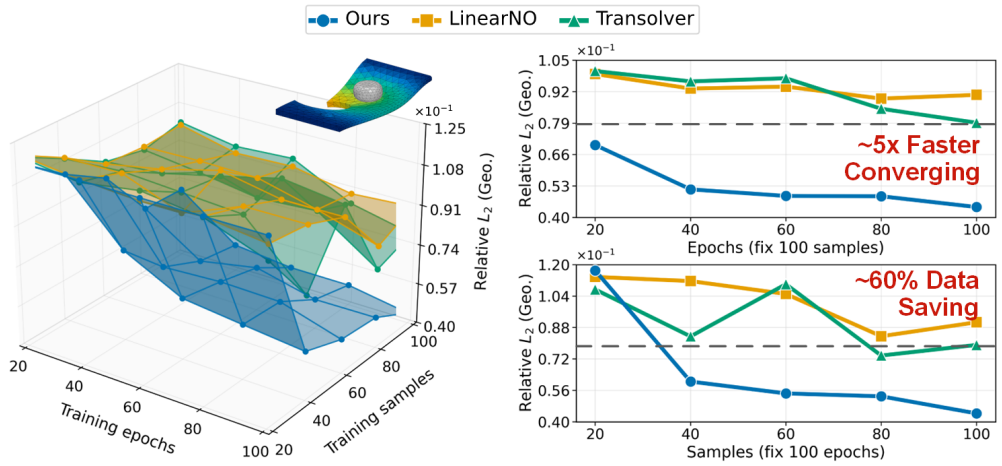


Figure 14: Performance comparison across epochs and samples on Plate Deforming (Geometry).

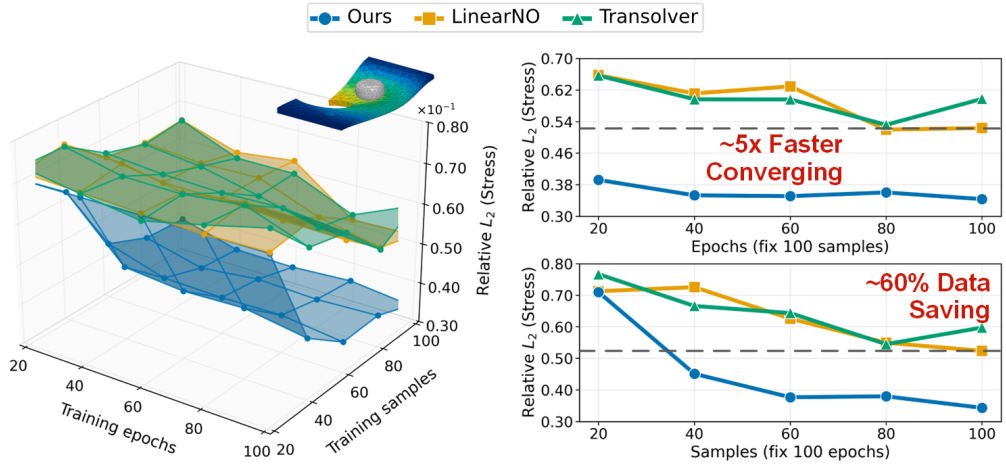


Figure 15: Performance comparison across epochs and samples on Plate Deforming (Stress).

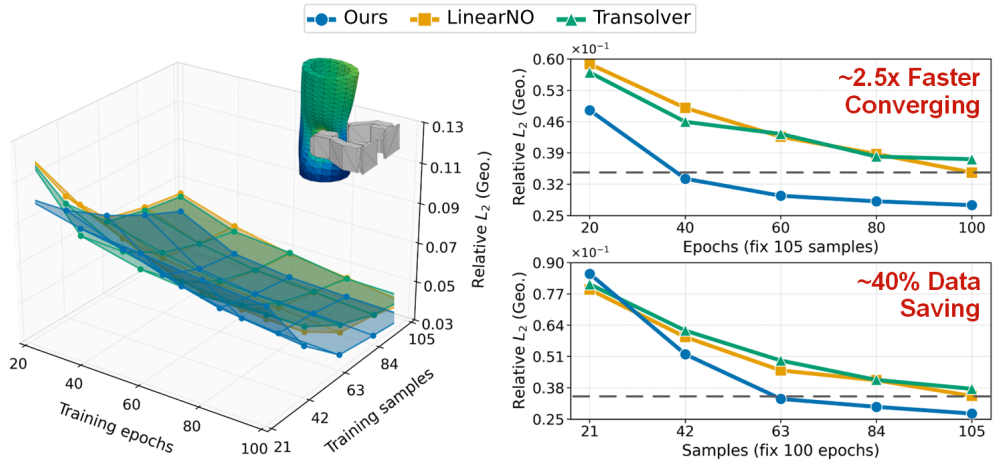


Figure 16: Performance comparison across epochs and samples on Cavity Grasping (Geometry).

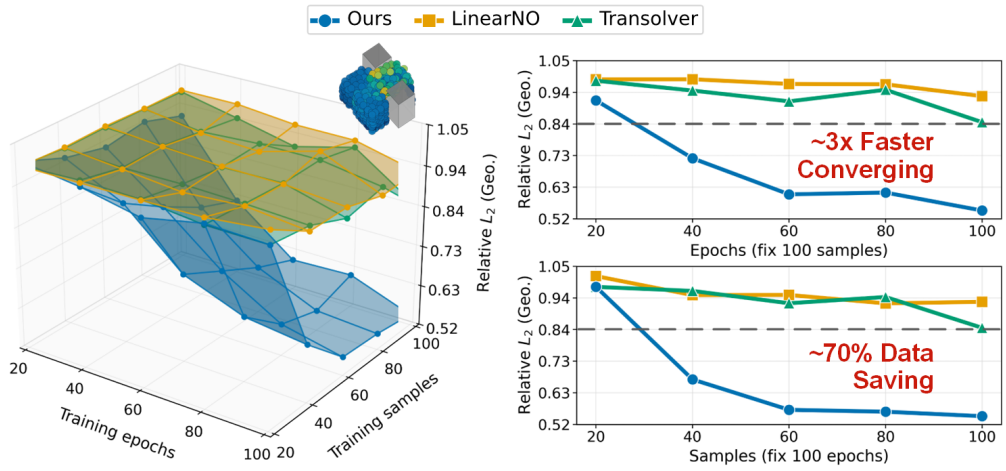


Figure 17: Performance comparison across epochs and samples on Rice Grasping (Geometry).

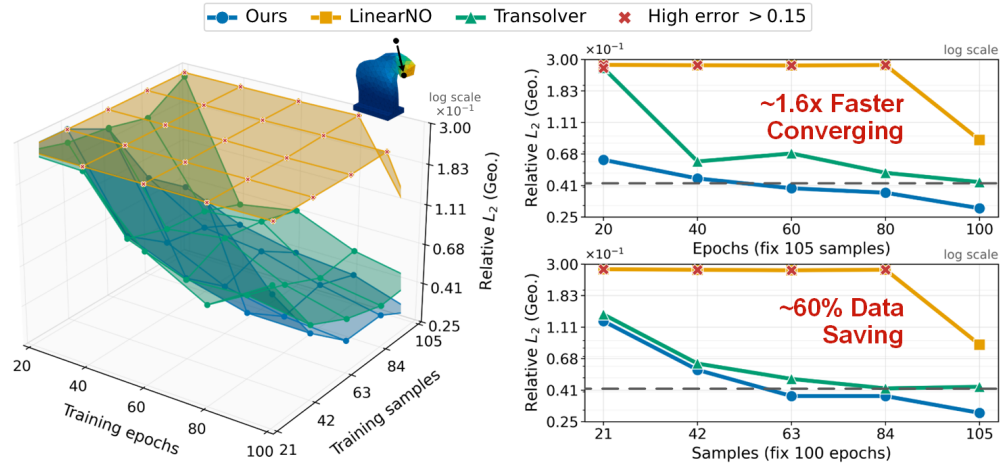


Figure 18: Performance comparison across epochs and samples on Tissue Manipulation (Geometry).

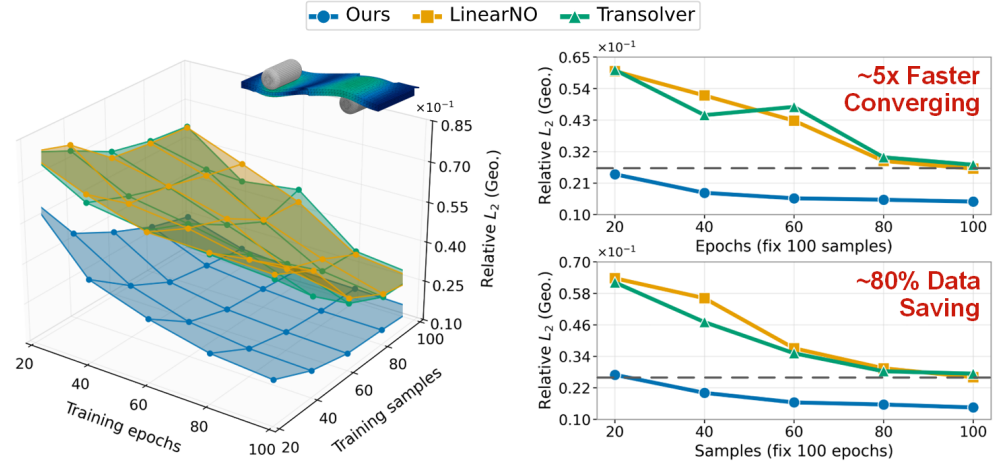


Figure 19: Performance comparison across epochs and samples on Bilateral Stamping (Geometry).

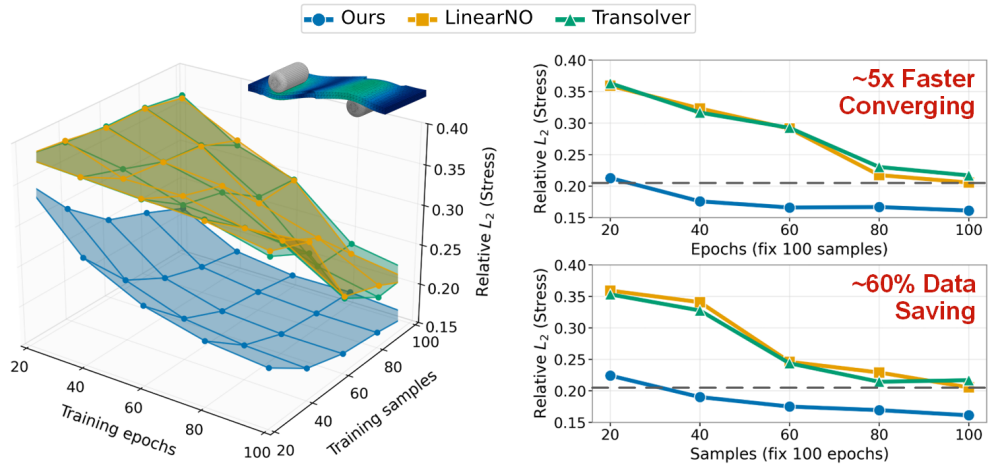


Figure 20: Performance comparison across epochs and samples on Bilateral Stamping (Stress).

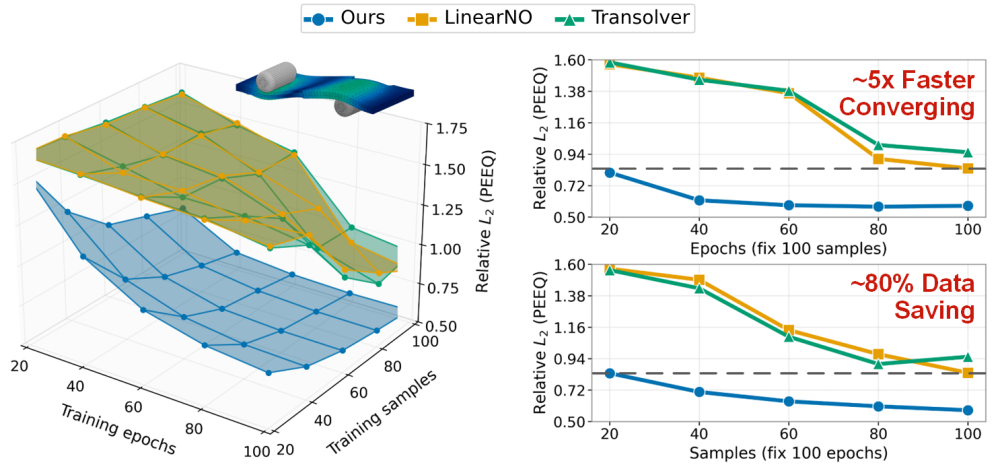


Figure 21: Performance comparison across epochs and samples on Bilateral Stamping (PEEQ).

Table 10: Comparison of parameter count, computational cost, and GPU memory usage between SolidX and representative baselines. The best result is in **bold** and the second best is underlined.

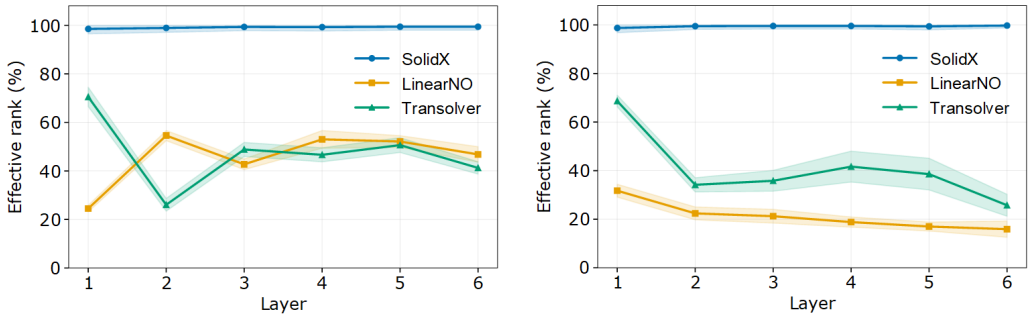
Metric	Model	Plate Deforming	Cavity Grasp.	Rice Grasp.	Tissue Manip.	Bilateral Stamping
Parameter	Transolver	1132948	1133635	1133379	1133891	1136085
	LinearNO	1032244	1033651	1033395	1033907	1038261
	SolidX	1067460	1070499	1070371	1070627	1080133
Computation (GFLOPs)	Transolver	$2.88_{\pm 0.56}$	$3.25_{\pm 0.00}$	$2.61_{\pm 0.18}$	$\underline{0.85}_{\pm 0.00}$	$33.43_{\pm 3.79}$
	LinearNO	$\underline{2.69}_{\pm 0.52}$	$\underline{3.06}_{\pm 0.00}$	$\underline{2.46}_{\pm 0.17}$	$\mathbf{0.80}_{\pm 0.00}$	$\underline{32.09}_{\pm 3.64}$
	SolidX	$\mathbf{2.12}_{\pm 0.46}$	$\mathbf{2.01}_{\pm 0.02}$	$\mathbf{2.00}_{\pm 0.17}$	$0.89_{\pm 0.00}$	$\mathbf{27.34}_{\pm 3.50}$
GPU Memory (MB)	Transolver	$144.51_{\pm 12.24}$	$159.22_{\pm 0.00}$	$\underline{141.36}_{\pm 5.30}$	$98.89_{\pm 0.00}$	$\underline{944.23}_{\pm 104.84}$
	LinearNO	$\underline{142.92}_{\pm 12.24}$	$\underline{157.35}_{\pm 0.00}$	$\mathbf{139.51}_{\pm 5.30}$	$\underline{97.13}_{\pm 0.00}$	$\mathbf{941.72}_{\pm 105.24}$
	SolidX	$\mathbf{96.08}_{\pm 15.85}$	$\mathbf{100.68}_{\pm 0.61}$	$148.34_{\pm 7.61}$	$\mathbf{61.16}_{\pm 0.00}$	$1141.40_{\pm 142.58}$

D.6 Additional Basis Analyses

Here, we provide detailed analyses of the physical bases learned by SolidX to further examine how they are organized under different physical groups. We first analyze the rank properties of $\Phi(Q)\Psi^\top(K)$, then study the similarities of the analysis and synthesis bases, and finally visualize the learned bases and their spatial distributions. Detailed results are presented below.

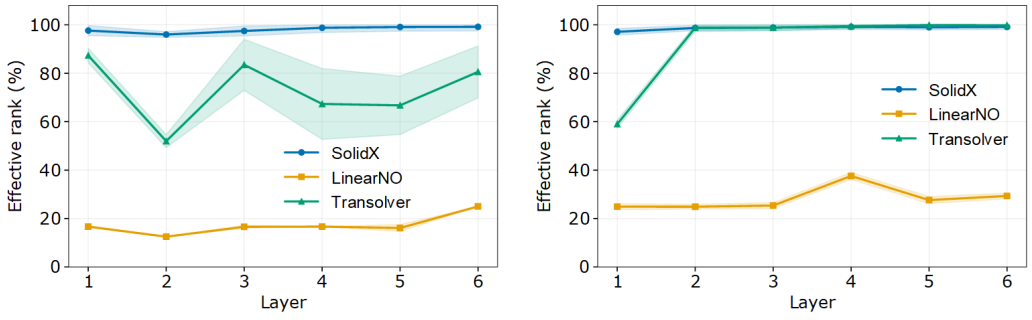
Effective Rank. To examine the learned basis organization, we measure the rank of $\Phi(Q)\Psi^\top(K)$, which reflects the number of linearly independent directions preserved by the analysis–synthesis mapping and the redundancy of the learned representation (Luo et al., 2025; Hu et al., 2026). The detailed results are presented in Figures 22–33. Specifically, Figures 22 and 23 report the effective-rank ratio across different network layers. Figures 24–28 further show the distributions of relative singular values after Singular Value Decomposition (SVD), providing a more detailed view of the effective rank by showing how many singular-value directions remain significant. Figures 29–33 further report the effective-rank ratio under different relative singular-value thresholds.

Across these results, SolidX consistently maintains a high effective rank, with the effective-rank ratio remaining close to 100% in most layers. The singular-value distributions also show that more directions remain active, and this tendency remains stable across different thresholds. These results indicate that the explicit physical grouping in SolidX reduces redundancy among the learned bases. By assigning different basis families to different physical groups, the bases can focus on different interaction and component regions and capture more complementary physical information, rather than collapsing into similar directions within a shared homogeneous basis space.



(a) Cavity Grasping (b) Rice Grasping

Figure 22: Effective rank of the interaction matrix across network layers. SolidX consistently maintains higher effective ranks than LinearNO and Transolver on (a) Cavity Grasping and (b) Rice Grasping, indicating that more independent interaction modes are effectively retained.



(a) Tissue Manipulation (b) Bilateral Stamping

Figure 23: Effective rank of the interaction matrix across network layers. SolidX consistently maintains higher effective ranks than LinearNO and Transolver on (a) Tissue Manipulation and (b) Bilateral Stamping, indicating that more independent interaction modes are effectively retained.

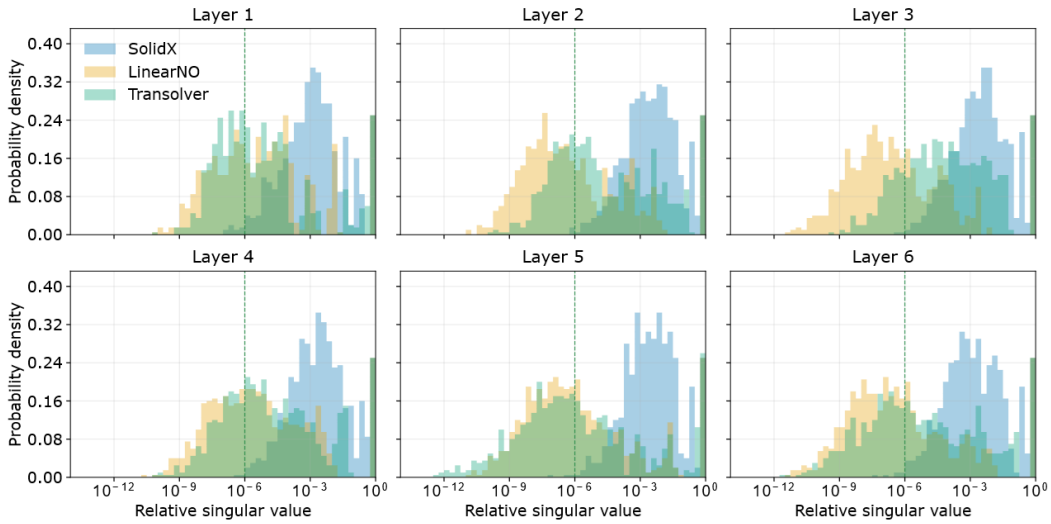


Figure 24: Relative singular-value distributions across different layers on Plate Deforming.

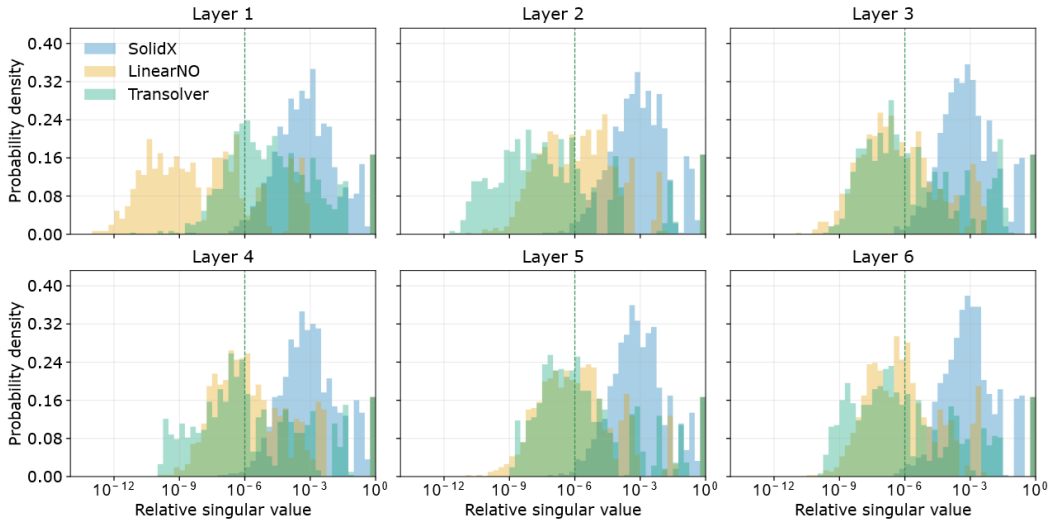


Figure 25: Relative singular-value distributions across different layers on Cavity Grasping.

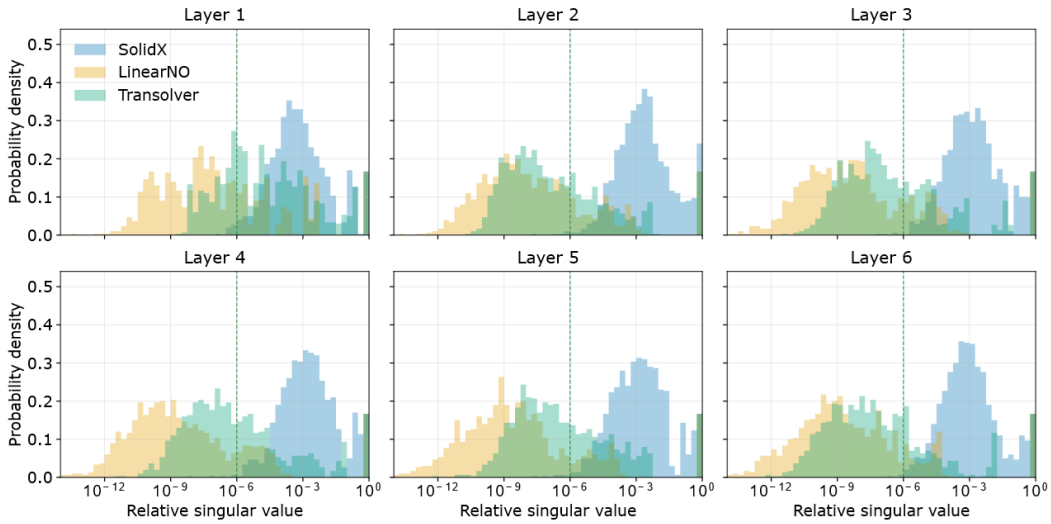


Figure 26: Relative singular-value distributions across different layers on Rice Grasping.

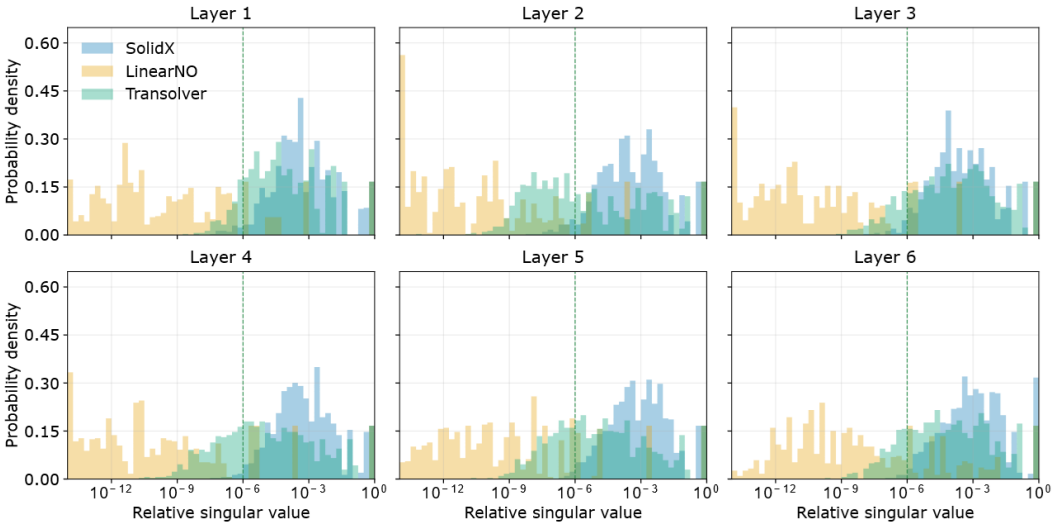


Figure 27: Relative singular-value distributions across different layers on Tissue Manipulation.

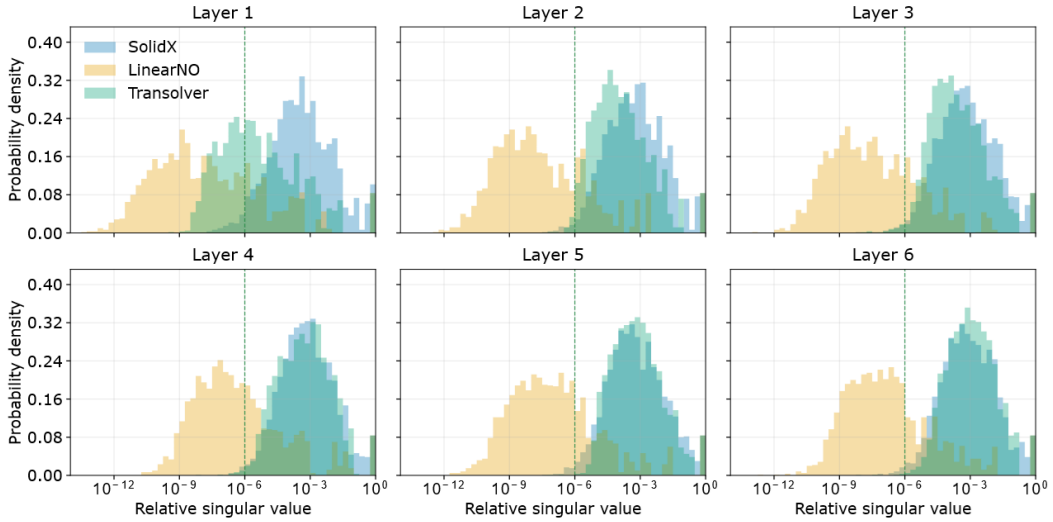


Figure 28: Relative singular-value distributions across different layers on Bilateral Stamping.

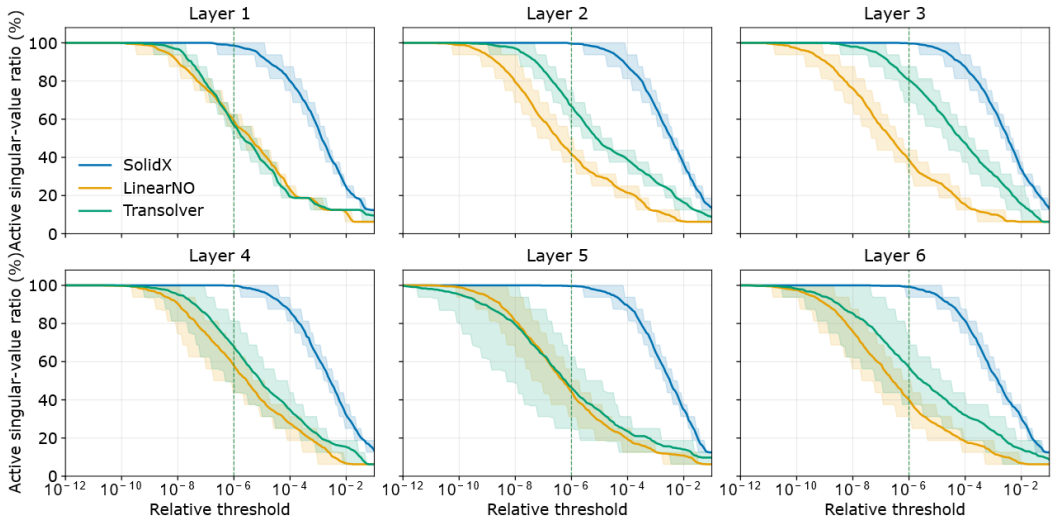


Figure 29: Active singular-value counts across different thresholds and layers on Plate Deforming.

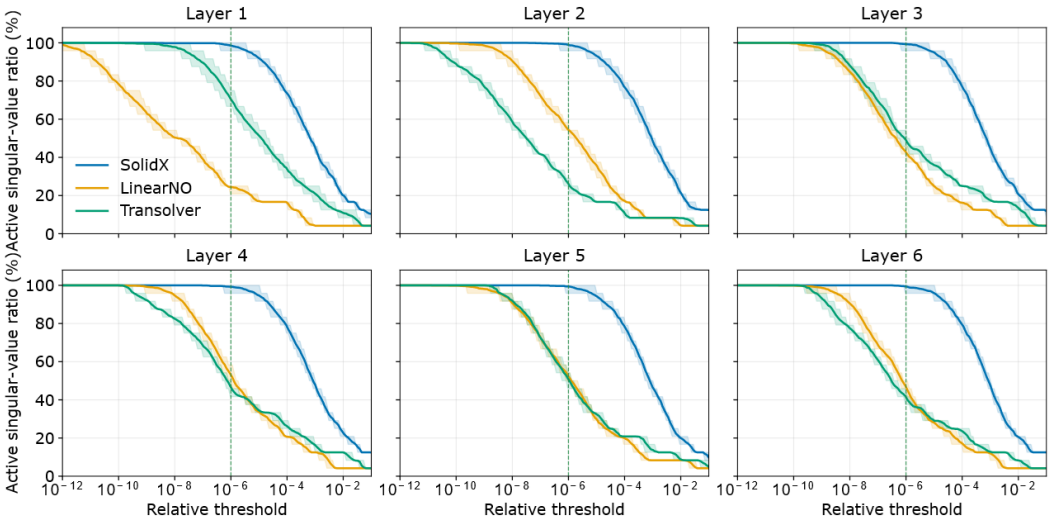


Figure 30: Active singular-value counts across different thresholds and layers on Cavity Grasping.

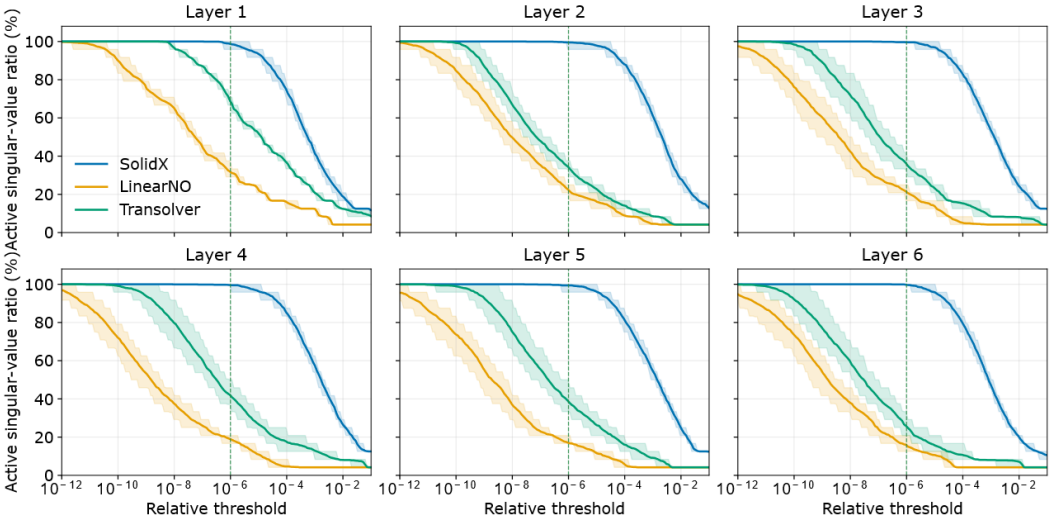


Figure 31: Active singular-value counts across different thresholds and layers on Rice Grasping.

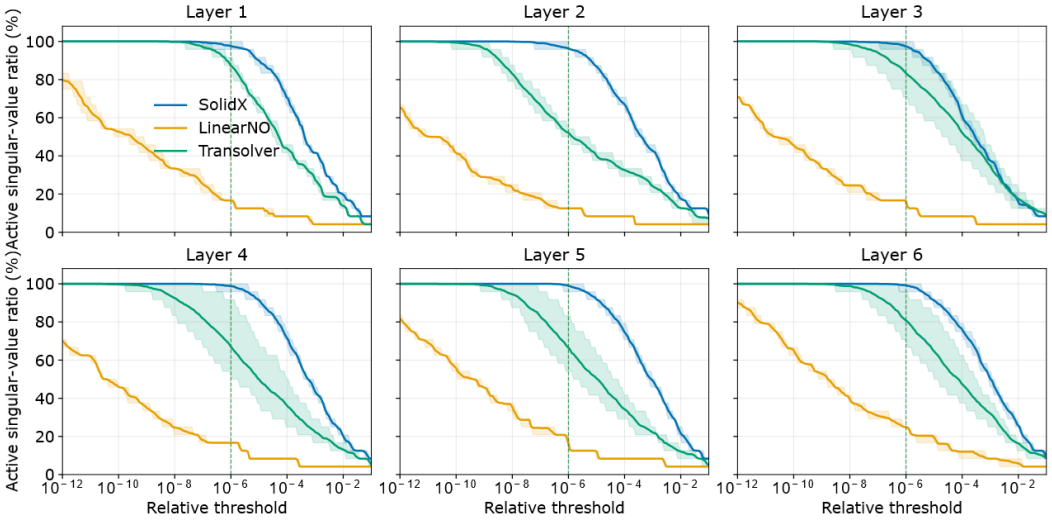


Figure 32: Active singular-value counts across different thresholds and layers on Tissue Manipulation.

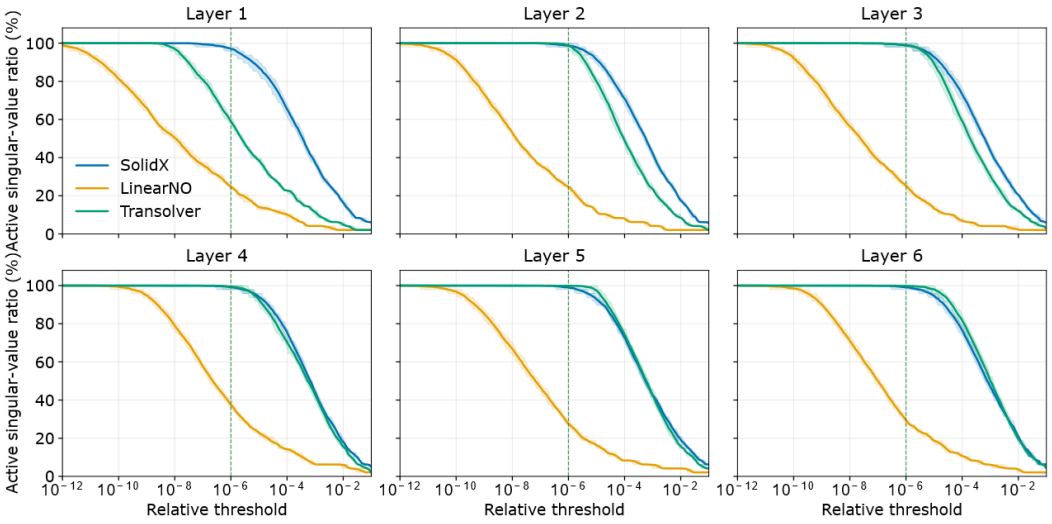


Figure 33: Active singular-value counts across different thresholds and layers on Bilateral Stamping.

Analysis–Synthesis Similarity Analysis. We further analyze the similarities induced by the learned analysis and synthesis bases through $\Phi(Q)\Psi^\top(K)$. Rather than directly comparing individual basis functions, we compute pairwise similarities, measured by Pearson correlation, between the row-wise and column-wise interaction profiles of A. Specifically, each row describes how a spatial point on the synthesis side interacts with all source points through the complete analysis–synthesis mapping, while each column describes how a source point on the analysis side contributes to all target points. Therefore, similarities between rows and columns characterize the structural similarities induced by the synthesis and analysis basis families, respectively. Higher similarity indicates more homogeneous effective interaction profiles, whereas lower similarity indicates stronger differentiation. The complete results are shown in Figure 34.

Compared with SolidX, LinearNO and Transolver exhibit uniformly high similarities in their interaction profiles, indicating largely homogeneous structures and considerable redundancy. In contrast, SolidX shows clearer group-wise differentiation on both the analysis and synthesis sides. Profiles within the same physical group are generally more similar, whereas those across groups are substantially more distinct. This within-group similarity and between-group differentiation is consistent with the design of SolidX, where different groups capture distinct physical roles and regions. These results further support the effectiveness of explicitly organizing physical bases according to the physical grouping of multi-solid systems.

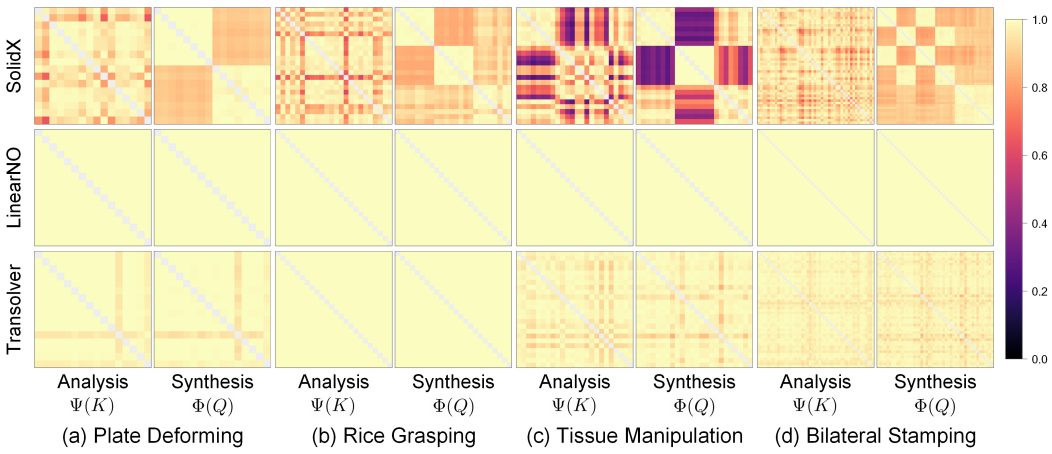


Figure 34: Basis correlation analysis across four datasets, comparing the similarities of the learned analysis and synthesis bases for SolidX, LinearNO, and Transolver.

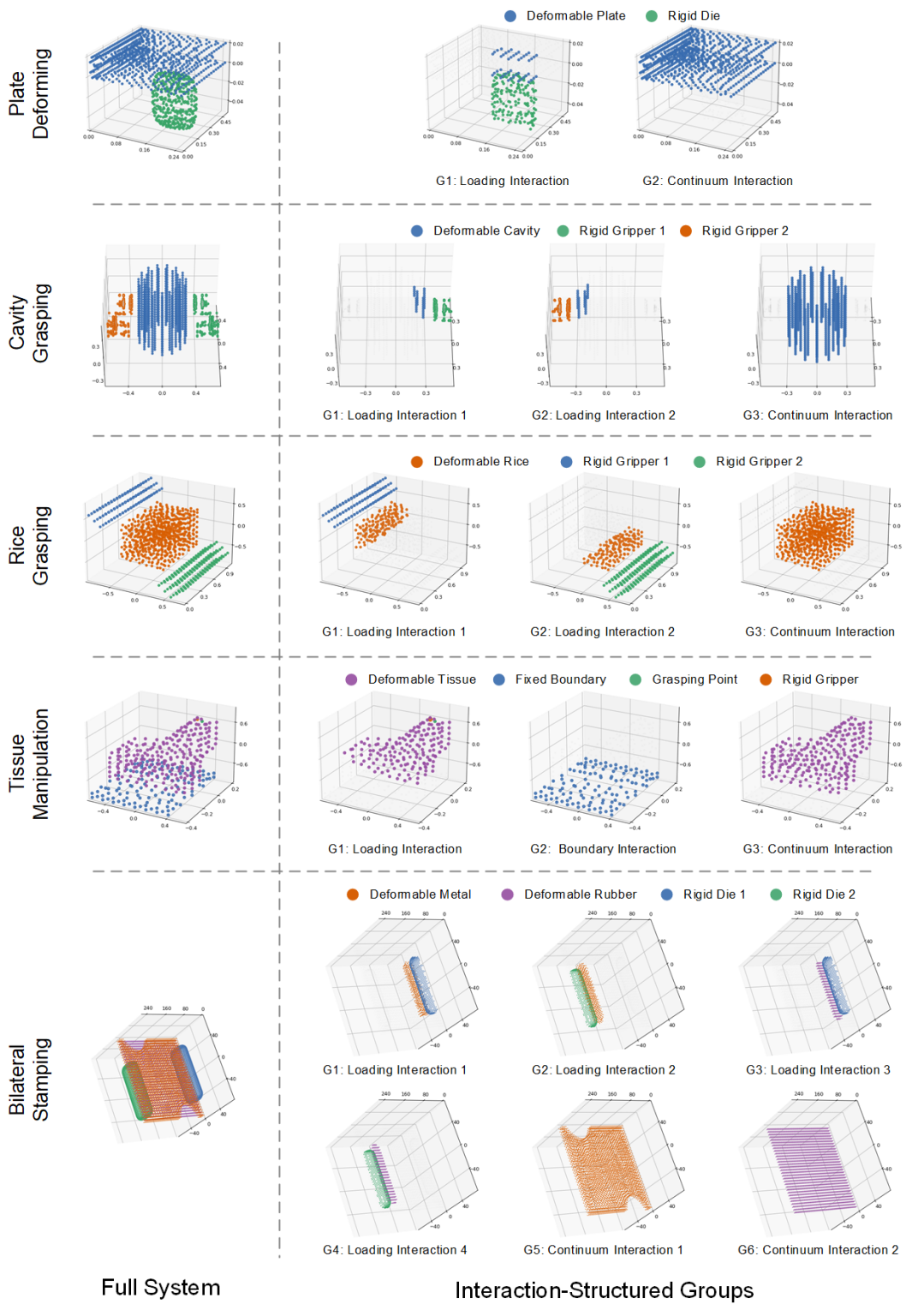


Figure 35: Visualization of interaction-structured physical groups across five multi-solid datasets. The left column shows the complete physical systems, while the right shows the groups organized according to distinct loading, continuum, and boundary roles.

Basis Visualization. Here, we further visualize the learned analysis and synthesis bases of SolidX and compare them with those of LinearNO and Transolver to directly examine how different methods organize their bases over the physical domain. The complete visualizations for the first four datasets are shown in Figures 36–51. We omit Bilateral Stamping from these figures because this more complex dataset uses 48 physical bases (i.e., slices) for SolidX and the baselines, which would result in substantially larger visualizations and further increase the already considerable length and file size of the appendix. We nevertheless observe the same basis behaviors on Bilateral Stamping as on the other datasets. For each reported dataset, we visualize the analysis bases at layers 0 and 5, followed by the synthesis bases at layers 0 and 5. For SolidX, the bases are arranged according to their physical groups, with each row corresponding to one group and labeled as G1, G2, G3, etc. We additionally use red circles to highlight representative regions primarily emphasized by different groups, making their spatial differences easier to observe.

The difference between SolidX and the baselines is clear. For LinearNO and Transolver, different slices or bases exhibit highly similar and spatially averaged patterns, indicating substantial overlap in the information captured by different bases. In contrast, the bases learned by SolidX show much clearer specialization: different physical groups focus on different spatial regions, while bases within the same group exhibit more consistent behaviors. Such specialization becomes particularly evident in the highlighted regions, where different groups concentrate on distinct interaction or component regions. These results further show that explicit physical grouping encourages the learned bases to capture complementary physical information rather than repeatedly modeling the same regions.

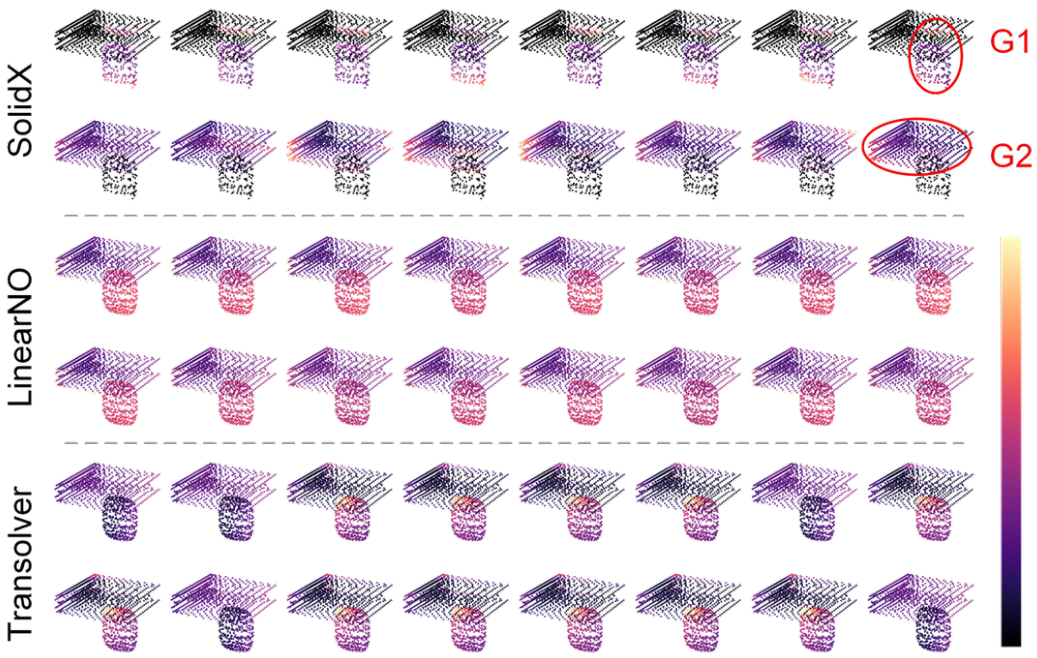


Figure 36: Visualization of the learned analysis bases $\Psi(K)$ for Plate Deforming at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.



Figure 37: Visualization of the learned analysis bases $\Psi(K)$ for Plate Deforming at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

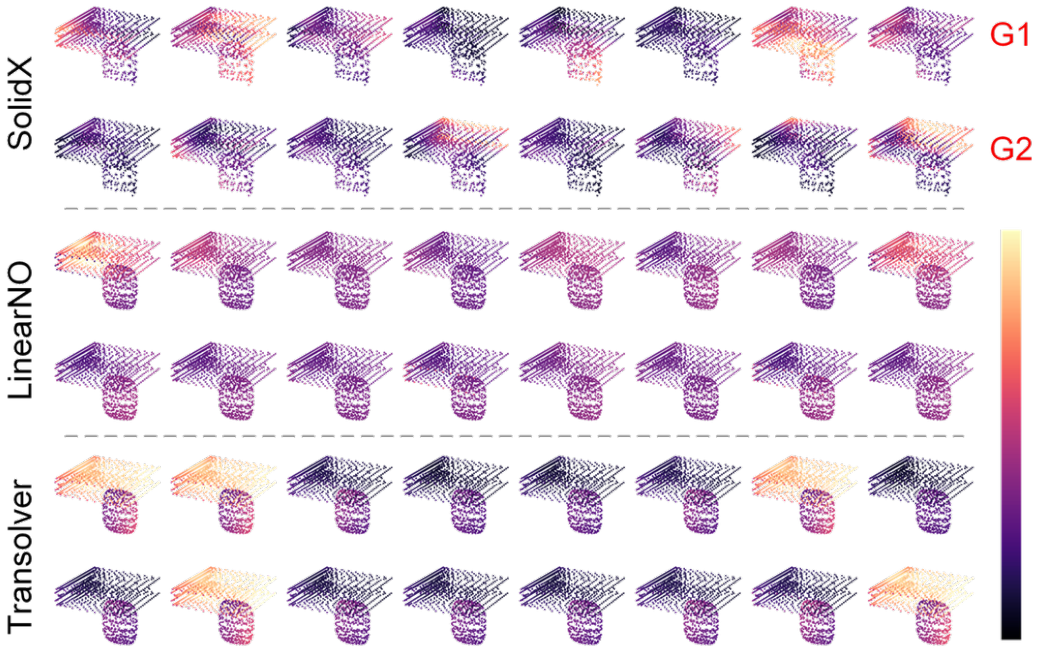


Figure 38: Visualization of the learned synthesis bases $\Phi(Q)$ for Plate Deforming at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

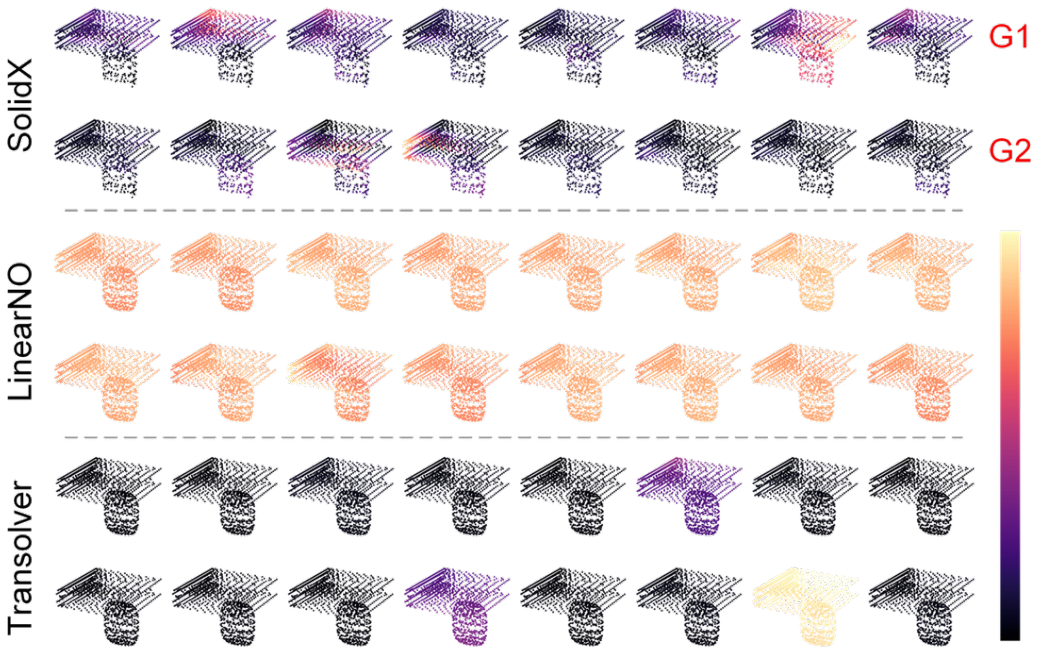


Figure 39: Visualization of the learned synthesis bases $\Phi(Q)$ for Plate Deforming at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

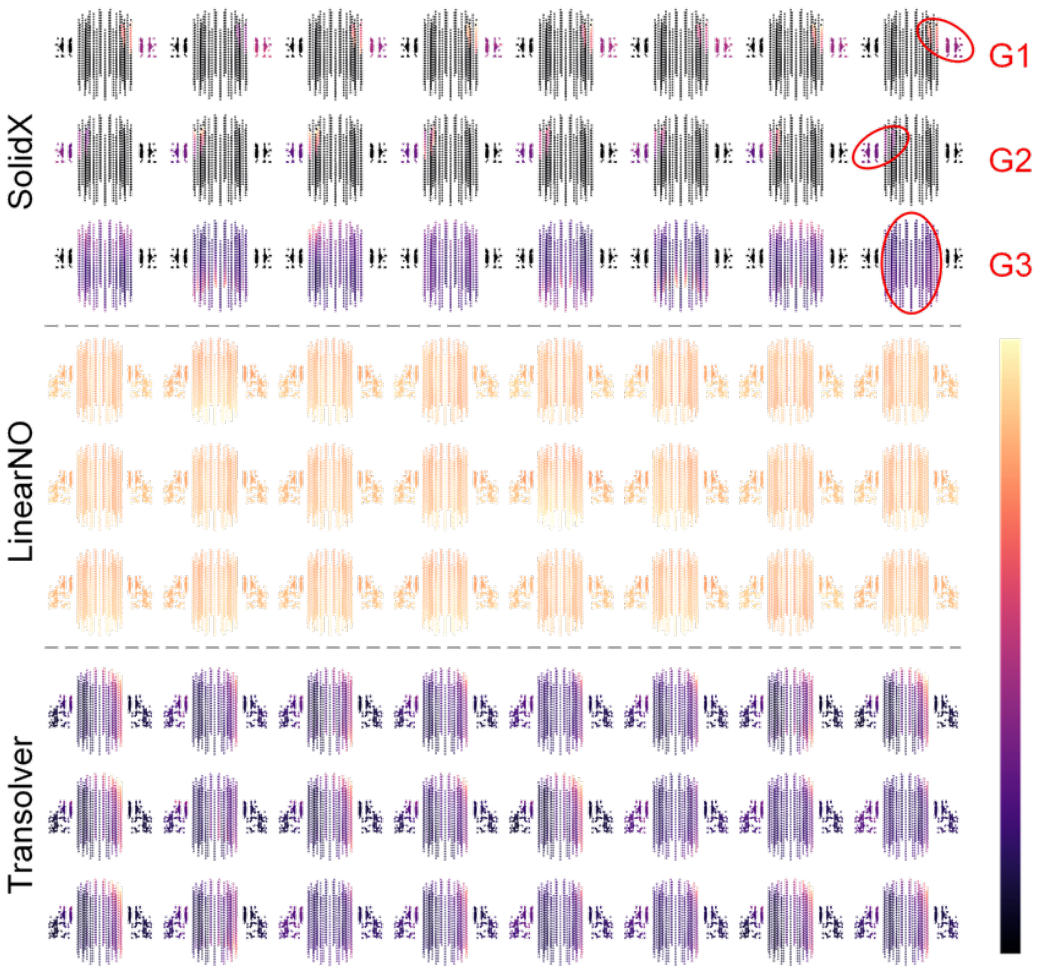


Figure 40: Visualization of the learned analysis bases $\Psi(K)$ for Cavity Grasping at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

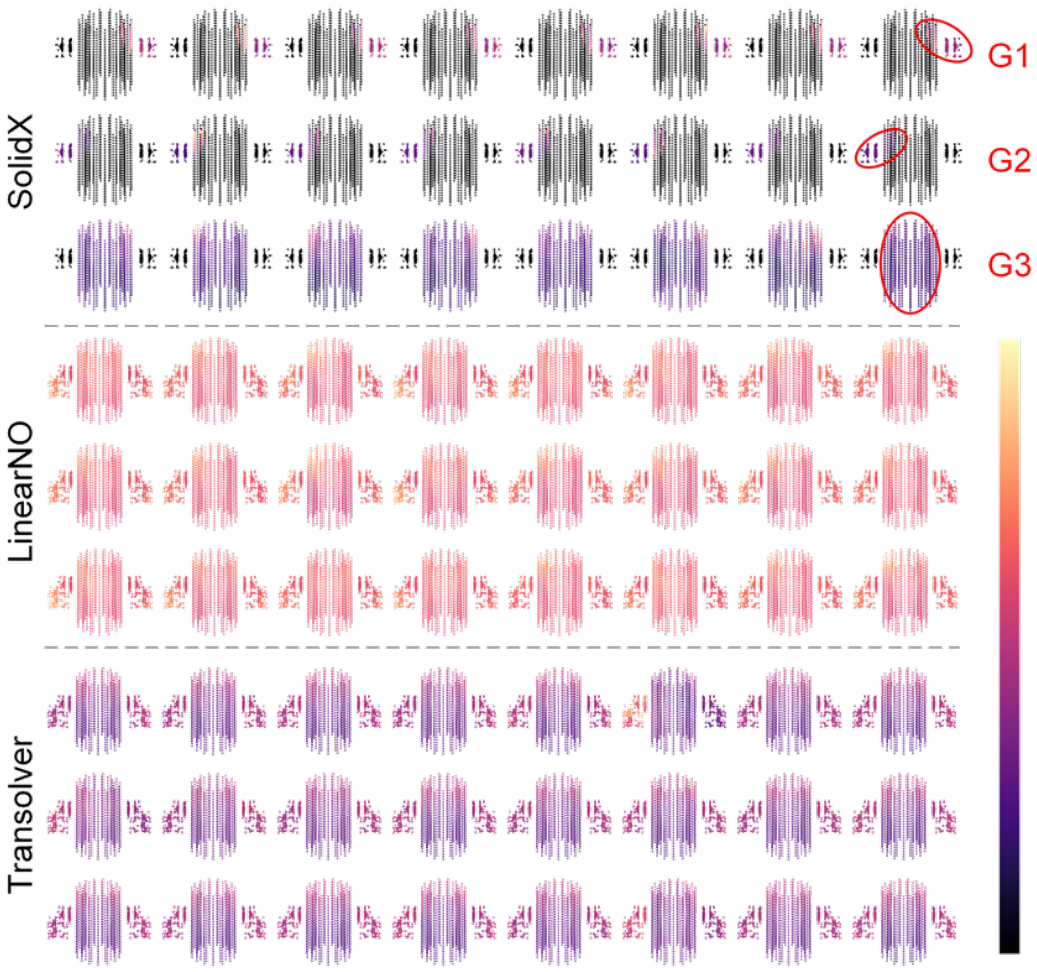


Figure 41: Visualization of the learned analysis bases $\Psi(K)$ for Cavity Grasping at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

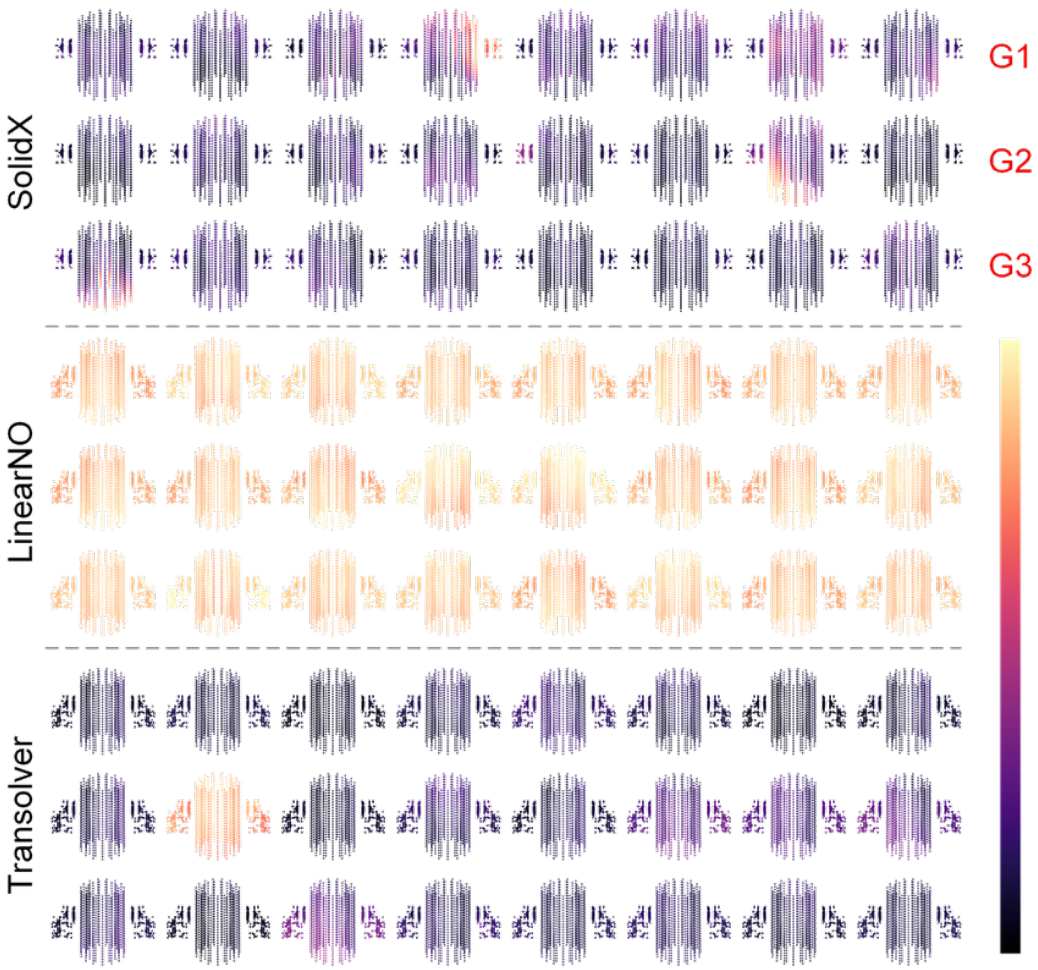


Figure 42: Visualization of the learned synthesis bases $\Phi(Q)$ for Cavity Grasping at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

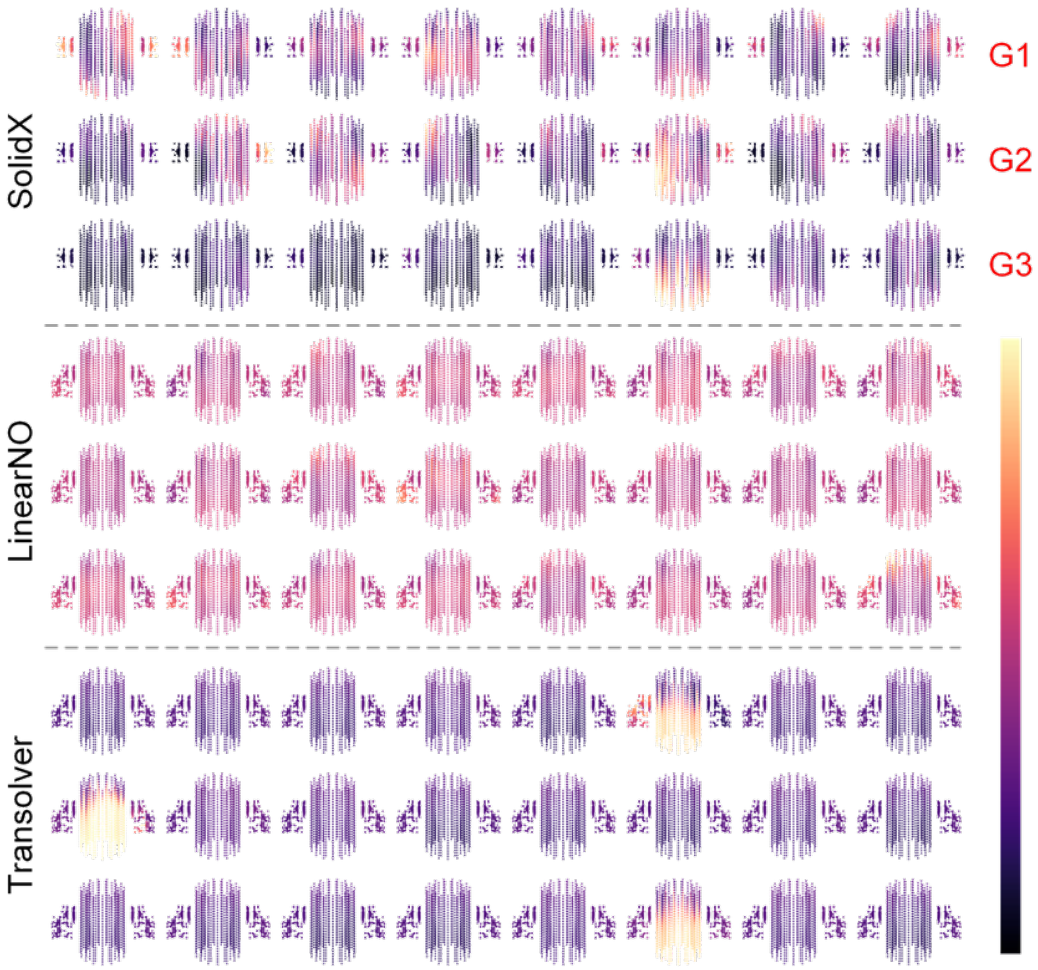


Figure 43: Visualization of the learned synthesis bases $\Phi(Q)$ for Cavity Grasping at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

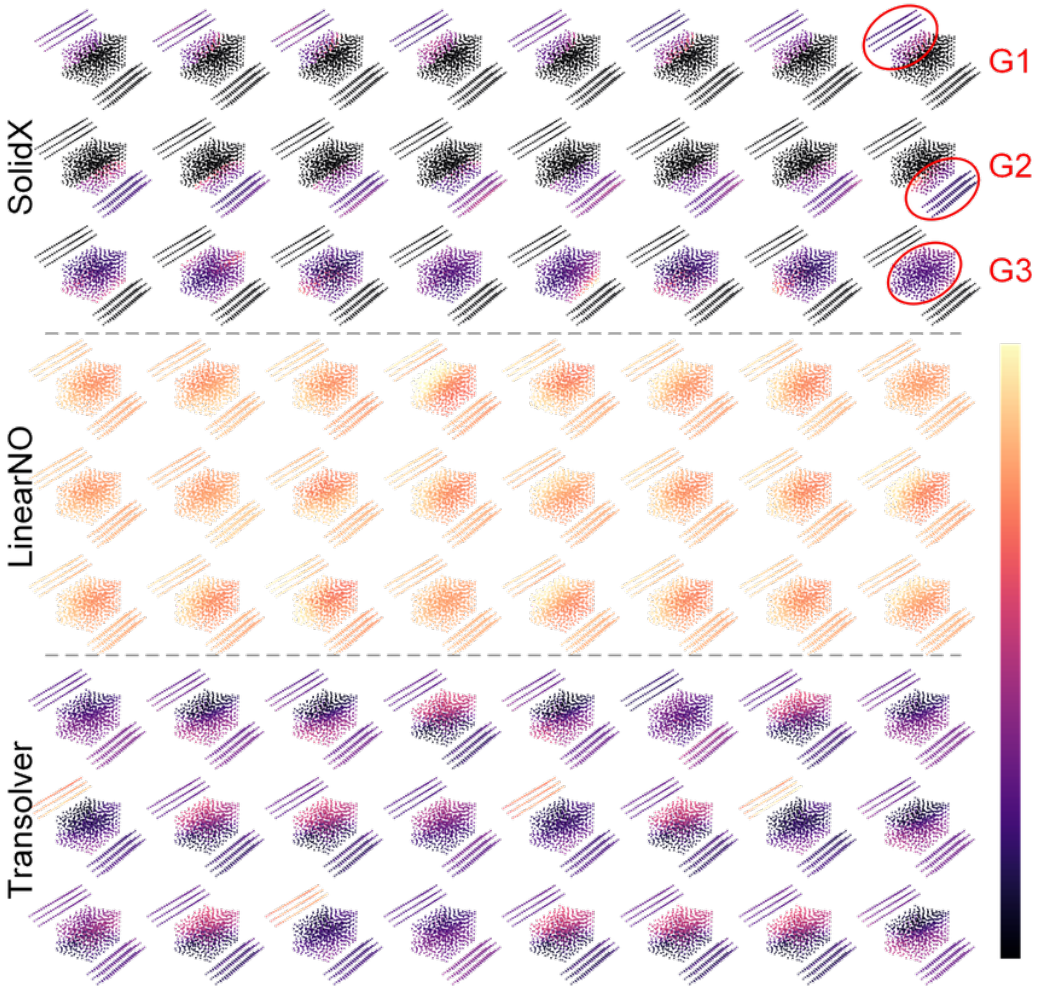


Figure 44: Visualization of the learned analysis bases $\Psi(K)$ for Rice Grasping at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

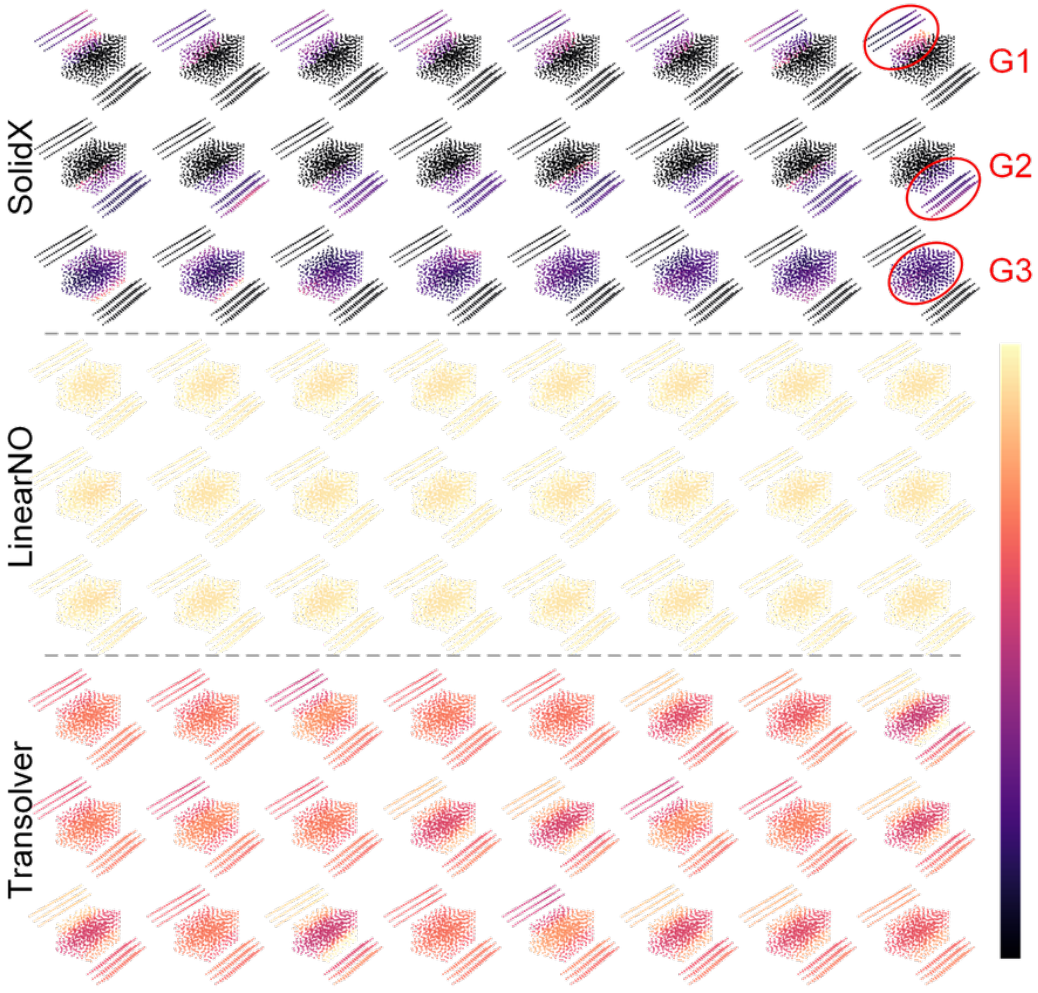


Figure 45: Visualization of the learned analysis bases $\Psi(K)$ for Rice Grasping at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

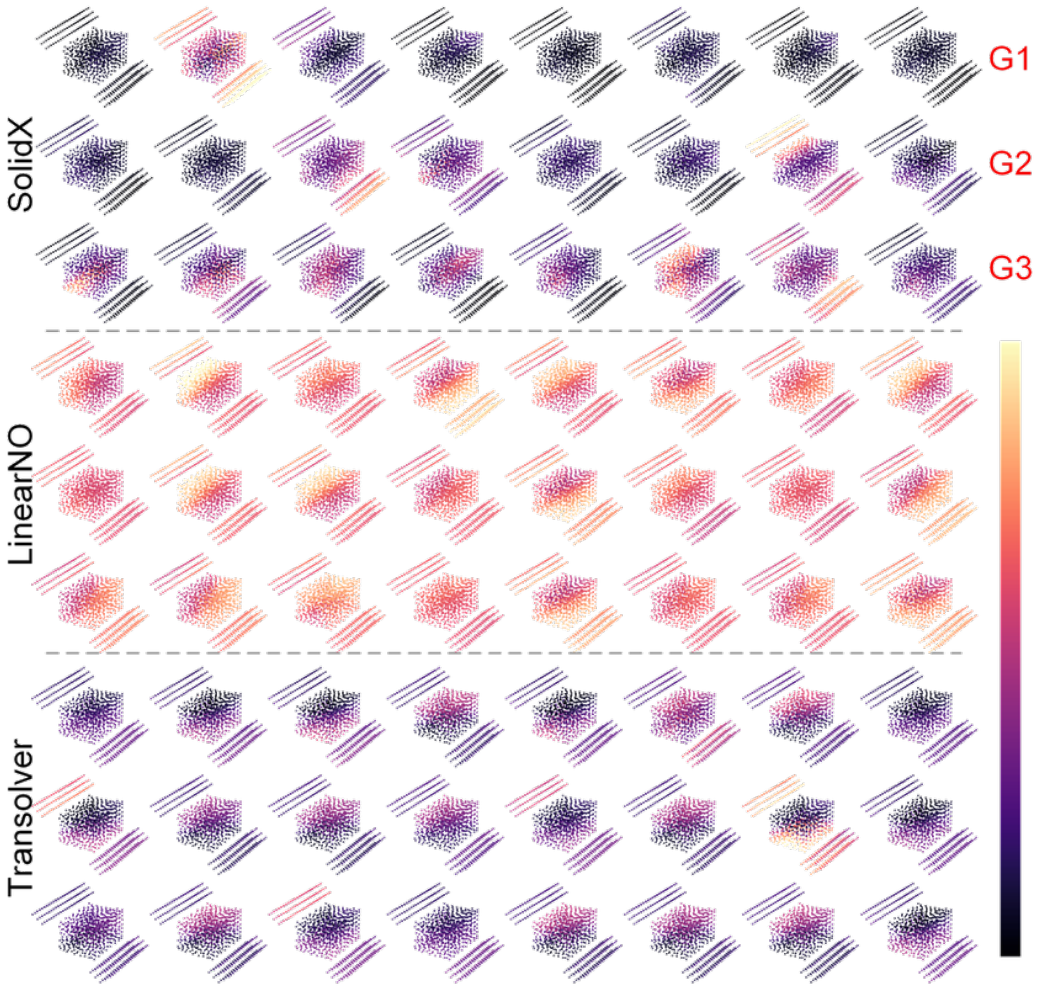


Figure 46: Visualization of the learned synthesis bases $\Phi(Q)$ for Rice Grasping at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

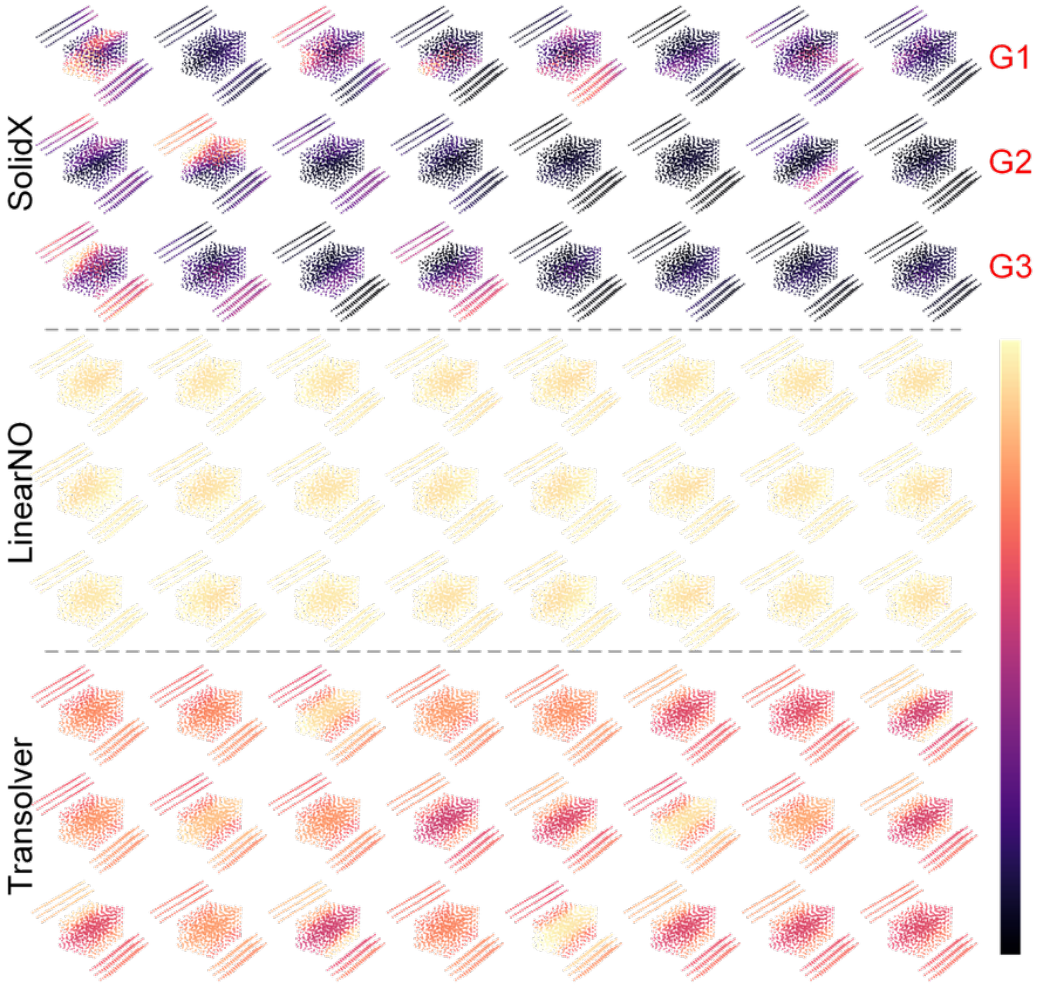


Figure 47: Visualization of the learned synthesis bases $\Phi(Q)$ for Rice Grasping at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.



Figure 48: Visualization of the learned analysis bases $\Psi(K)$ for Tissue Manipulation at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.



Figure 49: Visualization of the learned analysis bases $\Psi(K)$ for Tissue Manipulation at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.



Figure 50: Visualization of the learned synthesis bases $\Phi(Q)$ for Tissue Manipulation at layer 0. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.

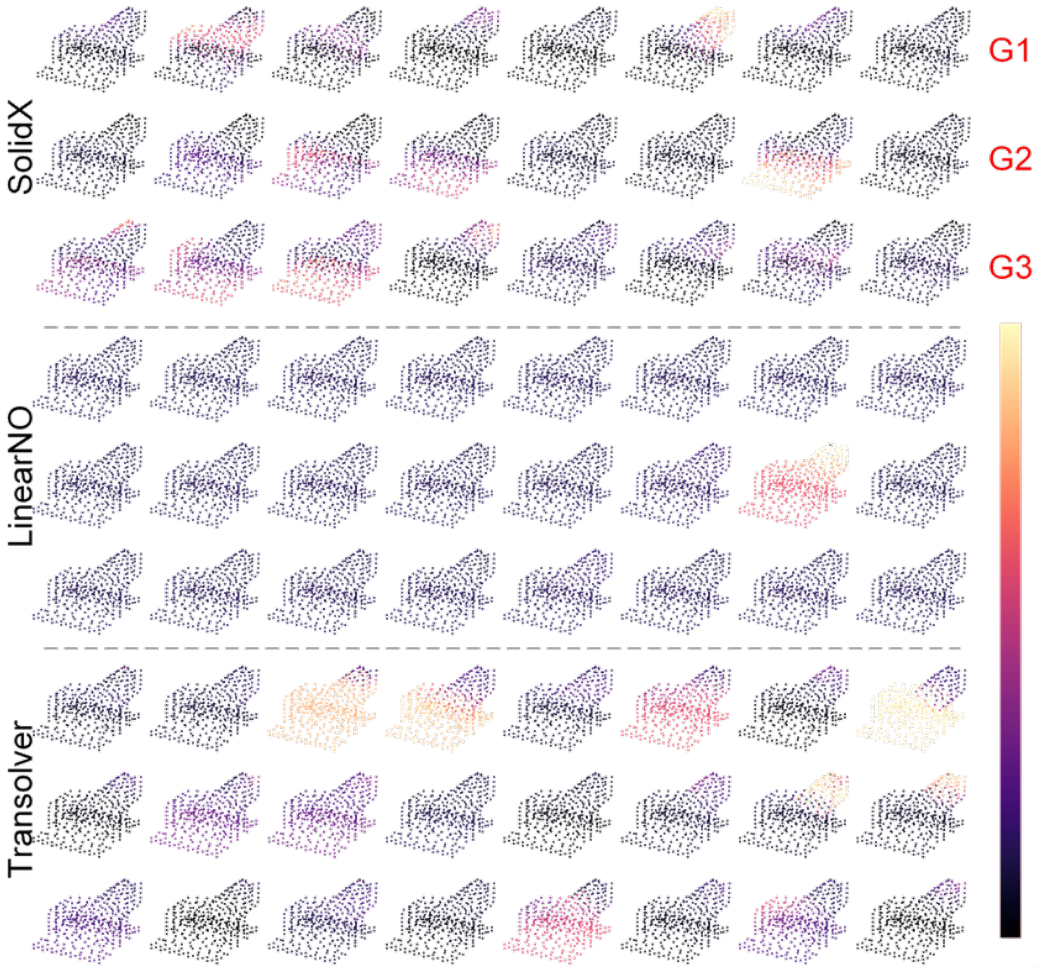


Figure 51: Visualization of the learned synthesis bases $\Phi(Q)$ for Tissue Manipulation at layer 5. For SolidX, each row corresponds to one physical group, while LinearNO and Transolver use ungrouped bases. Colors indicate the assignment weights of each basis over spatial points, reflecting how strongly the basis focuses on different regions. SolidX captures key physically relevant regions around local interactions, whereas the baselines' bases are markedly more homogeneous, with highly similar activation patterns that suggest stronger competition for shared representational capacity.