SHED Light on Segmentation for Dense Prediction

Anonymous authors
Paper under double-blind review

Abstract

Dense prediction infers per-pixel values from a single image and underlies 3D perception and robotics. Real-world scenes are highly structured, yet most models predict each pixel independently, producing blurred boundaries and geometrically inconsistent surfaces. We propose SHED, an encoder-decoder that builds an explicit hierarchy of image segments and then decodes it. The encoder groups pixels into progressively coarser segment tokens, and the decoder inverts this grouping, unpooling coarse tokens back to fine ones through the learned assignments rather than generic feature upsampling. The hierarchy is trained end-to-end from the dense prediction objective alone, without any segmentation labels. On monocular depth, SHED sharpens occlusion boundaries and intra-object coherence, transfers strongly from synthetic to real domains, and uses less computation than the comparable DPT baseline. The learned hierarchy also organizes features by scene layout, improving semantic alignment and 3D reconstruction. Applying the same mechanism to optical flow improves a strong baseline and surpasses a recent state-of-the-art method, indicating that decoding a learned segment hierarchy is a general principle for dense prediction.

1 Introduction

Dense prediction tasks, such as monocular depth estimation and semantic segmentation, serve as the foundation for spatial reasoning in robotics Liu et al. (2024). At their core, these tasks require an understanding of how 2D image projections relate to the underlying 3D structure of the world. However, modern state-of-the-art models based on Vision Transformers (ViTs) (Dosovitskiy, 2020) and pixel-wise regression often treat these tasks as a collection of independent point estimations rather than a coherent structural reconstruction. This disconnect frequently results in structural leakage, where predicted boundaries do not align with physical objects, leading to blurred edges and inconsistent surface geometry (Figure 1, row 1).

In contrast, the human visual system does not process images as an array of independent points. Instead, it employs a sophisticated bidirectional hierarchy (Hochstein & Ahissar, 2002). Our perception integrates depth and segmentation through a feedback loop where a global scene layout is first inferred by grouping fine-grained textures into parts and wholes. This high-level structural scaffold then guides the refinement of local details. This part-whole reasoning ensures that local pixel-level predictions are constrained by the global geometric context. This organization results in depth maps characterized by sharp, high-fidelity boundaries and smooth intra-object variations (Figure 1, row 2).

To realize this idea, we propose **SHED**, **S**egment **Hi**erarchy for **D**ense Prediction, a novel architecture that re-conceptualizes dense prediction as a process of hierarchical structural reasoning. Unlike traditional models that decode latent features directly back to a rigid pixel grid, SHED learns an implicit hierarchy of segments that adaptively decompose the scene. Note that these segments are not provided via external supervision. They are learned end-to-end, guided solely by the dense prediction objective. While hierarchical grouping on the encoder side builds on prior work, our central contribution is the inversion of this hierarchy for dense prediction: coarse structural abstractions must be turned back into spatially aligned, pixel-wise outputs without losing boundaries or local geometry, which generic feature upsampling does not address.

The technical core of SHED lies in its symmetric bidirectional flow. While the encoder iteratively groups pixels into increasingly abstract segment tokens, our contribution focuses on the structural inversion of this

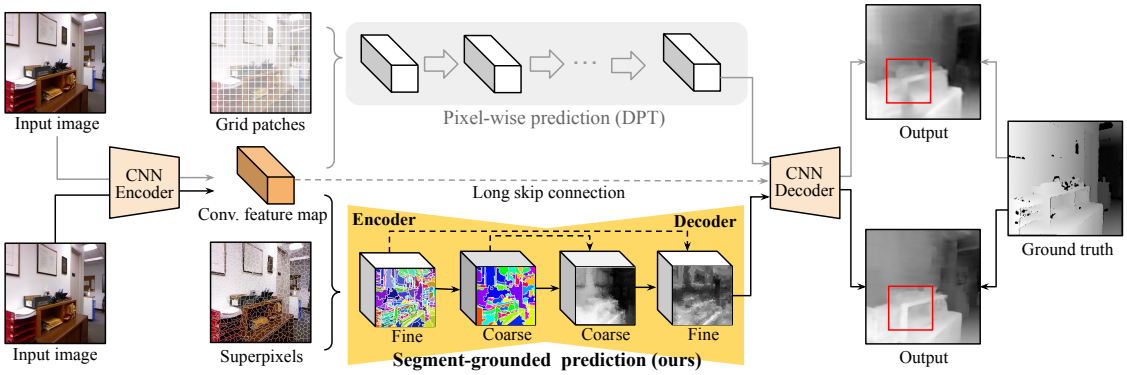

Figure 1: **Segment hierarchy for dense prediction (SHED).** Conventional methods such as DPT (Ranftl et al., 2021) perform pixel-wise prediction without considering structure, often resulting in blurry object shapes. SHED addresses this by leveraging a hierarchy of segment tokens to guide prediction. Unlike DPT, which uses fixed grid tokens across all layers, we adapt its ViT (Dosovitskiy, 2020) blocks into two stages: the encoder pools superpixel tokens into coarser segment tokens, and the decoder progressively refines predictions from coarse to fine segments, producing depth maps with structural coherence.

process in the decoder. To produce a structured dense map, SHED's decoder unpools features from coarser segments to finer ones using probabilistic assignments computed during the encoding phase. This reverse hierarchy, which has no counterpart in encoder-only segmentation frameworks, enables coarse structural context to directly constrain fine-grained predictions. By distributing segment-level features across associated regions, the model naturally enforces sharp boundaries between distinct objects and maintains geometric smoothness within a single surface. This mechanism allows high-level geometric priors to strictly inform the pixel-level output, preserving the global layout while capturing fine-grained structural detail.

Integrating hierarchical segmentation into the dense prediction pipeline grants SHED three key advantages:

- **Structured depth.** SHED produces depth maps with high-fidelity boundary alignment and intra-segment coherence. This structural bias improves robustness in cross-domain transfer from synthetic to real-world scenes, where pixel-wise models often fail to stay consistent, while also reducing computation.
- **Structure-aware representations.** Because the segment hierarchy emerges purely from depth supervision, it organizes features by scene geometry: SHED improves alignment with semantic regions and layout-aware retrieval over appearance-based grouping, without using any segmentation labels.
- **Generality.** The same reverse-hierarchy mechanism extends across dense prediction tasks, improving optical flow over strong baselines, and lifts predictions into coherent 3D reconstructions, with part-level structure emerging as a qualitative byproduct.

2 Related Work

Dense prediction is a core problem in computer vision, aiming to assign pixel-level outputs across an image (Forsyth & Ponce, 2002). It includes tasks such as semantic segmentation (Long et al., 2015), depth estimation (Eigen et al., 2014), optical flow (Teed & Deng, 2020), and image editing (Isola et al., 2017). Modern approaches typically adopt encoder-decoder architectures, such as U-Net (Ronneberger et al., 2015) and DPT (Ranftl et al., 2021), trained using task-specific supervision. These models perform well on benchmarks focused on per-pixel accuracy, as demonstrated by large-scale systems like Segment Anything (Ravi et al., 2025) and Depth Anything (Yang et al., 2024). However, recent studies show that even top-performing models often lack structural consistency (El Banani et al., 2024; Man et al., 2024). We argue that dense prediction should move beyond local estimation toward structured reasoning guided by region-level abstraction.

Monocular depth estimation is a representative dense prediction task, that infers per-pixel depth from a single image. It is widely used in 3D reconstruction (Song et al., 2017), autonomous driving (Geiger et al.,

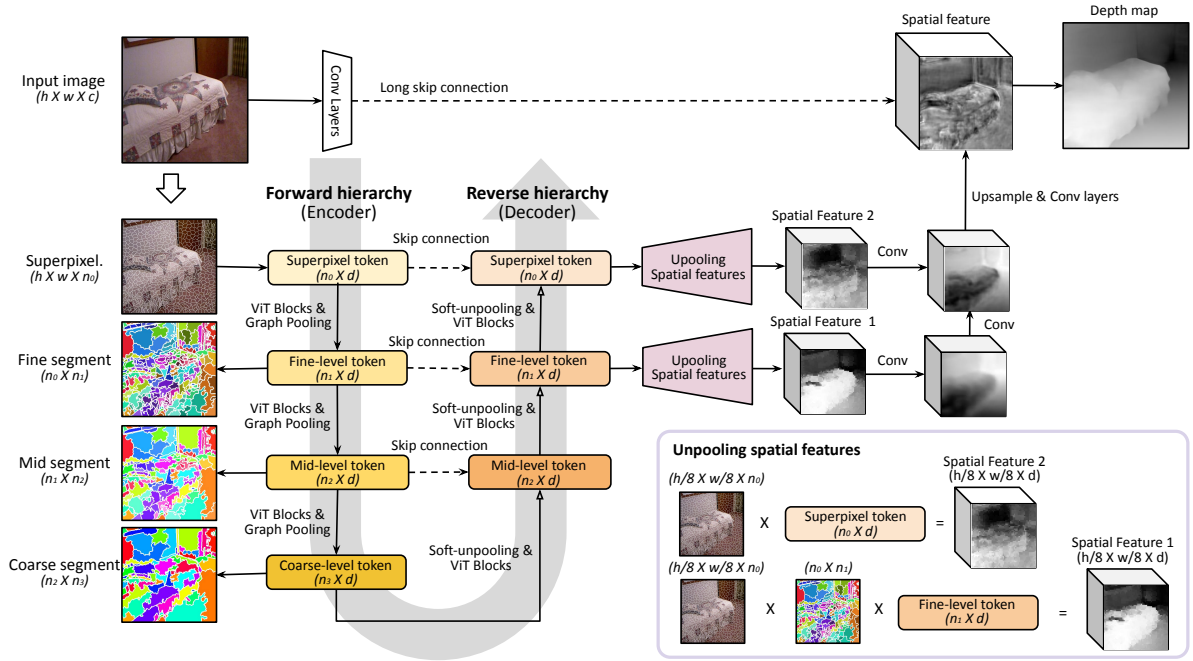

Figure 2: **SHED introduces a bidirectional segment hierarchy for dense prediction.** SHED decomposes dense prediction into two complementary processes: structural abstraction and structural inversion. **1)** The encoder constructs a forward hierarchy by grouping superpixel tokens into increasingly abstract segments. **2)** The decoder explicitly inverts this hierarchy by unpooling segment representations from coarse to fine, enabling global structural context to directly constrain pixel-level predictions. Segment tokens are projected into region-aligned 2D feature maps and fused across hierarchy levels, together with early convolutional features, to recover fine details and produce the final dense map.

2012), and robotic perception (Tateno et al., 2017). Structural cues in depth estimation have been extensively explored to enhance geometric coherence. Existing approaches can be broadly categorized into four types: **1)** Representation approaches modify how depth is encoded, such as by discretizing depth values (Fu et al., 2018; Bhat et al., 2021; Li et al., 2024) or modeling spatial dependencies (Liu et al., 2015; Cheng et al., 2018; Yuan et al., 2022). **2)** Regularization imposes geometric constraints through loss functions that promote smooth surfaces (Godard et al., 2017; Zhan et al., 2018; Bian et al., 2019), consistent normals (Yang et al., 2018), or planar regions (Yin et al., 2019; Watson et al., 2019). **3)** Multi-task learning jointly estimates depth with auxiliary signals, such as scene geometry (Eigen & Fergus, 2015; Yin & Shi, 2018) or semantics (Mousavian et al., 2016; Kendall et al., 2018; Chen et al., 2019; Guizilini et al., 2020; Zhu et al., 2020). **4)** Post-processing refines predictions using off-the-shelf techniques (Krähenbühl & Koltun, 2011; Chen et al., 2016).

Several multi-task approaches have explored segmentation as an auxiliary signal to improve depth estimation. Early works used segmentation as an additional supervision signal (Mousavian et al., 2016; Kendall et al., 2018), while more recent ones leveraged segment regions or boundaries to guide depth discontinuities (Chen et al., 2019; Guizilini et al., 2020; Zhu et al., 2020). In contrast, SHED integrates segmentation directly into the model's internal representation, enabling depth and segmentation to be jointly constructed and refined within a unified hierarchical framework. Moreover, the segment hierarchy in SHED is learned in an unsupervised manner, eliminating the need for additional human annotations.

Perceptual grouping is a key mechanism in human vision that organizes low-level elements into coherent global structures (Wertheimer, 1938; Marr, 2010). This principle has inspired a broad range of computer vision research, including perception (Locatello et al., 2020; Mo et al., 2021; Kang et al., 2022; Deng et al., 2023; Ranasinghe et al., 2023), segmentation (Arbeláez et al., 2012; Hwang et al., 2019; Ke et al., 2022; Xu et al., 2022), and generation (Hong et al., 2018; Mo et al., 2018; He et al., 2022). In particular, CAST (Ke et al., 2024) recently applied it to ViTs for concurrent segmentation and recognition. However,

most of these methods, including CAST, consider only a *forward hierarchy*, constructing representations and segmentations in a bottom-up manner. Without a reverse hierarchy, such representations cannot enforce spatial consistency in fine-grained dense prediction. In contrast, we adopt the complementary concept of a *reverse hierarchy* (Hochstein & Ahissar, 2002), where global structures guide and refine local parts through top-down feedback. While some prior works (Anderson et al., 2018; Shi et al., 2023; Eftekhar et al., 2023) have explored reverse hierarchies for recognition, they do not address dense prediction. Other studies (Eslami et al., 2016; Sajjadi et al., 2022; Seitzer et al., 2022) apply similar ideas to encoder-decoder architectures, but focus on object-centric representations, lacking the ability to model segment hierarchies. To the best of our knowledge, this is the first work to leverage bidirectional segment hierarchies to enhance dense prediction within a modern ViT.

3 SHED: Segment Hierarchy for Dense Prediction

We propose SHED, a dense prediction architecture that explicitly models a bidirectional segment hierarchy (see Figure 2). The encoder groups pixels into segment tokens and progressively pools them into coarser representations, while the decoder reverses this process by unpooling segment features from coarse to fine. By propagating segment-level structure back to pixel-level predictions, SHED enables global scene layout to directly constrain dense outputs, overcoming a limitation of pixel-wise models.

3.1 Encoder: Grouping segments via forward hierarchy

The encoder constructs a forward segment hierarchy that abstracts fine-grained regions into increasingly coarse structural units. This hierarchy serves as the structural backbone of SHED, capturing part–whole relationships that are difficult to represent using grid-based tokens. While similar hierarchical grouping strategies have been explored for image-level recognition, dense prediction requires such abstractions to be explicitly reversible, which motivates our overall bidirectional design.

To construct the forward hierarchy, we adopt a segment-based tokenization strategy that replaces square patch tokens with superpixel tokens and progressively clusters them based on feature similarity. This grouping mechanism follows prior work on hierarchical segmentation Ke et al. (2024), but in SHED it serves a fundamentally different role: providing intermediate structural representations that will later be inverted for dense prediction.

Unlike encoder-only hierarchical segmentation models, the forward hierarchy in SHED is not an end in itself. Instead, it defines an intermediate structural representation whose primary purpose is to be inverted by the decoder to enable structured dense prediction.

Tokenization. Given an image $X \in \mathbb{R}^{h \times w \times c}$, the encoder produces hierarchical segmentations $S_0, S_1, \ldots$ and corresponding embeddings $Z_0, Z_1, \ldots$, ordered from fine to coarse. This process begins by dividing the image into n_0 superpixels, which yields a one-hot assignment matrix $S_0 \in \mathbb{R}^{(h \cdot w) \times n_0}$ that maps each pixel to a superpixel. We extract a convolutional feature map $F_{\text{conv}} \in \mathbb{R}^{(h_0 \cdot w_0) \times d}$ with spatial stride 8 ($h_0 = h/8$, $w_0 = w/8$), add fixed sinusoidal positional embeddings, and average-pool features within each superpixel to obtain initial embeddings $Z_0 \in \mathbb{R}^{n_0 \times d}$. For global context modeling, we append a class token to form $\bar{Z}_0 \in \mathbb{R}^{(n_0+1) \times d}$, and feed it into the first ViT block.

Hierarchical clustering. Coarser segments are obtained by alternating ViT blocks and graph pooling (Ke et al., 2024). At each level l, given Z_{l-1} and S_{l-1} from the previous layer, we append a class token to form $\bar{Z}_{l-1}$, apply ViT blocks, and obtain updated features, excluding the class token.

To form coarser tokens $Z_l \in \mathbb{R}^{n_l \times d}$, we compute a soft assignment matrix $P_l \in \mathbb{R}^{n_{l-1} \times n_l}$ based on cosine similarity between fine- and coarse-level tokens:

$$P_l(i \to j) \propto \text{sim}(Z_{l-1}[i], Z_l[j]), \quad \text{for } i \in [n_{l-1}],\, j \in [n_l],$$

where $[n] := \{0, \ldots, n-1\}$. The coarse tokens Z_l are initialized via farthest point sampling (Qi et al., 2017) from Z_{l-1}, and refined by aggregating fine-level features by P_l, followed by an MLP and a residual connection:

$$Z_l \leftarrow Z_l + \text{MLP}(P_l^\top Z_{l-1} \oslash P_l^\top \mathbf{1}),$$

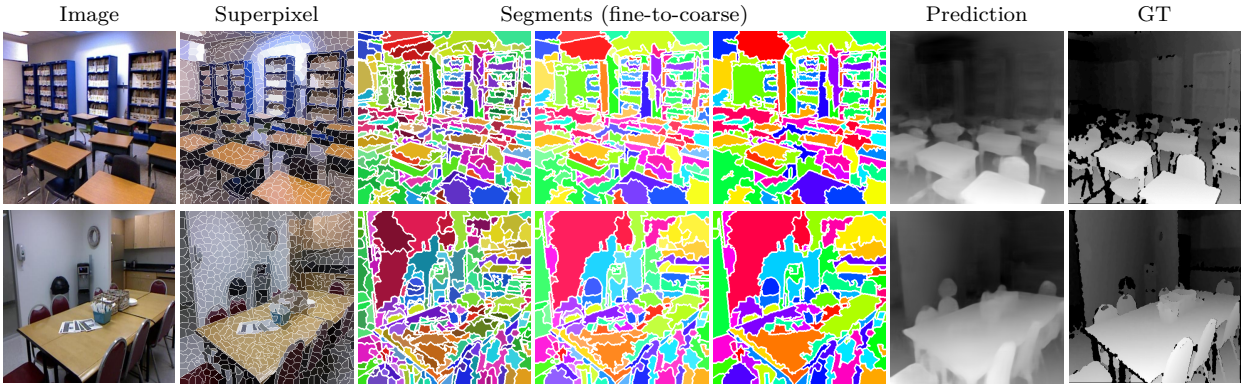

Figure 3: **SHED produces consistent structures in predicted depth map with spatio-layout.** We visualize the fine-to-coarse segments and corresponding depth maps from SHED, along with ground truth (GT) depth. Examples are from the NYUv2 test set. SHED captures fine structures through its segments, such as desks in a classroom, which allow the depth map to clearly separate them from the background (row 1). It also decomposes large objects, such as a table, into multiple parts, leading to smooth depth variations toward the back (row 2).

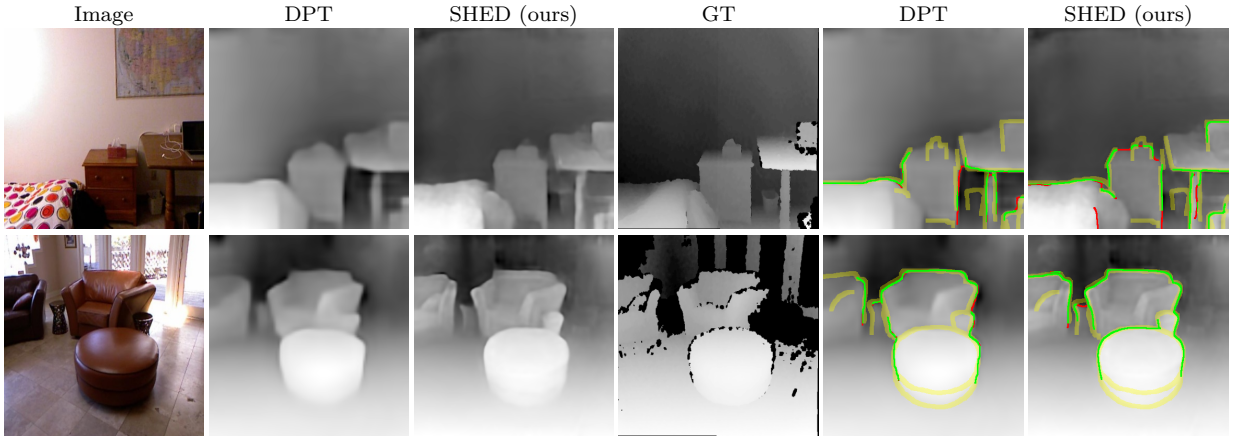

Figure 4: **SHED generates sharper object contours, clearer occlusion boundaries, and more coherent values within segments.** We compare depth maps (cols 2-4) and occlusion boundaries (cols 5, 6) from DPT, SHED on the NYUv2-OC++ dataset. Boundaries are extracted using a Canny edge detector and evaluated against GT, with GT edges shown in yellow, true positive in green and false positive in red. SHED more accurately captures object edges and produces smoother depth within segments. Its predicted boundaries also align more closely with the ground truth.

where $\oslash$ denotes element-wise division for normalization.

To propagate segmentation labels through the hierarchy, we compute coarser segmentations by composing the assignment matrices:

$$S_l = S_{l-1}\,\bar{P}_l, \quad l = 1, 2, \ldots, l_{\max},$$

where $\bar{P}_l$ is a hard assignment matrix obtained by taking the argmax over each row of P_l.

3.2 Decoder: Predicting outputs via reverse hierarchy

The decoder reconstructs spatial feature maps by reversing the encoder's segment hierarchy, progressively unpooling segment tokens $Z_{l_{\max}}, \ldots, Z_0$. This involves two steps: **1)** computing decoder features Z'_l by unpooling from Z'_{l+1} and fusing them with encoder features Z_l via skip connections; and **2)** projecting Z'_l to the image space to obtain a spatial feature map F_l of size (h_l, w_l).

Unpooling segment tokens. Formally, the reverse hierarchy distributes coarse segment features to finer segments through learned soft assignments. At each level $l = l_{\max} - 1, \ldots, 0$, we compute

$$Z'_l \leftarrow P_{l+1}^{\top} Z'_{l+1},$$

which distributes coarse features to finer segments. We then add unpooled features with corresponding encoder output:

$$Z'_l \leftarrow \mathrm{MLP}(Z'_l + Z_l),$$

followed by ViT blocks with class tokens.

Unpooling spatial features. We convert the segment tokens Z'_l into spatial feature maps by composing the soft assignment matrices:

$$P_{0\to l} = P_1 \cdots P_l \quad \in \mathbb{R}^{n_0 \times n_l},$$

and applying them to the initial superpixel-to-pixel map S_0 to obtain soft segmentations $S_{0\to l} = S_0 P_{0\to l}$. The spatial feature map is then reconstructed as

$$F_l = S_{0\to l}\, Z'_l, \quad F_l \in \mathbb{R}^{(h_l \cdot w_l)\times d}.$$

The set of spatial maps $\{F_l\}_{l=1}^{l_{\max}}$ is fused using convolutional layers, combined with F_{conv}, and refined through final convolution and upsampling to produce the final dense prediction.

A common design in dense prediction is to build a spatial hierarchy by progressively reducing feature resolution. DPT (Ranftl et al., 2021) is a representative example of this approach. DPT reduces the spatial resolution of feature maps F_l at each level by a factor of 2^l, with $h_l = h_0/2^l$, $w_l = w_0/2^l$, producing coarse maps in early ViT layers that are progressively refined. However, it relies on local aggregation, which lacks structural consistency. In contrast, SHED replaces the spatial hierarchy with a segment hierarchy that explicitly models region abstraction and inversion. By operating on segment tokens rather than grid locations, SHED enforces structural consistency across hierarchy levels and within regions. Therefore, we omit spatial reduction in SHED and simply set $h_l = h_0$, $w_l = w_0$.

4 Experiments

We demonstrate the benefits of SHED by integrating segmentation into the loop for dense prediction: **1)** Segment-consistent depth estimation that preserves occlusion boundaries and intra-segment coherence, leading to improved accuracy and efficiency; **2)** Structure-aware representation learning through dense supervision; **3)** 3D scene reconstruction from predicted depth maps, yielding globally coherent and part-aware structures. We focus our analysis on depth estimation and its downstream structure, and finally show that the same mechanism generalizes to another dense prediction task, optical flow.

4.1 Setup

For fair comparison with prior dense prediction methods, we follow the standard training protocol commonly used in DPT-style models (Ranftl et al., 2021). Specifically, we adopt the DPT-Hybrid configuration, which combines ResNet-50 (He et al., 2016) and ViT-Small (Dosovitskiy, 2020), and refer to it simply as DPT throughout the paper. For in-domain evaluation, we primarily train and evaluate on NYUv2 (Nathan Silberman & Fergus, 2012), a standard benchmark for indoor depth estimation. For cross-domain evaluation, we train SHED on the synthetic HyperSim dataset (Roberts et al., 2021) and assess its *synthetic-to-real transfer* to the real-world NYUv2 dataset. Throughout the paper, DPT is our primary *controlled* baseline, since it shares the same backbone, pre-training, and training data as SHED and therefore isolates the effect of the proposed structural design. We additionally report stronger prior-based models, including the DPT-style Depth Anything v2 (Yang et al., 2024) and Marigold (Ke et al., 2025), both fine-tuned on HyperSim, as *contextual references*; these derive much of their performance from substantially larger-scale pre-training pipelines and are therefore not architecturally matched comparisons. Depth Anything v2 uses the DPT decoder with DINOv2 (Oquab et al., 2023) encoder.

Table 1: **SHED improves boundary accuracy and object-level consistency.** We evaluate the structural quality of depth maps using two metrics: **1)** Occlusion boundary error, evaluated on the NYUv2-OC++ dataset. Occlusion boundaries are extracted using a Canny edge detector, and the Chamfer distance is computed in both directions: from prediction to ground truth and vice versa. **2)** Intra-segment coherence measures how well the predicted depth values within each object align with the ground-truth using object-level annotations.

Method	Boundary Error ↓		Object-wise Depth Accuracy ↑	Object-wise Depth Error ↓		
	ε_a	ε_c	$\delta > 1.25$	AbsRel	RMSE	$\log_{10}$
DPT	6.395	1.438	0.802	0.144	0.500	0.061
SHED (ours)	**5.713**	**0.608**	**0.814**	**0.142**	**0.496**	**0.060**

Table 2: **SHED achieves competitive in-domain performance and strong synthetic-to-real transfer (synthetic → real).** We report accuracy and error metrics on the NYUv2 test set. DPT is the controlled baseline (matched backbone, pre-training, and data); Depth Anything v2 Yang et al. (2024) and Marigold Ke et al. (2025) are stronger contextual references that rely on far larger pre-training. SHED yields in-domain results comparable to DPT and, under synthetic-to-real transfer, improves clearly over DPT while remaining competitive with these much larger models.

Method	Pre-training	Training	Depth Accuracy			Depth Error		
			$\delta > 1.25$ ↑	$\delta > 1.25^2$ ↑	$\delta > 1.25^3$ ↑	AbsRel ↓	RMSE ↓	$\log_{10}$ ↓
In-domain depth estimation (trained on NYUv2)								
DPT	IN-1K	NYUv2	0.839	0.971	**0.992**	0.132	0.457	0.055
SHED (ours)	IN-1K	NYUv2	**0.846**	**0.972**	**0.992**	**0.130**	**0.451**	**0.054**
Joint segmentation–depth learning								
Mousavian et al. Mousavian et al. (2016)	IN-1K	NYUv2	0.568	0.856	0.956	0.200	0.816	0.061
Simsar et al. Simsar et al. (2022)	IN-1K	NYUv2	0.847	0.971	**0.993**	**0.116**	0.448	-
SHED (ours)	IN-1K	NYUv2	**0.855**	**0.974**	**0.993**	0.123	**0.433**	**0.052**
Cross-domain synthetic-to-real transfer (HyperSim → NYUv2)								
Marigold Ke et al. (2025)	Laion-5b	HyperSim	0.375	0.659	0.833	**0.542**	1.243	0.171
Depth Anything v2 Yang et al. (2024)	LVM-142M	HyperSim	0.592	**0.902**	**0.960**	0.749	0.808	0.110
SHED (ours)	IN-1K	HyperSim	**0.632**	0.892	**0.960**	0.583	**0.740**	**0.102**

Table 3: **Ablation study of hierarchical encoding and decoding in SHED.** We compare the full model with a variant that removes the reverse hierarchy in the decoder on the NYUv2 test set. The results highlight the role of coarse-to-fine unpooling in depth prediction.

Method	Depth Accuracy			Depth Error		
	$\delta > 1.25$ ↑	$\delta > 1.25^2$ ↑	$\delta > 1.25^3$ ↑	AbsRel ↓	RMSE ↓	$\log_{10}$ ↓
DPT	0.839	0.971	**0.992**	0.132	0.457	0.055
SHED with forward hierarchy only	0.755	0.951	0.989	0.163	0.552	0.069
SHED (ours)	**0.846**	**0.972**	**0.992**	**0.130**	**0.451**	**0.054**

Tokenization. Input images of size 640×480 are randomly cropped to 384×384 during preprocessing. We generate 576 superpixels using the SEEDS algorithm (Van den Bergh et al., 2012), matching the 24×24 token grid of DPT, which corresponds to 16×16 patches. Features are extracted from intermediate ResNet-50 blocks at 1/4 and 1/8 of the input resolution; the latter initializes segment token embeddings, while both are passed to the final decoder via skip connections. This entire preprocessing and tokenization pipeline is applied consistently in all experiments.

Architecture. With graph pooling and unpooling layers, the encoder consists of three stages, each with two ViT blocks followed by graph pooling, progressively reducing the number of segment tokens to 256, 128, and 64. The decoder mirrors this with unpooling and receives skip connections from the corresponding encoder stages. We treat this token-count schedule as a generic multiscale scaffold that fixes the capacity per level; the pooling assignments and the content routed into each segment remain learned and image-dependent.

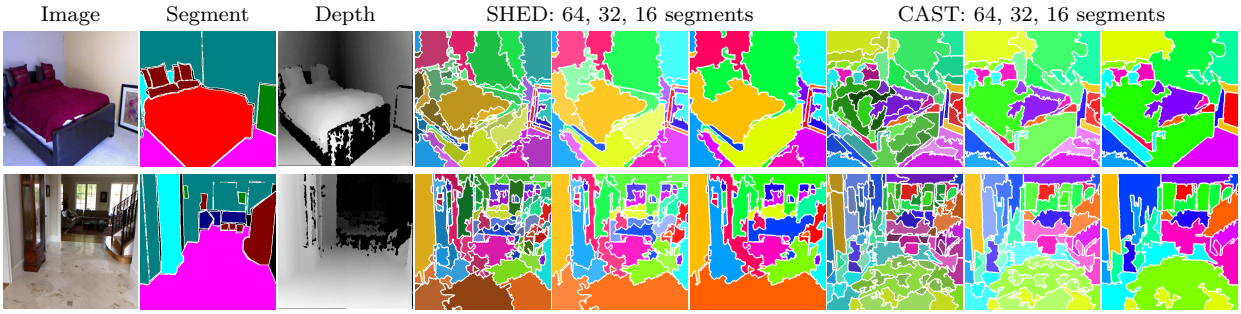

Figure 5: **SHED learns depth-aware segment hierarchies, while CAST relies on visual cues.** We compare segmentations from SHED and CAST (Ke et al., 2024) at the same hierarchy levels: 64, 32, and 16 segments. SHED captures meaningful part structures, such as separating the blanket and pillow from the bed (row 1). It also decomposes large structures like the floor based on depth, grouping nearby regions into a single large segment while dividing distant areas into smaller ones (row 2). In contrast, CAST relies on appearance cues and fails to capture geometric structure. For instance, it groups white floor regions by color but divides them arbitrarily, ignoring depth. These results highlight the value of depth supervision in learning 3D-aware segmentations.

We train SHED and DPT on NYUv2 using a batch size of 16 for 50 epochs with the Adam optimizer (Kingma & Ba, 2014) and a learning rate of 5e-5. With pretrained ResNet and ViT backbones, we follow DPT's default training recipe, including the scale-invariant logarithmic loss computed against ground-truth depth. At inference time, predicted depth maps at 384×384 resolution are bilinearly upsampled to 640×480 to match the ground-truth size.

4.2 Segment-consistent depth estimation

SHED generates structured depth maps using a learned segment hierarchy. We first visualize this structural alignment, then evaluate boundary quality and coherence, and finally demonstrate that this hierarchical approach improves per-pixel metrics and efficiency. Figure 3 shows how the segment hierarchy in SHED benefits geometry. The learned segments capture contours of objects, such as desks in a classroom, allowing the depth to clearly separate them from the floor. They also decompose larger structures, like tables, into parts, enabling smooth depth transitions from front to back. This suggests that structure guides depth prediction toward more accurate and interpretable results.

Boundary accuracy. We assess the structural quality of SHED by comparing occlusion boundaries against DPT on the NYUv2-OC++ dataset (Ramamonjisoa et al., 2020). Figure 4 shows predicted depth maps and their occlusion boundaries, extracted using a Canny edge detector (Canny, 1986). For quantitative evaluation, we follow the standard protocol (Koch et al., 2018) and compute the average Chamfer distance (Fan et al., 2017) in two directions: from prediction to ground truth, and vice versa. SHED produces sharper contours and outperforms DPT on both metrics with its fine-grained segmentation.

Intra-segment coherence. Beyond boundary, we evaluate how coherently depth values vary within each segment. We employ object-wise depth accuracy and error, treating ground-truth semantic segmentation masks as structural references on NYUv2 (Nathan Silberman & Fergus, 2012). As shown in Figure 4, SHED produces smoother depth variations within segments. This is reflected quantitatively in Table 1. SHED achieves lower error and higher accuracy within segments, confirming that our hierarchical pooling promotes smoother and more coherent depth predictions.

Per-pixel metrics & generalization. Table 2 reports standard per-pixel depth metrics under both in-domain and cross-domain settings. Under in-domain evaluation on NYUv2, SHED outperforms the DPT baseline and prior approaches that incorporate segmentation cues for depth estimation. For example, methods such as Simsar et al. (2022) enforce object-level consistency via over-segmentation, while multi-task frameworks like Mousavian et al. (2016) jointly supervise semantics and depth. The benefits of structural modeling become pronounced under cross-domain evaluation. Despite relying only on ImageNet pre-training,

Table 4: **SHED attains the best boundary accuracy, even against large foundation models.** Occlusion-boundary error ($\varepsilon_a/\varepsilon_c$, lower is better) on NYUv2-OC++. Despite using only ImageNet-1K pre-training, SHED trained on HyperSim achieves the lowest error on *both* metrics, indicating that *structured* decoding captures object boundaries more effectively than scaling data or model size.

Method	Pre-training / Setting	$\varepsilon_a \downarrow$	$\varepsilon_c \downarrow$
Controlled, trained on NYUv2 (IN-1K)			
DPT (Ranftl et al., 2021)	IN-1K	6.395	1.438
SHED (ours)	IN-1K	5.713	0.608
Foundation models, zero-shot			
Marigold (Ke et al., 2025)	LAION-5B	6.676	2.603
Depth Anything v2 (Yang et al., 2024)	LVM-142M	5.459	0.290
UniDepth v2 (Piccinelli et al., 2025)	large-scale	5.399	0.259
Trained on HyperSim			
Depth Anything v2 (Yang et al., 2024)	LVM-142M	5.763	0.682
SHED (ours)	IN-1K	**5.214**	**0.232**

Figure 6: **SHED produces structured 3D reconstructions.** We visualize 3D point clouds reconstructed from single-view depth maps, following the semantic scene completion protocol (Song et al., 2017), using predictions from DPT, SHED, and the ground truth on NYUv2 examples. Frontal views (cols 2-4) show that DPT fails to preserve planar structures, producing curved wall boundaries, whereas SHED more accurately recovers straight lines. This difference is more apparent in the bird's-eye views (cols 5-7): DPT yields warped surfaces, while SHED produces flatter layouts.

SHED markedly improves over its controlled DPT baseline under synthetic-to-real transfer from HyperSim to NYUv2. It further remains competitive with, and on most metrics surpasses, the much larger contextual references Depth Anything v2 (Yang et al., 2024), which leverages a DINOv2 (Oquab et al., 2023) encoder, and Marigold (Ke et al., 2025), pre-trained on billions of images. We emphasize that this transfer setting is practically meaningful precisely because recent depth pipelines themselves rely heavily on synthetic-to-real transfer during training; it probes whether structural knowledge learned from synthetic supervision remains transferable to real scenes, rather than measuring the native zero-shot ability of the reference models. These results suggest that incorporating explicit structural constraints offers a data-efficient path to robust depth estimation across domains. We report results on a diverse set of benchmarks in Section C.

Comparison with foundation models. A recurring question is how SHED relates to large depth foundation models. We therefore compare occlusion-boundary accuracy against Depth Anything v2 (Yang et al., 2024), Marigold (Ke et al., 2025), and UniDepth v2 (Piccinelli et al., 2025) (Table 4). Although these models are pre-trained on orders of magnitude more data, SHED trained on HyperSim attains the best boundary accuracy and consistency of all compared models, despite using only ImageNet-1K pre-training. This confirms that structured decoding captures object boundaries more effectively than scaling data or model size. We retain DPT as our primary controlled baseline because it isolates the effect of the proposed design under matched pre-training and data, whereas foundation models conflate architecture with large-scale supervision.

Computational efficiency. For a fair and hardware-independent comparison of structural complexity, we report both the number of parameters and the computational cost in FLOPs. DPT-Hybrid contains 41.88 million parameters and requires 135.0 GFLOPs. In contrast, SHED uses 46.60 million parameters but reduces the computation to 103.2 GFLOPs, which is approximately a **24% decrease in FLOPs**. This substantial reduction demonstrates that SHED is structurally more efficient and achieves lower theoretical latency despite

Table 5: **3D alignment induced by structured depth estimation.** We compute the Chamfer distance between point clouds from the predicted and ground-truth depths. SHED achieves lower errors than DPT, indicating improved geometric alignment.

Method	Precision / Recall ↓
DPT	0.171 / 0.251
SHED (ours)	**0.158** / **0.244**

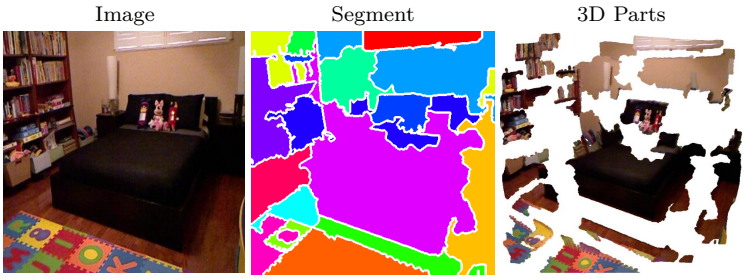

Figure 7: **SHED discovers 3D part structures.** Concurrent segmentation and depth estimation enable part-level decomposition of the 3D point clouds.

Table 6: **Semantic segmentation comparison results on ADE20K.** Segment hierarchies learned by SHED improve alignment with semantic regions and boundaries on ADE20K.

Method	mIoU	boundary F-score
ViT-small	41.7	–
CAST	43.1	36.5
SHED (ours)	**44.5**	**37.7**

having a slightly larger parameter count. For context, Depth Anything v2 requires 57.5 GFLOPs (small) and 201.1 GFLOPs (base), so SHED is also substantially cheaper than the base foundation model while remaining competitive in cross-domain transfer. To address the cost of the superpixel preprocessing explicitly, we also report *end-to-end* latency including the SEEDS step: SHED runs at 22.8 ms per image at 224×224 (≈44 FPS), and SEEDS accounts for less than 1% of inference time and can be parallelized across CPU cores. The preprocessing is thus not a practical bottleneck, and is not intrinsic to the design: it can be further reduced with GPU-accelerated superpixel implementations.

Ablation. We conducted an ablation study to isolate the contributions of the hierarchical clustering in the encoder and the hierarchy reversing in the decoder. To verify the necessity of our proposed decoder, we experimented with a variant of SHED by removing the progressive unpooling process in the decoder. Table 3 shows removing the reverse hierarchy leads to a sharp performance degradation, falling significantly behind the DPT baseline. This demonstrates that the coarse-to-fine unpooling mechanism in the decoder is essential to recover fine-grained spatial details from the grouped representations. We note that the reverse decoder is not a generic module that can be bolted onto an arbitrary backbone: its unpooling operations are defined by the soft assignment matrices produced during the encoder's grouping. Attaching it to a standard ViT or DPT encoder, which has no such grouping history, is therefore architecturally ill-defined: the encoder grouping and the reverse decoder are two halves of a single mechanism, and it is their inversion, not the grouping alone (which follows CAST (Ke et al., 2024)), that constitutes our contribution.

4.3 Structure-aware representation learning

Depth-aware image segmentation. We analyze the segment hierarchy learned by SHED by comparing it to CAST, an encoder trained for image recognition using segment-based representations. Specifically, we use CAST, trained on ImageNet (Deng et al., 2009) with the MoCo-v3 objective (Chen et al., 2021), a self-supervised learning by instance discrimination (Wu et al., 2018) that clusters visually similar images. To ensure a fair comparison, we adapt the graph pooling layers of SHED to generate the same number of segments (64, 32, and 16) from 196 superpixels at a 224×224 resolution.

Figure 5 shows that SHED learns hierarchical structures that align with scene geometry. It separates objects like blankets and decomposes large structures such as floors into segments that reflect their spatial extent. In contrast, CAST groups regions based on appearance. For example, it clusters white floor areas by color but fails to account for geometric cues. We attribute this difference to the training objective. CAST learns segments through image-level recognition, while SHED is guided by dense prediction.

Table 7: **SHED generalizes beyond depth to optical flow.** We replace RAFT's CNN encoder with the SHED encoder (RAFT-SHED) and train on KITTI-15 under RAFT's original setup. RAFT-SHED substantially improves over RAFT and its DPT-based variant, and surpasses the prior state-of-the-art Flow-Anything, indicating that the segment-hierarchy inductive bias transfers across dense prediction tasks. Lower is better for both metrics.

Method	EPE ↓	F1-all ↓
Flow-Anything (Liang et al., 2025) *(prior SOTA)*	1.79	6.98
RAFT (Teed & Deng, 2020)	5.04	17.4
RAFT-DPT	1.98	8.35
RAFT-SHED (ours)	**1.61**	**6.11**

Semantic segmentation. We further evaluate whether the learned segment hierarchies align with semantic structures on the ADE20K dataset (Zhou et al., 2017). Following the CAST evaluation protocol, we report region-level mean IoU and boundary F-score between the predicted segments and ground-truth annotations. As shown in Table 6, SHED outperforms CAST on both metrics, and the plain ViT baseline in mIoU, under the identical protocol. Notably, this improvement cannot be attributed solely to encoder pre-training. While both models leverage segment hierarchy, SHED incorporates a hierarchical decoding pipeline that progressively reverses the segment hierarchy, enabling fine-grained spatial details to be recovered from grouped representations. This coarse-to-fine decoding process facilitates more accurate alignment with semantic boundaries, leading to improved segmentation quality without requiring explicit semantic supervision.

4.4 3D scene reconstruction with part structures

We demonstrate SHED's capability for 3D scene understanding. While plausible pixel values may suffice for 2D depth estimation, accurate and structured depth is particularly critical when projected into 3D space. Accordingly, SHED enables high-quality 3D reconstruction and supports unsupervised 3D part discovery through segmentation. To evaluate the structural quality of depth maps, we project them into 3D point clouds on the NYUv2 dataset (Nathan Silberman & Fergus, 2012), following the semantic scene completion protocol (Song et al., 2017) and using NYUv2 camera intrinsics. For interpretability, all depth values are scaled by 1/1000. Figure 6 shows that SHED produces cleaner reconstructions with sharper boundaries and flatter surfaces that better align with ground truth geometry, whereas DPT yields curvier, less faithful shapes. We quantify reconstruction performance with the Chamfer distance (Fan et al., 2017) in both directions. Table 5 shows that SHED consistently achieves lower distances than DPT, confirming its advantage in structured 3D prediction. By jointly predicting segmentation and depth, SHED lifts 2D parts into 3D space, enabling part-level decomposition of scenes.

Figure 7 shows an example from NYUv2, where segments corresponding to objects form coherent 3D structures in point clouds. This demonstrates SHED's potential for unsupervised 3D part reasoning, a key capability for interactive and dynamic scene understanding (Mo et al., 2019).

4.5 Generalization to optical flow

While our analysis so far centers on depth, segment-hierarchy decoding is a general mechanism for dense prediction. As a final demonstration, we apply it to optical flow by replacing RAFT's CNN encoder with SHED (RAFT-SHED), trained under RAFT's original protocol on KITTI-15. The features fed to RAFT's correlation volume are taken *after* the reverse-hierarchy unpooling, so RAFT-SHED exercises the full encoder–decoder mechanism, including the reverse decoder, rather than the encoder grouping alone. As shown in Table 7, RAFT-SHED improves markedly over both RAFT and a DPT-based variant, and surpasses the recent state-of-the-art Flow-Anything (Liang et al., 2025). Since optical flow requires separating moving objects from the static background, this indicates that the learned segment hierarchy provides structural cues that transfer to dense prediction tasks other than depth.

5 Conclusion

We shed light on the role of segmentation in depth estimation. SHED learns a segment hierarchy in the encoder and reverses it in the decoder to predict dense maps. This results in depth maps with segment-consistent structure, structure-aware representations, and coherent 3D scenes with interpretable parts. Our principle of unifying reconstruction and reorganization offers a new direction for 3D vision and robotics, particularly for tasks that require fine-grained interaction with physical components. Additional results and limitations are discussed in Section E.

Broader impact. This technology can enhance the 3D environmental perception of autonomous driving systems, thereby improving road safety, and can help robots interact more safely and efficiently with their surroundings. Before deploying this model in real-world scenarios, it must undergo rigorous and thorough validation for robustness and safety across a wide range of conditions. We discuss broader impacts further in Section E.

Reproducibility. Full implementation, training, and evaluation details are provided in Section A. We will release code and trained models upon publication.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3378–3385. IEEE, 2012.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4009–4018, 2021.

Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 2624–2632, 2019.

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9640–9649, 2021.

Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 103–119, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Zhiwei Deng, Ting Chen, and Yang Li. Perceptual group tokenizer: Building perception with iterative grouping. *arXiv preprint arXiv:2311.18296*, 2023.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ainaz Eftekhar, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Ani Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. *arXiv preprint arXiv:2311.04193*, 2023.

David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pp. 2650–2658, 2015.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21795–21806, 2024.

SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2002–2011, 2018.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017.

Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. *arXiv preprint arXiv:2002.12319*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xingzhe He, Bastian Wandt, and Helge Rhodin. Ganseg: Learning to segment by unsupervised hierarchical image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1225–1235, 2022.

Shaul Hochstein and Merav Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5):791–804, 2002.

Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7986–7994, 2018.

Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7334–7344, 2019.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Hyunwoo Kang, Sangwoo Mo, and Jinwoo Shin. Oamixer: Object-aware mixing layer for vision transformers. *arXiv preprint arXiv:2212.06595*, 2022.

Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *arXiv preprint arXiv:2505.09358*, 2025.

Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2571–2581, 2022.

Tsung-Wei Ke, Sangwoo Mo, and X Yu Stella. Learning hierarchical image segmentation for recognition and by recognition. In *The Twelfth International Conference on Learning Representations*, 2024.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2041–2050, 2018.

Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *IEEE Transactions on Image Processing*, 2024.

Yingping Liang, Ying Fu, Yutao Hu, Wenqi Shao, Jiaming Liu, and Debing Zhang. Flow-anything: Learning real-world optical flow estimation from large-scale single-view images. *arXiv preprint arXiv:2506.07740*, 2025.

Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10): 2024–2039, 2015.

Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ri-Zhao Qiu, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Yunze Man, Shuhong Zheng, Zhipeng Bao, Martial Hebert, Liangyan Gui, and Yu-Xiong Wang. Lexicon3d: Probing visual foundation models for complex 3d scene understanding. *Advances in Neural Information Processing Systems*, 37:76819–76847, 2024.

David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 909–918, 2019.

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. *arXiv preprint arXiv:1812.10889*, 2018.

Sangwoo Mo, Hyunwoo Kang, Kihyuk Sohn, Chun-Liang Li, and Jinwoo Shin. Object-aware contrastive learning for debiased scene representation. *Advances in Neural Information Processing Systems*, 34: 12251–12264, 2021.

Arsalan Mousavian, Hamed Pirsiavash, and Jana Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 611–619. IEEE, 2016.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025.

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

Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14648–14657, 2020.

Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5571–5584, 2023.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12179–12188, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10912–10922, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd Van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in neural information processing systems*, 35:9512–9524, 2022.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. *arXiv preprint arXiv:2209.14860*, 2022.

Baifeng Shi, Trevor Darrell, and Xin Wang. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2102–2112, 2023.

Enis Simsar, Evin Pınar Örnek, Fabian Manhardt, Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-aware monocular depth prediction with instance convolutions. *IEEE Robotics and Automation Letters*, 7(2):5389–5396, 2022.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 567–576, 2015.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1746–1754, 2017.

Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6243–6252, 2017.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin De Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pp. 13–26. Springer, 2012.

Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2162–2171, 2019.

Max Wertheimer. Laws of organization in perceptual forms. 1938.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18134–18144, 2022.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5684–5693, 2019.

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1983–1992, 2018.

Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3916–3925, 2022.

Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 340–349, 2018.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13116–13125, 2020.

A Implementation Details

A.1 Training details

We train our model on the NYUv2 dataset (Nathan Silberman & Fergus, 2012) using the official training split. Each RGB image is first cropped to remove invalid boundaries (coordinates: 43, 45, 608, 472), then resized to 384×384 resolution. The corresponding depth maps undergo the same spatial preprocessing and are normalized by dividing raw depth values by 1000.

For data augmentation, we apply horizontal flipping with a probability of 0.5, gamma correction with $\gamma \in [0.9, 1.1]$, brightness scaling using a random factor from $[0.75, 1.25]$, and per-channel color jittering with multiplicative factors in $[0.9, 1.1]$. After augmentation, random spatial crops of size 384×384 are applied to both images and depth maps.

For tokenization, we generate superpixels using OpenCV's SEEDS (Van den Bergh et al., 2012) algorithm. Each image is segmented into 676 superpixels using a single-level hierarchy (`num_levels=1`) and a histogram bin size of 5. The algorithm is run for 50 iterations to refine superpixel boundaries.

A.2 Evaluation details

We evaluate on the official NYUv2 (Nathan Silberman & Fergus, 2012) test split, which contains 654 images. All evaluations use an input resolution of 384×384 pixels, with depth values clamped to the range $[10^{-3}, 10.0]$.

Per-pixel depth metrics. We compute standard depth estimation metrics over valid pixels where ground truth depth is available. Depth error metrics include AbsRel (mean absolute relative error), RMSE (root mean squared error), and Log10 (mean absolute logarithmic error). Accuracy is measured using threshold metrics δ^1, δ^2, and δ^3, which denote the percentage of pixels where the predicted-to-ground-truth depth ratio is below 1.25, 1.25^2, and 1.25^3, respectively. All metrics are computed with numerical safeguards, including epsilon clamping at 1e-6 to prevent division by zero and log-domain errors.

Occlusion boundary. We follow the evaluation protocol of the NYUv2-OC++ dataset (Ramamonjisoa et al., 2020) to assess occlusion boundary accuracy. Each predicted depth map is first min-max normalized, followed by the application of the OpenCV Canny edge detector (Canny, 1986) with low and high thresholds of 100 and 200 to produce a binary mask of predicted boundary pixels. Using the ground-truth boundary labels from NYUv2-OC++, we compute two metrics: ε_a (accuracy), the average distance from each predicted edge pixel to the nearest ground-truth edge; and ε_c (consistency), the average distance from each ground-truth edge pixel to the nearest predicted edge. Both are reported in squared pixels, where lower values indicate better alignment, and 0 denotes perfect correspondence.

3D scene reconstruction. We reconstruct 3D point clouds by back-projecting each pixel of the predicted depth maps into 3D space using the known camera intrinsics from the NYUv2 dataset, following the standard protocol for semantic scene completion (Song et al., 2017). To evaluate reconstruction quality, we compute the Chamfer distance between the predicted and ground truth point clouds. Recall quantifies how well the predicted points cover the ground truth surface, while precision measures how accurately the predicted points align with the true geometry.

Detailed evaluation protocol for **layout-aware retrieval** is provided in Section B, respectively.

B Layout-aware Retrieval

Our architecture not only improves depth prediction but also facilitates structure-aware representation learning. First, SHED learns features that reflect scene layout, enabling more accurate layout-aware image retrieval than DPT (Ranftl et al., 2021). Second, its segment hierarchy captures geometric cues informed by depth supervision, whereas CAST (Ke et al., 2024) relies on visual cues.

a) Top-5 retrieval examples

b) Top-K retrieval accuracy (%)

Method	Top-1	Top-3	Top-5
Scene retrieval			
DPT	45.2	69.7	77.2
SHED (ours)	**60.5**	**78.7**	**87.0**
Frame retrieval (k=5)			
DPT	18.5	31.0	38.3
SHED (ours)	**30.5**	**42.3**	**48.3**

Figure 8: **SHED learns layout-aware representations through depth supervision.** We evaluate image retrieval on NYUv2 based on cosine similarity between class tokens from the final ViT block. **a)** Top-5 results (ranked left to right), with similarity scores shown below. SHED retrieves images with similar layouts, such as a central desk and a rear bookshelf, while DPT retrieves unrelated scenes. **b)** Top-K accuracy at the scene and frame level ($k = 5$), where the targets are different views from the same scene or nearby frames. SHED significantly outperforms DPT in all settings, indicating that our depth-guided segmentation effectively encodes spatial layout.

We provide a detailed explanation of our proposed metric, *layout-aware retrieval*, which evaluates the structural quality of learned representations by measuring how well the model retrieves frames from the same 3D scene in a video, either across the full sequence or within nearby frames.

To compute this metric, we extract the `[CLS]` token from the final layer of the vision transformer in DPT and SHED for each image, denoted as $\mathbf{z}_{\text{cls}} \in \mathbb{R}^D$, and apply l_2 normalization. We then construct a full pairwise similarity matrix, where each entry is computed as the cosine similarity between l_2-normalized image embeddings: $S_{ij} = \mathbf{z}_{\text{cls}}^{(i)} \cdot \mathbf{z}_{\text{cls}}^{(j)}$.

Retrieval performance is evaluated in two settings: *scene-level* and *frame-level*. In the scene-level setting, the goal is to retrieve other frames from the same annotated scene, testing the model's ability to maintain structural consistency under varying viewpoints. In the frame-level setting, each frame is treated as an independent query, focusing on retrieving visually similar frames regardless of scene membership. We additionally define a frame-k variant, where retrieval is restricted to the k temporally adjacent frames, allowing us to assess the model's sensitivity to local layout changes. We report top-K nearest neighbor retrieval accuracy on the NYUv2 (Nathan Silberman & Fergus, 2012) test set.

To further evaluate the robustness of layout-aware retrieval, we vary both the candidate set size (k) and the top-K threshold. A smaller k imposes a stricter constraint, requiring the model to identify the most similar frame from a limited pool. As shown in Figure 9, SHED consistently outperforms DPT across all settings. Performance improves for both methods as k increases, with the largest gap observed at low top-K values. These results suggest that SHED captures layout similarity more precisely and excels at retrieving the most relevant match.

C Evaluation on Diverse Benchmarks

We extend our experiments to a mixed-domain training setup using Virtual KITTI Cabon et al. (2020), HyperSim Roberts et al. (2021), and MegaDepth Li & Snavely (2018). We train the SHED using a uniform sampling strategy across all datasets for 20 epochs with batch size of 64. We then evaluate this model in a fully zero-shot manner on three unseen datasets, including KITTI (outdoor) Geiger et al. (2013), NYUv2 (indoor) Nathan Silberman & Fergus (2012), SUN-RGBD (indoor) Song et al. (2015).

The results below demonstrate that SHED consistently outperforms the DPT baseline across domains, including both indoor and outdoor scenes. The segment-hierarchy inductive bias of SHED is not specific to the NYUv2 domain but transfers effectively to outdoor and mixed-domain settings, consistently outperforming the baseline.

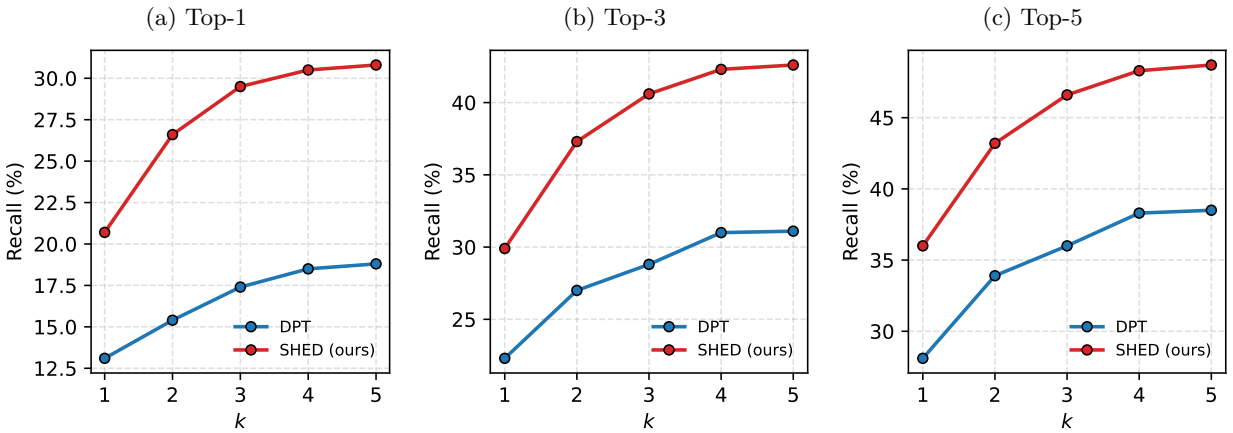

Figure 9: **Frame-k recall with varying temporal range k.** We report Top-K retrieval recall ($K \in \{1, 3, 5\}$) for DPT and SHED across temporal ranges $k \in [1, 5]$. The consistent gains highlight the robustness of our depth-supervised representation to spatial and viewpoint changes.

Table 8: **SHED improves the mixed-domain setting.** We report standard depth accuracy and error metrics on a more diverse set of benchmarks. When trained on the mixed-domain data and evaluated zero-shot on unseen datasets, SHED consistently outperforms DPT across both indoor and outdoor scenes.

Method	KITTI		NYUv2		SUN-RGBD	
	AbsRel ↓	$\delta > 1.25$ ↑	AbsRel ↓	$\delta > 1.25$ ↑	AbsRel ↓	$\delta > 1.25$ ↑
DPT	0.286	0.475	0.247	0.559	8.58	0.169
SHED (ours)	**0.272**	**0.500**	**0.244**	**0.571**	**7.16**	**0.190**

D Additional Visualizations

D.1 More visualizations of depth maps

Figure 10: **Comparison of depth maps.** SHED captures more accurate shapes with sharper boundaries, whereas DPT produces blurrier results.

D.2 More visualizations of image retrieval

Figure 11: **Comparison of top-1 image retrieval results.** SHED retrieves samples that are more structurally similar to the query, indicating that its global embedding effectively captures scene layouts. In contrast, DPT focuses more on visual appearance, as shown in column 3, where it retrieves an image from a different scene that shares a similar color of blue.

D.3 More visualizations of 3D reconstruction

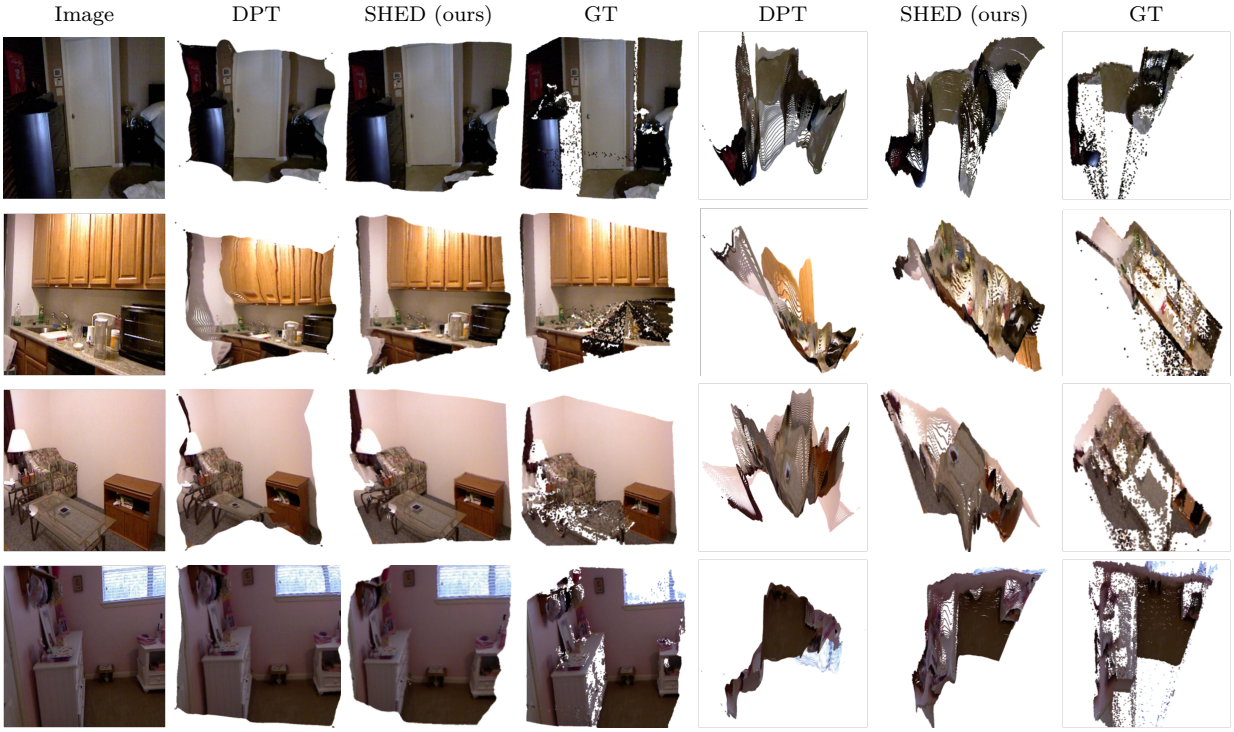

Figure 12: **Comparison of 3D reconstruction results.** Frontal views (cols 2–4) and bird's-eye views (cols 5–7). DPT yields curved wall boundaries in the frontal views, which lead to distorted 3D reconstructions visible in the bird's-eye views. In contrast, SHED produces sharp depth edges that preserve straight object contours and more faithfully represent the ground-truth 3D geometry.

E Limitations and Broader Impacts

E.1 Limitations and future works

SHED uses a fixed token-count schedule: the number of tokens pooled at each level is specified in advance, and the unpooling process assumes a fixed correspondence between pooled and unpooled tokens. We stress that this should be read as a *generic multiscale scaffold* rather than a dataset-specific structural prior, analogous to the fixed stage resolutions in CNNs or the fixed token grids in ViTs: the token counts set the *capacity* at each level, while the learned pooling assignments and the content routed into each segment remain image-dependent. The relevant question is therefore not whether the scaffold is fixed in principle, but whether the model is excessively brittle to it in practice. Our ablation (Table 3) shows that removing the reverse decoder degrades performance sharply, indicating that the structural gains are driven by the reverse hierarchical decoding itself rather than by a precisely tuned schedule.

Nonetheless, the schedule is not yet content-adaptive: regardless of whether a scene is a near-empty wall or a cluttered desk, the encoder compresses features into the same fixed number of segment tokens, which may limit adaptability to scenes with highly varying structural complexity. The most promising direction for future work is therefore *scene-adaptive scheduling*, in which the number of tokens at each level is determined dynamically from scene complexity. In addition, exploring more flexible unpooling mechanisms that do not require a strict one-to-one correspondence between pooled and unpooled tokens could further improve robustness to diverse scene layouts. Finally, the current tokenizer is initialized with a non-differentiable superpixel algorithm (SEEDS); while our analysis shows the learned representation is not narrowly tied to this choice, a fully differentiable, end-to-end tokenizer remains an open direction.

E.2 Broader impacts

Structured understanding of the 3D world and accurate depth estimation are central challenges in AI, with direct impact on safety-critical applications such as autonomous driving, augmented reality, and robotics. In practice, failures in these systems often result not from a lack of data but from insufficient structured reasoning, making predictions vulnerable to occlusion, unusual viewpoints, and dynamic environments. Our framework promotes geometry-aware perception by producing robust and interpretable depth estimates that align with scene structure. This can improve reliability in complex real-world settings and lead to systems with more transparent failure modes.