Mamba Learns in Context: Structure-Aware Domain Generalization for Multi-Task Point Cloud Understanding

Supplementary Material

This document provides additional technical details, ablation studies, and qualitative evaluations to complement the main paper. The contents are organized as follows:

- **A.** Further discussion of our Structure-Aware Serialization (SAS), including the spectral formulation of the Centroid Distance Spectrum (CDS), the motivation for geodesic graphs in the Geodesic Curvature Spectrum (GCS), and empirical evidence of structural drift under domain shifts.
- **B.** More ablation studies on key designs, including serialization strategies, Hierarchical Domain-Aware Modeling (HDM), Spectral Graph Alignment (SGA), and the complexity/runtime breakdown of the full framework. Unless noted, all ablations use MP3DObject as the target domain and report Chamfer Distance (CD) on reconstruction, denoising, and registration.
- **C.** Details of the MP3DObject dataset, including construction, qualitative comparisons with existing benchmarks, and class-wise visualizations in original versus aligned poses.
- **D.** Additional qualitative comparisons, with separate figures for all different target domains and tasks.
- **E.** Training and architectural hyperparameters corresponding to the released codes.

A. Structure-Aware Serialization: Further Discussion

The main paper introduces Structure-Aware Serialization (SAS) composed of the Centroid Distance Spectrum (CDS) and the Geodesic Curvature Spectrum (GCS), which together produce transformation-invariant and structure-consistent sequences for Mamba. Here we provide additional explanation of the CDS implementation and the choice of geodesic graphs for GCS.

A.1. Centroid Distance Spectrum (CDS)

GPU-friendly Spectral formulation of CDS. CDS is designed to impose a topology-aware ordering over patch tokens by expanding from the centroid along intrinsic surface connectivity. A literal implementation would first build a KNN graph over token centers and then perform a breadth-first search (BFS) from the centroid-nearest node. Although conceptually simple, this queue-based traversal is inherently sequential: each frontier must be fully expanded before the next level, leading to poor utilization of GPU parallelism during large-batch training.

To obtain a topology-coherent ordering without relying on sequential queue operations, we adopt a GPU-compatible spectral formulation. Based on the token graph $\mathcal{G}_{\mathrm{CDS}}$, we compute its normalized graph Laplacian $\mathbf{L}$. Let ϕ_1 denote the first non-trivial eigenvector of the generalized eigenproblem $\mathbf{L}\phi_k = \lambda_k \phi_k$. CDS then orders tokens by sorting their scalar embeddings:

$$\pi_{\mathrm{CDS}}(i) = \mathrm{argsort}\big(\phi_1(i)\big). \tag{16}$$

The Fiedler vector varies smoothly over well-connected regions, assigning nearby scalar values to intrinsically adjacent tokens. Consequently, the resulting serialization follows a topology-aware progression that stimulates the intended BFS behavior, while being fully realizable via batched linear-algebra operations on GPUs.

Neighborhood preservation analysis. To quantify how faithfully spectral CDS reflects the BFS expansion pattern, we introduce a neighborhood preservation rate (NPR) with naive BFS as reference ordering. For token i, let $\mathcal{N}_r^{\mathrm{BFS}}(i)$ and $\mathcal{N}_r^{\mathrm{CDS}}(i)$ denote its r-hop neighborhoods under naive BFS and spectral CDS, respectively, both defined on the same KNN graph. NPR is computed as

$$\mathrm{NPR} = \frac{1}{G}\sum_{i=1}^{G} \frac{\big|\mathcal{N}_r^{\mathrm{BFS}}(i) \cap \mathcal{N}_r^{\mathrm{CDS}}(i)\big|}{\big|\mathcal{N}_r^{\mathrm{BFS}}(i)\big|}, \tag{17}$$

where G is the number of tokens. NPR captures how well the spectral ordering preserves local neighborhoods induced by BFS and serves purely as an auxiliary diagnostic.

Comparison with naive BFS. We implement a naive BFS on CPU, computing the exact BFS layer sequence for reference. We then compare naive BFS and spectral CDS in terms of their neighborhood preservation rate (NPR), average Chamfer Distance (CD) on MP3DObject under reconstruction, denoising, and registration, and per-batch runtime, as shown in Table 3. NPR is computed with $r = 2$ hops by default, providing a locality-sensitive measure of how well the spectral ordering preserves BFS neighborhoods.

Table 3. Naive BFS vs. spectral CDS under MP3DObject as target.

Variant	NPR ↑	Avg. CD ↓	Runtime / batch ↓
Naive BFS (CPU)	1.00	4.27	2.22s
Spectral CDS (GPU, ours)	0.97	4.33	0.75s

Practical usage. Naive BFS is used only as an offline diagnostic to validate the spectral approximation on a small subset. All training and inference in the main paper employ the GPU-based spectral CDS implementation, which is stable, fully batched, and scalable.

A.2. Geodesic Curvature Spectrum (GCS)

Why Euclidean distances are insufficient. GCS is designed to capture local curvature and surface continuity within each patch. A naive approach would be to directly measure Euclidean distances between patch centers and use them as pairwise distances for subsequent diffusion. However, on curved surfaces, Euclidean distances in the ambient space can be misleading: two tokens on opposite sides of a folded surface may be close in straight-line Euclidean distance while being far apart along the surface itself. As illustrated in Figure 7, a straight segment connecting two tokens across a fold cuts through the volume and ignores curvature, whereas the intrinsic distance along the surface follows the bend.

For heat diffusion and curvature-sensitive descriptors, what matters is connectivity along the surface, not through the volume. Using Euclidean distances directly for diffusion would therefore

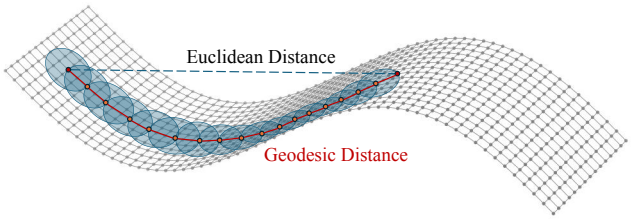

Figure 7. Illustration of why Euclidean distance fails on curved surfaces: straight-line distances shortcut across the token, whereas geodesic distances correctly follow the intrinsic surface geometry.

create artificial shortcuts that bypass folds and holes, breaking the intrinsic continuity of the object.

Geodesic graphs over patch centers. To reflect intrinsic surface geometry rather than raw Euclidean proximity, we build a geodesic graph whose nodes correspond to patch centers $\{u_i\}$. Each node is connected to its nearest neighboring patches, producing a locally coherent connectivity that follows the surface layout. Although these connections are established using distances in the ambient space, the resulting graph encourages shortest paths to propagate along the surface, which provides a closer approximation to intrinsic geodesic relations than direct center-to-center measurements.

Based on this graph, we apply heat diffusion at multiple time scales. Short diffusion times emphasize very local geometric variation, while longer times aggregate information over broader neighborhoods of the surface. For each patch, the diffusion responses are consolidated into an intrinsic scalar score that varies smoothly along regions of similar curvature and changes more distinctly across folds or high-curvature transitions. Sorting patches according to this score produces the GCS ordering, which reflects the intrinsic structure of the surface rather than depending on ambient-space distances between patch centers.

Robustness on real-scan domains. This graph-based formulation is particularly important on real-scan domains such as ScanObjectNN and MP3DObject, where occlusions, missing regions, and sensor noise are common. In such cases, Euclidean distances between sparse, noisy patch centers can be unstable. In contrast, GCS relies on local connectivity among patches and multi-scale diffusion, which remain more stable under partial observations and provide a consistent, domain-agnostic curvature cue for Mamba.

A.3. Structural Drift Evidence

Definition of structural drift. In the main paper, we show that coordinate-driven serializations are fragile under domain shifts. By *structural drift*, we mean that perturbations such as noise, occlusion, and pose variation can distort sequence-local neighborhoods induced by coordinate-based orders, causing them to deviate from the intrinsic topological or geometric neighborhoods of the underlying shape. This mismatch is particularly harmful to sequence models such as Mamba, whose recurrent updates rely on stable local ordering.

Neighborhood preservation rate under domain shifts. Similar to the NPR in Sec. A.1, we quantify this effect by comparing sequence-local windows with intrinsic neighborhoods. Given a serialization order π, let $\mathcal{W}_\pi^h(i)$ denote the local window centered at token i with radius h in the serialized sequence. We then define two intrinsic references: (1) *Topo-NPR*, where $\mathcal{N}_k^{\text{topo}}(i)$ is the k-NN neighborhood of token i on the token graph; (2) *Geo-NPR*, where $\mathcal{N}_k^{\text{geo}}(i)$ is the top-k neighborhood defined by encoder-feature similarity. For either definition, the NPR is computed as

$$\text{NPR}(h) = \frac{1}{G}\sum_{i=1}^{G} \frac{\left|\mathcal{N}_k(i) \cap \mathcal{W}_\pi^h(i)\right|}{|\mathcal{N}_k(i)|}. \tag{18}$$

A higher value indicates that the serialization better preserves intrinsic neighborhoods under domain perturbation.

Empirical comparison under pose perturbation. We evaluate the NPR under random rotations as a controlled form of domain shift and compare our CDS/GCS-based serialization against coordinate-driven baselines. As shown in Table 4, Z-order and Hilbert curves yield noticeably lower Topo-NPR and Geo-NPR, indicating stronger structural drift. In contrast, our structure-aware serialization preserves substantially more intrinsic neighborhoods, supporting our claim that SAS reduces drift by maintaining sequence-local neighborhoods that remain more faithful to the underlying geometry.

Table 4. Neighborhood preservation rate under random rotations (higher is better). Our serialization preserves intrinsic neighborhoods more faithfully than coordinate-driven baselines.

Method	Topo-NPR ↑	Geo-NPR ↑
Z-order	0.59	0.48
Hilbert curve	0.62	0.48
SAS (CDS+GCS, ours)	**0.69**	**0.67**

B. More Ablation Studies

We present more ablation studies on the main components of Structure-Aware Serialization (SAS), Hierarchical Domain-Aware Modeling (HDM), and Spectral Graph Alignment (SGA). Unless otherwise noted, all experiments follow the cross-domain protocol in which MP3DObject is treated as an unseen target domain. We report Chamfer Distance (CD $\times 10^{-3}$, lower is better) across reconstruction, denoising, and registration.

B.1. Serialization Strategies

Naive serialization and random traversal. The main paper compares coordinate-based serialization, CDS-only, and GCS-only variants. Here we further examine two additional baselines:

(1) *Naive FPS order*: tokens are ordered according to the Farthest Point Sampling (FPS) index, without any reordering;

(2) *Random traversal*: tokens are randomly permuted with a fixed random seed per instance, simulating a Transformer-style input where the model does not receive explicit structural ordering.

We compare these baselines with our full SAS (CDS+GCS) in Table 5. Naive FPS order already improves over completely unstructured inputs by preserving some spatial coverage, but it ignores intrinsic topology and curvature and therefore underperforms SAS. Random traversal further disrupts structural continuity, producing the worst CD among the three. These results confirm that SAS provides a meaningful and non-trivial serialization signal for Mamba.

Table 5. Ablations of serialization strategies on MP3DObject. Naive FPS order and random traversal are clearly inferior to our structure-aware serialization.

Variant	Reconstruction	Denoising	Registration
Naive FPS order	7.73	13.64	7.54
Random traversal	8.17	12.92	7.24
SAS (CDS+GCS, ours)	**3.55**	**6.61**	**2.84**

Fixed vs. data-adaptive kernel scales. In CDS and GCS, the Gaussian kernels that define edge weights are controlled by scale parameters σ and γ (as in eq. (4) and eq. (8)). In the main model, both are set in a data-adaptive way based on the median pairwise distances in the corresponding graphs, which automatically adjusts to varying object scale and point density.

To examine the effect of this design, we compare the adaptive setting with fixed kernel scales, which are commonly used in graph-based Gaussian weighting. Concretely, we consider:

(1) *Fixed kernel*: $\sigma, \gamma \in \{0.05, 0.1, 0.2\}$ in the normalized coordinate space;

(2) *Adaptive kernel (ours)*: σ and γ set to the median of local distances in CDS and GCS graphs, respectively.

As summarized in Table 6, the adaptive median-based scales consistently match or slightly surpass the best fixed choices, while eliminating the need for domain-specific tuning. Their data-driven nature allows the kernel scales to automatically adjust to variations in object size, sampling density, and noise patterns, thus mitigating cross-domain shifts and accommodating data captured from different sensors or scanning conditions.

Table 6. Fixed vs. data-adaptive kernel scales on MP3DObject. We report average Chamfer Distance (CD $\times 10^{-3}$) over reconstruction, denoising, and registration. Data-adaptive scales based on median distances provide robust performance without tuning.

Kernel setting	Avg. CD	Comment
Fixed $\sigma = \gamma = 0.05$	5.76	too local, sensitive to noise
Fixed $\sigma = \gamma = 0.10$	5.02	tuned baseline
Fixed $\sigma = \gamma = 0.20$	5.52	overly smooth, loses detail
Adaptive (median-based, ours)	**4.33**	robust across categories

B.2. Interleaving in Hierarchical Domain-Aware Modeling (HDM)

HDM first performs intra-domain structural modeling and then fuses prompt and query tokens via a global Mamba operating on an interleaved sequence. To assess the role of interleaving, we compare the proposed design to a simple concatenation variant:

$$Z_{\text{concat}} = [Z^p_{\pi(1)}, \ldots, Z^p_{\pi(4N)}, Z^q_{\pi(1)}, \ldots, Z^q_{\pi(4N)}], \tag{19}$$

where Z^p and Z^q denote prompt and query tokens serialized by SAS. As shown in Table 7, concatenation consistently underperforms the interleaved variant across three tasks on MP3DObject. A hard domain boundary in the concatenated sequence restricts state propagation from prompts to queries, while interleaving enforces fine-grained structural alignment between domains and allows Mamba to exploit prompt information at every step.

Table 7. Interleaving vs. concatenation in HDM on MP3DObject. Interleaving tokens from both domains enables more effective cross-domain information flow.

Variant	Reconstruction	Denoising	Registration
w/ concat	5.40	7.11	4.23
w/ interleave (ours)	**3.55**	**6.61**	**2.84**

B.3. Spectral Graph Alignment (SGA)

SGA vs. simple feature shifting. To highlight the role of SGA, we compare it with a simple feature shifting strategy that pushes target features towards source prototypes in the feature space:

$$X^t_* \leftarrow \beta\, X^t_* + (1-\beta)(P^s_* - X^t_*), \tag{20}$$

where P^s_* denotes source-domain prototypes and β is a fixed scalar (set to 0.5 by default). This baseline does not use spectral decomposition or frequency-aware mixing.

Table 8 shows that simple feature shifting recovers part of the domain gap but remains clearly inferior to SGA, particularly on registration. This indicates that aligning in the spectral domain of CDS/GCS graphs, rather than in raw feature space, is crucial for preserving structural consistency.

Table 8. SGA vs. simple feature shifting on MP3DObject. Spectral alignment yields consistently better domain generalization.

Variant	Reconstruction	Denoising	Registration
No SGA	11.95	17.38	9.43
Simple feature shift	7.62	12.56	7.96
SGA (ours)	**3.55**	**6.61**	**2.84**

Alignment strength. SGA employs adaptive cosine-similarity mixing weights that modulate spectral components according to domain affinity. To assess the influence of alignment magnitude, we replace the adaptive weights with a fixed global coefficient $\alpha \in \{0.0, 0.5, 1.0\}$ applied uniformly to the spectral mixing term. As shown in Table 9, disabling alignment ($\alpha = 0.0$) leads to weaker cross-domain consistency, while very strong alignment ($\alpha = 1.0$) risks over-correction. A moderate fixed strength ($\alpha = 0.5$) improves stability, but the adaptive cosine-similarity scheme remains the most effective overall, as it naturally adjusts to variations across domains, object geometry, and sensor conditions without requiring per-domain tuning.

Table 9. Effect of alignment strength in SGA on MP3DObject. We report average Chamfer Distance (CD $\times 10^{-3}$) over three tasks.

Setting	Avg. CD	Comment
$\alpha = 0.0$ (no SGA)	12.92	no alignment applied
$\alpha = 0.5$ (fixed)	6.77	improves over no alignment
$\alpha = 1.0$ (fixed)	10.20	over-alignment, less stable
Adaptive (cosine, ours)	**4.33**	similarity-based, best overall

B.4. Complexity and Runtime Breakdown

Complexity analysis. A natural question is whether SAS and SGA introduce substantial overhead beyond the Mamba backbone. In our implementation, both modules operate on a patch-token graph with G nodes after FPS+KNN grouping, rather than on the raw point cloud with P points. In all experiments, $G = 64 \ll P = 1024$, so the additional cost remains at the token level and is much smaller than operating directly on dense point sets.

Let G denote the token number, S the patch size, d the feature dimension, and L the serialized sequence length. CDS and SGA require spectral decomposition on a $G \times G$ token-graph Laplacian, resulting in complexity $O(G^3)$. GCS computes patch-wise spectra on G local graphs of size $S \times S$, leading to complexity $O(GS^3)$. Therefore, the overall overhead introduced by SAS and SGA is

$$T_{\text{SAS+SGA}} = O(G^3 + GS^3). \tag{21}$$

For sequential backbones, Mamba scales linearly as $O(Ld)$, while Transformer self-attention scales quadratically as $O(L^2 d)$.

Runtime breakdown. Table 10 reports the runtime breakdown of SADG, including SAS, Mamba forward, and SGA, together with FLOPs and parameter counts. Although SAS and SGA introduce additional graph computations, the total runtime of SADG remains lower than DG-PIC due to the linear-time Mamba backbone. This confirms that our structure-aware design improves efficiency while preserving strong domain generalization performance.

Table 10. Runtime breakdown and model complexity.

Method	SAS (s)	Fwd. (s)	SGA (s)	Total (s)	FLOPs (G)	Params (M)
DG-PIC [28]	–	0.94	–	0.94	21.07	27.57
SADG (ours)	0.33	0.25	0.17	0.75	14.89	18.87

C. MP3DObject Dataset: Construction and Characteristics

C.1. Construction Pipeline

MP3DObject is constructed from Matterport3D [5] by extracting object-level point clouds from indoor scenes. For each annotated object instance, we crop the corresponding points, center them, and normalize into a unit sphere, without enforcing any canonical orientation. Extremely incomplete objects whose visible surface area falls below a threshold, as well as degenerate cases with too few points, are discarded. The final dataset contains 4,015 training and 1,003 testing samples over seven shared categories and exhibits substantial variation in layout, occlusion, and pose.

C.2. Qualitative Comparison with Other Datasets

We qualitatively characterize each existing dataset along four conceptual axes: *curvature complexity* (how frequently the surface bends or folds), *extent of missing regions* (size and frequency of holes), *noise and artifacts* (measurement noise, misalignment, and clutter), and *pose variability* (degree of non-canonical orientation).

As shown in Table 11 and Figure 8, MP3DObject tends to have more complex furniture, larger unobserved regions, more cluttered surroundings, and more diverse poses compared to conventional datasets. This makes it a particularly demanding real-scan domain for structure-aware modeling and domain generalization.

Table 11. Qualitative characterization of datasets along four difficulty dimensions. "Low / mid / high" indicate relative levels to highlight trends. MP3DObject sits at the most challenging end across all dimensions.

Dataset	Curvature	Missing	Noise	Orientation
ModelNet	low	low	low	low
ShapeNet	low–mid	low	low	low
ScanNet	mid	mid–high	mid	mid
ScanObjectNN	mid–high	high	high	high
MP3DObject	**high**	**high**	**high**	**high**

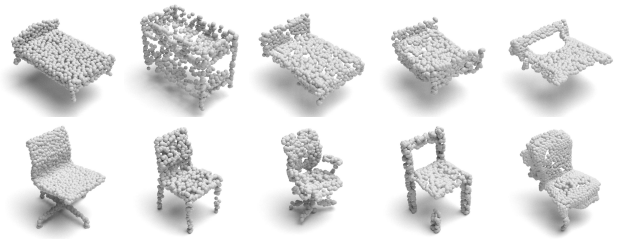

Figure 8. **Visual comparison of datasets.** MP3DObject instances present complex indoor objects in cluttered scenes with substantial occlusions and highly varied poses. *For visualization clarity, MP3DObject samples are shown in a manually aligned canonical view; all training and evaluation use the original unaligned scans.*

C.3. Class-wise Visualization

To further illustrate MP3DObject, we provide class-wise visualizations in both the original unaligned pose and a manually aligned pose used only for visualization. For each category (e.g., bed, bookshelf, cabinet, chair, monitor, sofa, table), we randomly select several instances and render them as pairs in Figure 9.

D. Qualitative Evaluation

We follow the leave-one-domain-out protocol, selecting one dataset as the unseen target and training on the remaining four. Representative qualitative results for more domains are shown in Figure 10, covering reconstruction, denoising, and registration. The visual patterns follow the quantitative trends in Table 1. Classical method [63] and augmentation-based DG approach methods [94] exhibit incomplete recovery of thin structures and occasional topological breaks. DG-PIC [28] produces more stable outputs but often retains coarse geometry, especially when inputs are severely partial. Coordinate-based Mamba ICL improves long-range consistency but frequently yields fragmented local patches due to its sensitivity to traversal order.

As the target domain becomes more challenging, *i.e.,* MP3DObject with complex furniture and large unobserved regions, the qualitative gap widens. Baselines commonly hallucinate missing parts, collapse curved surfaces, or generate disconnected fragments. Across all domains and tasks, our method provides the most coherent reconstructions, smoothest denoising, and most stable registrations, preserving both global structure and fine-grained

details. These qualitative observations highlight the benefits of structure-aware serialization, hierarchical domain-aware modeling, and spectral graph alignment under severe domain shift.

E. Training and Model Hyperparameters

For reproducibility, we summarize the main training and architectural hyperparameters used in our experiments. These settings correspond to the released codes.

Training configuration. We train all models using AdamW with cosine learning rate scheduling:

- Optimizer: AdamW, learning rate 1×10^{-4} (for global batch size 96, following linear scaling), weight decay 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$.
- Scheduler: cosine decay (CosLR) for 300 epochs, with a warmup stage over the first 20 epochs and a minimum learning rate of 10^{-6}.
- Point sampling: each shape is downsampled to 1,024 points for both training and testing.
- Batch setting: total batch size 96, gradient accumulation step 1.
- Loss: Chamfer Distance L2 (CDL2) for all tasks.

Patch and token configuration.

- Number of patches per shape: $G = 64$.
- Patch size: $S = 32$ points per patch.
- Patch encoder dimension: 256.
- Serialization: default type `both_parallel`, applying CDS and GCS in parallel and concatenating their outputs into a sequence of length $4G$.

Mamba backbone.

- Token embedding dimension (`trans_dim`): 256.
- Number of encoder Mamba layers: 4.
- Number of decoder Mamba layers: 2.
- Bidirectional Mamba: enabled (`bidir=true`) for all experiments.
- Drop path rate: 0.1.
- Masking ratio on target segments: 0.7.
- Hierarchical Domain-Aware Modeling (HDM): enabled by default.

These settings are kept fixed across the experiments reported in the main paper and this supplementary document, except where explicitly varied in ablation studies.

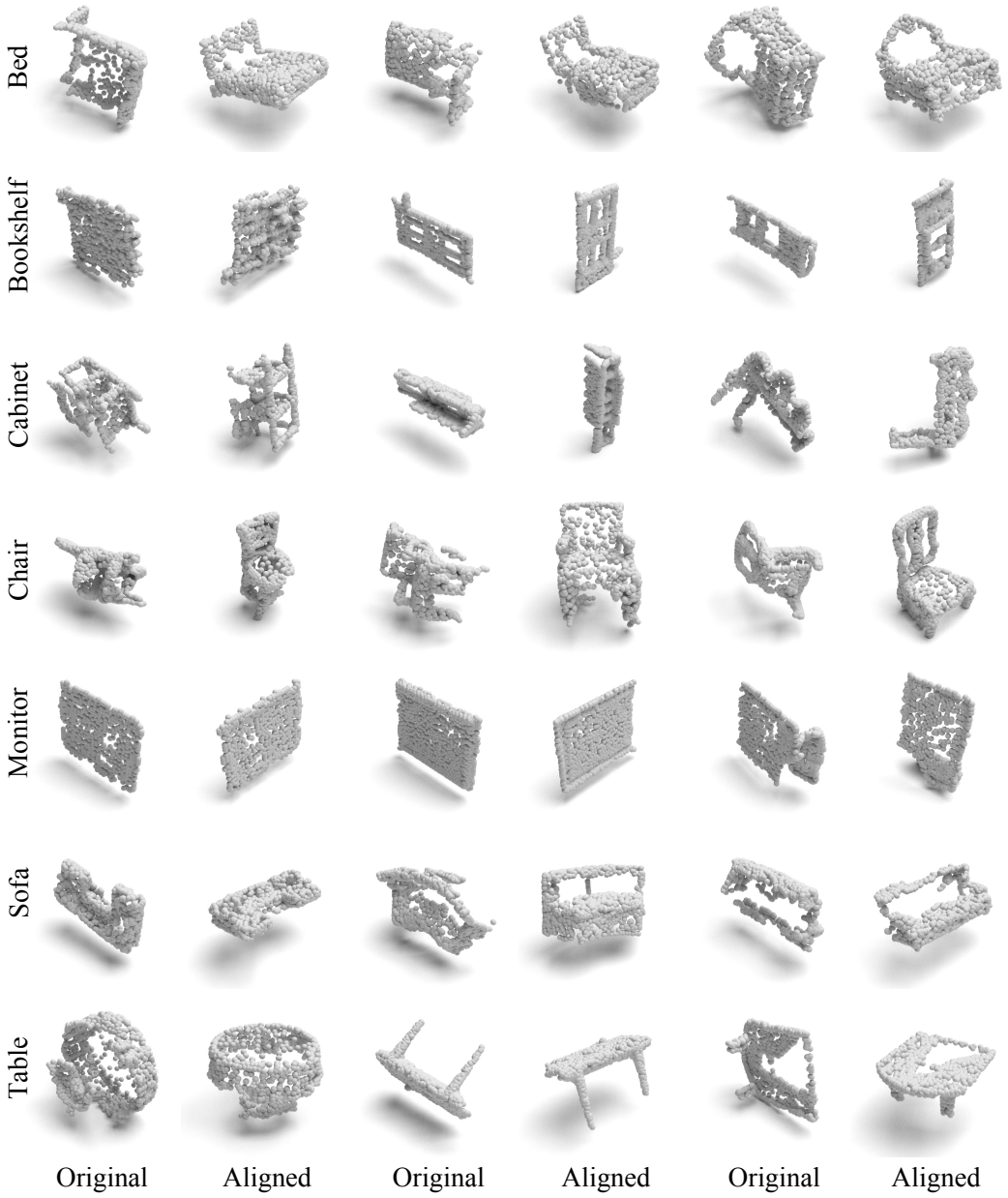

Figure 9. **MP3DObject per-class visualization in original and aligned poses.** For each class, we show several instances in their original unaligned pose (left of each pair) and in a manually aligned pose (right of each pair) to facilitate visual inspection. The original pose distribution is highly diverse, reflecting realistic scanning conditions. *Alignment is applied **only** for visualization; all training and evaluation in our experiments use the original unaligned MP3DObject scans.*

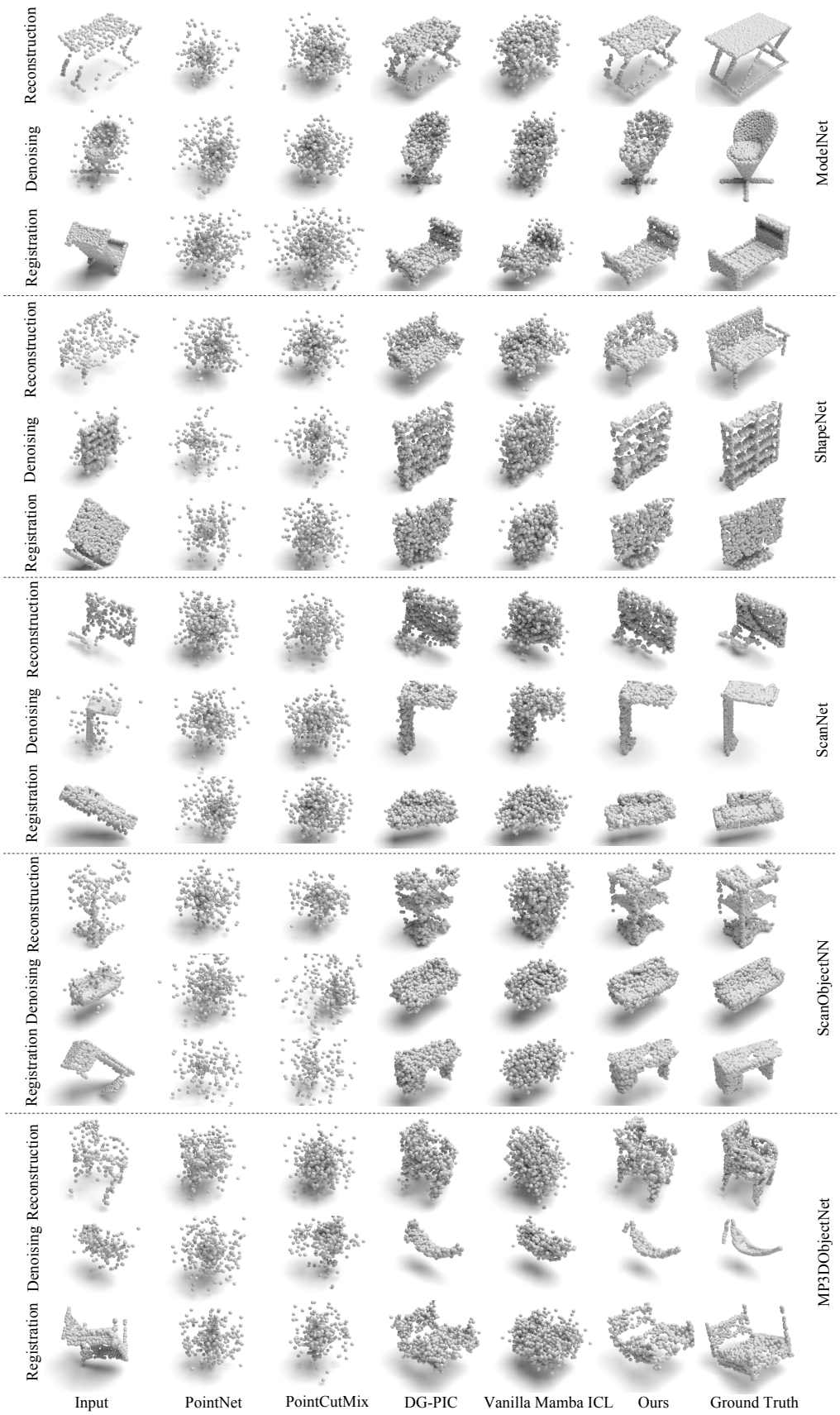

Figure 10. Qualitative comparisons on different target domains.