Scaling Dense Prediction with Latent Decoding

Xiaoyang Wu[1], Yixing Lao[1], Chengyao Wang[2], Senqiao Yang[2],
Yujia Zhang[1], and Hengshuang Zhao[1]

[1] The University of Hong Kong
[2] The Chinese University of Hong Kong
https://pointcept.org/l-dpt

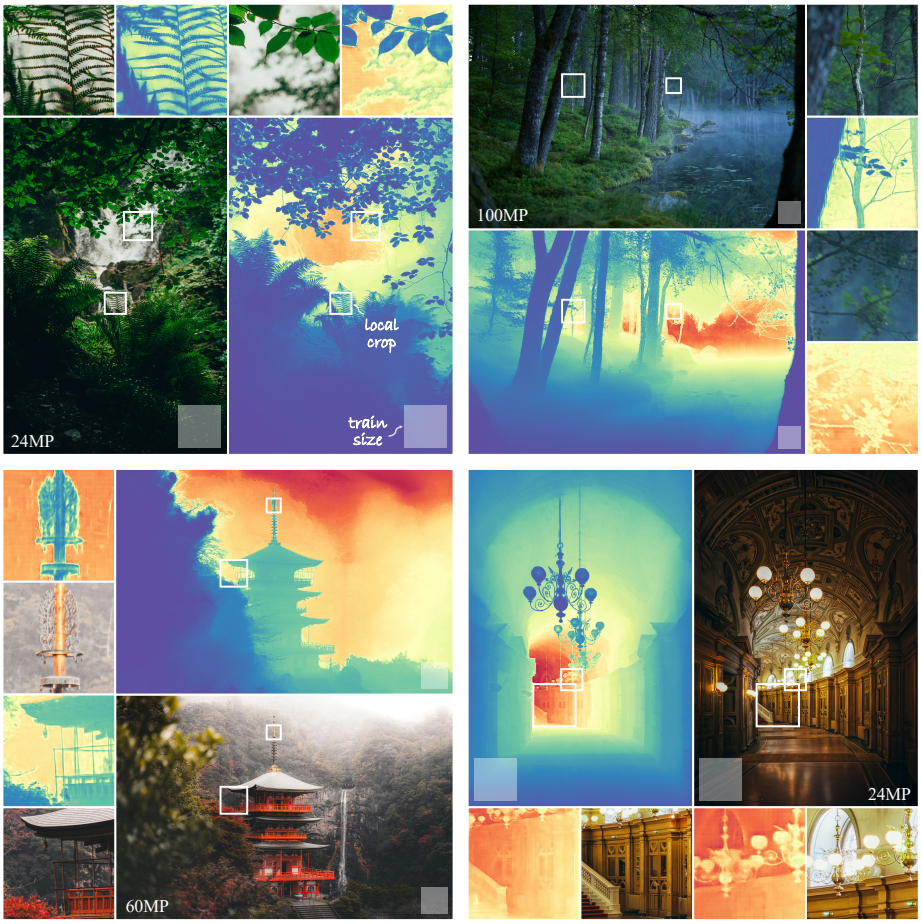

Fig. 1: Train at 768, run at 100 megapixels. Our L-DPT concentrates modeling capacity in latent decoding and renders to pixels with a lightweight readout. Trained only at a fixed 768 training resolution, it remains stable, efficient, and detail-preserving when applied to ultra-high-resolution photographs up to 100MP. At this scale, the output is not a blurred upscale: fine structures that are often lost at standard resolutions (e.g., leaves, vines, and branches) remain sharp and coherent. The 100MP limit reflects the highest-resolution photographs available in our tests, not the model's capacity.

Abstract. Modern dense predictors largely follow a pyramidal pixel-decoding paradigm: they progressively lift low-resolution representations to dense, high-resolution outputs. While effective, this design tightly couples computation with output resolution, making dense prediction increasingly costly to scale. In this work, we argue that strong dense prediction can be achieved by separating capacity from resolution: performing high-capacity reasoning in a compact latent space, and reading out to pixels with a lightweight operator. To instantiate this principle, we introduce *Latent Dense Prediction Transformer* (L-DPT), a minimal architecture that decouples dense decoding from pixel-space rendering. Instead of progressively decoding toward the pixel grid, L-DPT performs attention-based dense decoding entirely in latent space, keeping computation largely stable as resolution increases while enabling more flexible feature integration from the encoder. The final output is produced via a lightweight pixel readout, such as pixel shuffle, so scaling to extreme resolutions does not amplify decoding cost. As a result, L-DPT achieves stronger dense prediction accuracy while enabling inference on professional ultra-high-resolution images up to 100 megapixels. More broadly, we view L-DPT as an instance of a scalable design principle rather than a task-specific decoder for dense prediction.

Keywords: Dense Prediction · Scalable Architecture

1 Introduction

Modern dense predictors [3, 4, 26, 41, 42, 43] largely follow a pyramidal pixel-decoding paradigm [35,47]: they progressively lift low-resolution representations to dense, high-resolution outputs. While effective, this design tightly couples computation with output resolution [11], making dense prediction increasingly costly to scale. This paradigm underlies a wide range of high-performing systems for depth estimation [30], semantic segmentation [11,22,27,59], and image generation [25], where decoding is realized as a cascade of upsampling, multi-scale fusion, and pixel-space refinement. As dense prediction becomes a core primitive for robotics, autonomous driving [7,20], mixed reality [1,13], and professional imaging, the desired operating scale continues to grow along multiple axes, including resolution, capacity, and feature integration. Progressive pixel-space decoding becomes the dominant tax on scalability, manifesting not only as rising computational cost, but as an efficiency wall and a capability wall.

We argue that the fundamental limitation is not the need for dense output, but the entanglement between capacity and the pixel grid. Pyramidal pixel decoding places high-capacity computation directly on feature maps whose size tracks the output grid, so both compute and memory inevitably scale with resolution. This coupling forces a trade-off between prediction quality and scalability, and it constrains how flexibly the decoder can integrate task-relevant information from the encoder. In this work, we advocate a different design principle for dense prediction: separate reasoning from rendering by concentrating modeling capacity in a compact latent space, and reading out to pixels with a lightweight

operator. In other words, dense prediction should reason in latents and render to pixels, so scaling resolution becomes a matter of readout rather than a tax on decoding. That is, *put capacity in latents, and let pixels be the readout.*

To instantiate this principle, we introduce L-DPT, a minimal architecture that decouples dense decoding from pixel-space rendering. L-DPT casts dense prediction as a two-stage process. First, it performs high-capacity "reasoning" in a compact latent representation, using attention-based decoding on a compact latent representation, keeping the bulk of computation off the pixel grid. This latent decoder carries most of the modeling capacity and is designed to flexibly integrate task-relevant information from the encoder, without committing to a progressive pixel-space pyramid. Second, L-DPT produces pixels via a lightweight "rendering" operator, kept simple and modular, e.g., a pixel-shuffle readout. In doing so, L-DPT relocates dense decoding from pixel-aligned grids to latents and turns pixel-space rendering into an explicit, minimal design choice.

Our scalability gains are not accidental. They are a direct consequence of decoupling capacity and fusion from the pixel grid, and treating resolution as a readout problem, so that high-capacity computation is concentrated in latents rather than on the pixel grid. First, latent decoding makes adaptive feature fusion cheap and expressive: encoder signals from multiple levels can be integrated at a shared latent scale, without hard-wiring stage assignments in a progressive pixel pyramid. This expands the fusion space while keeping computation off the output grid, translating into stronger representations and delivering consistent accuracy gains on relative depth prediction and semantic segmentation. Second, once resolution is delegated to readout, generalization relies on positional modeling and training dynamics rather than pixel-grid computation. We couple strongly augmented RoPE-based positional modeling with a scalable training recipe, yielding a model trained at 768×768 while extrapolating far beyond the training grid. This supports stable inference on professional ultra-high-resolution imagery up to 100 megapixels.

2 Pilot Study: When Pyramids Stop Scaling?

We conduct a focused analysis of the pyramidal pixel-decoding paradigm through two strong representative baselines, Depth Anything V2 [61] and the DINOv3 [50] depther, both instantiated with DPT-style decoders. We keep model weights fixed and vary only input resolution, asking a simple question: when dense prediction is pushed to higher resolution, what breaks first?

This is not just a "slow model" story. The wall we hit is first a feasibility wall, and then a latency wall. Pyramidal decoding forces the system to materialize and fuse large pixel-aligned activations across multiple stages, so peak memory rises rapidly and soon approaches hardware limits. At the same time, latency does not degrade gracefully: beyond a certain regime (e.g., around 9MP and 17MP in our sweep), the decoder enters a much worse operating point, turning high-resolution inference into a systems bottleneck. Once this happens, the operating point is no

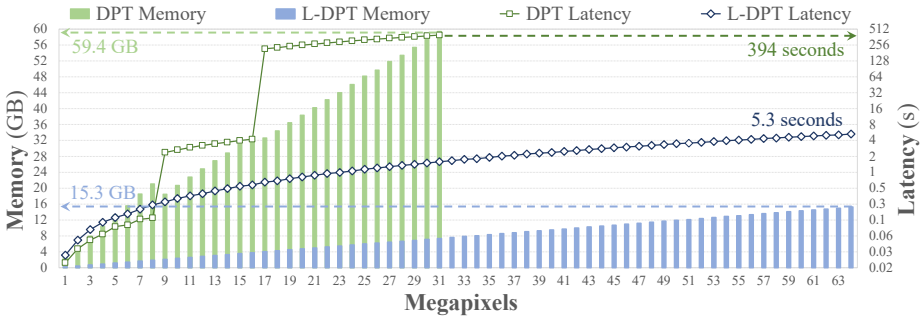

Fig. 2: Pixels scale. Efficiency does not. We benchmark decoder-side peak memory and latency during inference as input resolution increases, comparing a conventional DPT decoder with our latent decoder (L-DPT). DPT shows sharply rising memory consumption and pronounced latency regime changes as resolution grows, whereas L-DPT scales more smoothly across the full range. These results highlight the earlier efficiency wall of pyramidal pixel decoding.

longer determined by modeling preference, but by hardware constraints, forcing practical compromises such as resizing or tiling.

2.1 Feasibility: When Does the Efficiency Wall Hit?

We benchmark inference-time efficiency by sweeping input resolution and measuring the decoder-only peak GPU memory and latency for two decoder families: the conventional pyramidal pixel decoder (DPT) and our latent decoder (L-DPT). Fig. 2 summarizes the trend from low to ultra-high resolution in a single sweep. As the pixel grid grows, the gap widens quickly. At the high end of our sweep, DPT reaches 59.4 GB peak memory and 394 s latency, whereas L-DPT remains at 15.3 GB and 5.3 s, respectively. In practice, this gap already affects end-to-end feasibility: the DINOv3 depther (with ViT-7B encoder) with a DPT-style decoder OOMs on 32MP inputs on a B200 (192GB memory) GPU.

This is not just a "slow model" story. The wall we hit is first a feasibility wall, and then a latency wall. In DPT, large pixel-aligned activations must be materialized and fused across multiple stages, so peak memory rises rapidly with resolution and soon approaches hardware limits. At the same time, latency does not degrade gracefully: beyond a certain regime, it enters a much worse operating point, turning high-resolution inference into a systems bottleneck. By contrast, L-DPT keeps the heavy computation in latent space, which substantially reduces both memory growth and runtime escalation. As a result, the efficiency wall arrives much earlier for pyramidal pixel decoding, forcing compromises such as resizing or tiling, while latent decoding remains within a deployable regime.

2.2 Capability: When More Pixels Stop Helping?

Feasibility is only half of the scaling story. Even when inference remains possible, increasing resolution does not guarantee better dense prediction. We therefore visualize depth maps produced by DPT-style decoders across megapixel regimes

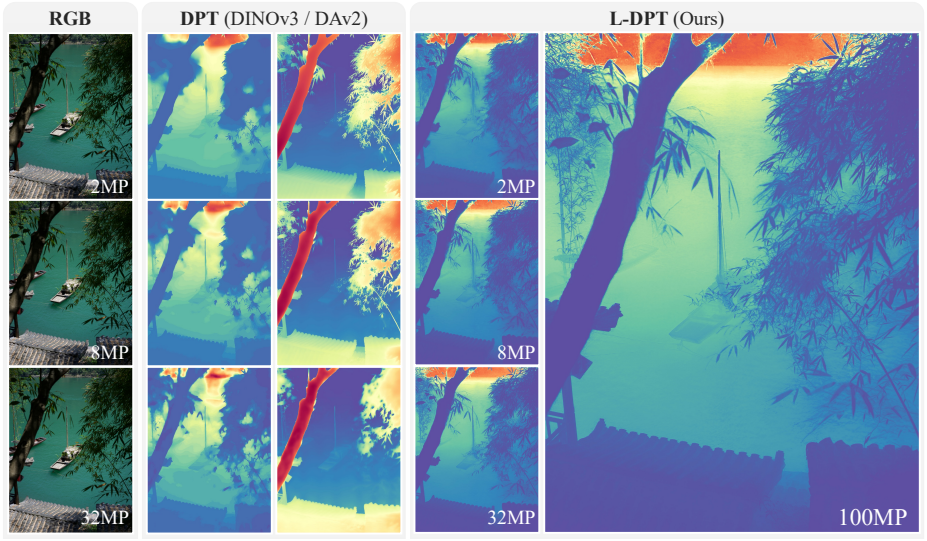

Fig. 3: More pixels, blurrier depth. We compare depth predictions across increasing input resolutions. From top to bottom, the inputs are evaluated at 2MP, 8MP, and 32MP; for DPT, the left column shows the DINOv3 depther and the right column shows Depth Anything V2 (disparity). Across both DPT-based models, increasing resolution does not yield sharper or more informative depth: the same coarse layout is largely preserved, while fine structures such as leaves, branches, and thin poles become increasingly smooth and washed out. By contrast, our L-DPT (trained only at 768px resolution) continues to benefit from denser inputs, remaining sharp and coherent from 2MP to 32MP and further scaling to 100MP, where additional pixels are translated into usable geometric detail rather than a blurrier prediction.

and ask a direct question: do additional pixels actually buy sharper boundaries, finer structures, and more faithful geometry?

Fig. 3 gives a clear answer: often, they do not. Across 2MP, 8MP, and 32MP inputs, the global depth layout barely changes. The decoder keeps reproducing essentially the same coarse scene interpretation, even as the input grid becomes much denser. But the more striking failure is local. Fine structures do not simply stop improving; they can get worse. Thin branches, leaves, and narrow poles remain unresolved, and at higher resolutions they often become even smoother, weaker, and more washed out. Instead of turning extra pixels into extra geometry, the decoder increasingly blurs away exactly the high-frequency structures that higher resolution should have helped recover.

This is the second wall. In pyramidal pixel decoding, more pixels do not necessarily mean more capability; beyond a certain regime, they can mean more cost for worse predictions. Once dense prediction is tied to progressive pixel-space refinement, scaling resolution can preserve the same coarse layout while making local geometry less distinct and less useful. The system is not extracting more structure from the added pixels. It is paying more to produce a denser, but blurrier, answer.

2.3 Paradox of the Pyramid

Our audit reveals two walls of pyramidal pixel decoding: an efficiency wall, where memory and latency rise sharply with resolution, and a capability wall, where more pixels stop improving prediction and can even make local geometry worse. This is a paradox of the pyramid. A design meant to refine across scales ends up tying both cost and quality to the pixel grid. This motivates a different hypothesis. Instead of progressively rebuilding dense predictions on the pixel grid, what if we place decoding capacity entirely in latent space, and let pixels be only the readout? If valid, this would move dense decoding off the pixel grid, making resolution primarily a readout cost instead of a decoder bottleneck. Scalability would then be determined by architecture.

We therefore take latent decoding as the working design principle of this work.

3 Latent Dense Prediction Transformer

3.1 Problem Setup and Notation

Given an image $x \in \mathbb{R}^{H\times W\times 3}$, a vision transformer produces a hierarchy of intermediate representations $\{F^{(s)}\}_{s=1}^{S}$ from S selected layers, where $F^{(s)} \in \mathbb{R}^{N\times C_s}$. For ViT-style encoders with patch size P, the token count is

$$N = \frac{H}{P}\frac{W}{P}. \tag{1}$$

Our goal is to predict a dense field $y \in \mathbb{R}^{H\times W\times K}$ (e.g., K=1 for depth, K semantic classes for segmentation) without progressive pixel-space pyramids.

3.2 Macro Design: Reason in Latents, Render to Pixels

As illustrated in Fig. 4, we decompose dense prediction into two stages: a high-capacity latent decoder that performs "reasoning" on a compact token grid, and a lightweight readout that performs "rendering" onto the output pixels. Concretely, rather than expanding the spatial resolution progressively, we maintain a latent grid aligned with the ViT patch layout as the sole spatial carrier throughout the decoding process. We denote the initial state of this latent grid as

$$Z^{(0)} \in \mathbb{R}^{N\times D}, \tag{2}$$

where D is the latent width. To maintain generality, $Z^{(0)}$ can be initialized either as fixed constants or learnable tokens. By default, we use zero initialization, though it can readily serve as a learnable visual prompt when adapting the decoder to different dense prediction tasks.

Given encoder features $\{F^{(s)}\}_{s=1}^{S}$, our latent decoder iteratively refines Z via attention-based feature integration,

$$Z^{(\ell)} = \mathcal{D}\Big(Z^{(0)}, \{F^{(s)}\}_{s=1}^{S}\Big), \tag{3}$$

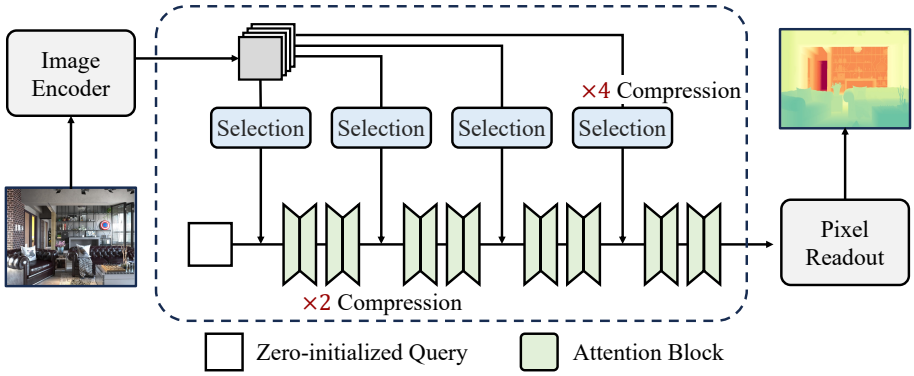

Fig. 4: L-DPT decodes in latents and reads out to pixels. Given an image encoder and its intermediate features, L-DPT keeps a single latent query stream as the sole carrier of decoding. At each stage, a lightweight selection module, implemented as a linear projection, compresses the encoder feature pool into stage-specific conditioning, which is injected into the latent stream and processed by attention blocks. Dense outputs are produced at the end through a lightweight pixel readout, making capacity and feature fusion live in latent space rather than on a progressive pixel-space pyramid.

where $\mathcal{D}$ operates entirely in latent space without a progressive pixel-space pyramid. The final dense output is then produced by a lightweight pixel readout

$$y = \mathcal{R}\Big(Z^{(L)}; H, W\Big), \tag{4}$$

where $\mathcal{R}$ maps latent tokens to a pixel-aligned field. This separation makes the role of computation explicit: capacity and feature fusion live in the latent decoder, while resolution is delegated to the readout.

3.3 Latent Feature Integration: Selection and Injection

A fixed latent grid only becomes useful if it can access the hierarchy encoded by a modern ViT. Conventional pyramidal decoders solve this by construction: they bind encoder layers to predetermined decoding stages and fuse features through a deep-to-shallow schedule coupled to progressive upsampling. In latent decoding, this binding is unnecessary. Once decoding happens on a single latent grid, feature integration becomes a lightweight, repeatable operation, and the decoder can decide what to pull from the encoder at each stage instead of following a hand-designed schedule.

Selection. We collect intermediate features from S encoder layers and treat them as a shared feature pool. Rather than assigning layers to stages, we form a unified representation by concatenating features along channels, and apply stage-specific linear projections to obtain a compact feature bank for each decoding stage. Concretely, letting $\bar{F} = [F^{(1)} \| F^{(2)} \| \cdots \| F^{(S)}]$, we compute

$$U^{(\ell)} = \bar{F} W_{\text{sel}}^{(\ell)}. \tag{5}$$

This simple operation serves as an adaptive selector over the entire encoder hierarchy. Each stage sees the same pool but learns a different projection, enabling re-selection and re-compression of encoder information without multi-resolution pyramids. The result is a clean separation: feature selection is performed in channel space, while spatial structure stays on single latent grids.

Injection. After selection, we inject $U^{(\ell)}$ into the latent state $Z^{(\ell)}$ using lightweight conditioning. This injection is intentionally minimal: its role is to expose the latent grid to encoder information, not to reintroduce dense pixel-space fusion. We consider three compatible mechanisms, including additive injection, cross-attention injection, and adaptive normalization injection. In our default setting, we adopt the simplest additive form,

$$Z^{(\ell)} \leftarrow Z^{(\ell)} + U^{(\ell)}, \tag{6}$$

which provides sufficient conditioning capacity for the downstream dense prediction tasks studied in this paper. The other mechanisms remain drop-in alternatives when richer conditioning is required, e.g., for multi-task training or heterogeneous outputs.

3.4 Latent Decoding with RoPE-Augmented Attention

Once encoder information is selected and injected, we refine the latent grid with a lightweight stack of attention blocks. The goal is simple: concentrate modeling capacity where it scales, and keep the decoder off the pixel grid. Each block performs self-attention over latent tokens to propagate context and enforce global coherence, followed by an FFN for refinement. Importantly, positional reasoning is handled inside attention rather than via pixel-space pyramids.

To make this latent reasoning stable under large-resolution shifts, we equip all attention layers with 2D RoPE, using the continuous-coordinate formulation adopted by DINOv3. RoPE is parameterized by normalized coordinates in $[0, 1]$, thereby keeping the attention geometry consistent as the absolute image resolution changes. During training, we further augment these coordinates with random scaling and jitter before computing RoPE. This explicitly exposes the decoder to variations in sampling density and aspect, encouraging resolution extrapolation without changing the architecture. Together, RoPE-augmented attention and coordinate augmentation provide a minimal, modular mechanism for scalable latent decoding.

3.5 Pixel Readout: Let Pixels Be the Readout

Once latent decoding finishes "reasoning", the remaining step is purely "rendering": mapping the final latent grid to a pixel-aligned output field. This step should not become another decoder. If the readout carries significant capacity, the design collapses back to pyramidal pixel decoding in disguise, and scalability again becomes hostage to pixel-grid computation. We therefore treat pixel

rendering as an explicitly lightweight operator whose role is to translate latent predictions onto pixels, not to perform additional high-capacity inference.

In our implementation, we use a minimal readout consisting of a linear projection followed by pixel shuffle. This design keeps pixel-space computation fixed and lightweight, while allowing the latent decoder to remain the primary locus of capacity and feature fusion. We emphasize that the readout is modular: our framework does not rely on pixel shuffle specifically, and can be instantiated with other lightweight renderers when required by the task or deployment setting. In this paper, we intentionally choose the simplest sufficient readout to isolate the effect of latent decoding and keep the overall architecture minimal.

4 Main Properties

4.1 General Setup

Backbone and controlled setting. We follow the dense prediction protocol of DINOv3 [50] and freeze its ViT-7B encoder as our primary backbone. This choice is intentional for three reasons. First, when affordable, freezing a large encoder is often more iteration-efficient than unfreezing a smaller one, since training cost stays concentrated on the decoder rather than on full-backbone backpropagation. Second, it yields a cleaner controlled setting: by fixing the encoder, performance differences more directly reflect decoder design rather than encoder optimization. Third, DINOv3 is a recent and strong vision foundation model, and its reported dense prediction baselines provide a representative reference point for modern decoder evaluation.

L-DPT variants. We instantiate L-DPT as a scalable decoder family with three standard variants: small (S), base (B), and large (L). As summarized in Tab. 1, the variants differ in decoder width, depth, number of attention heads, and the attention down ratio used in the QKV projections, while sharing the same overall design. We choose these configurations to roughly match the parameter scales of common ViT variants, so that scaling trends can be interpreted under familiar capacity regimes. Unless otherwise stated, all variants use an MLP ratio of 4.

Variant	Dim	Attn Down	Layers	Heads	MLP	Params
L-DPT S	256	1	(2,2,2,2)	8	4	23M
L-DPT B	768	2	(2,2,2,2)	12	4	98M
L-DPT L	1024	1	(4,4,4,4)	16	4	269M

Table 1: L-DPT variants. We use three variants throughout the paper. Params are approximate trainable parameters of the full task head (including encoder-adapted linear projection and the readout projection), and may vary slightly across tasks.

Unified training recipe. All dense prediction tasks share the same resolution-oriented training recipe. We train with a fixed 768×768 crop, while applying strong scale augmentation before cropping: each image is randomly resized with a bicubic scale sampled from $[0.5, 2.0]$ and then cropped back to 768×768. This keeps optimization fixed while exposing the decoder to a broad range of effective sampling densities. We use the same standard photometric augmentations across

tasks, and enable RoPE-based coordinate jitter and rescaling in latent attention for all L-DPT variants (coordinate jitter = 1.2; coordinate rescaling sampled from $\times 1/8$ to $\times 8$). Together, these augmentations make resolution shifts part of the training distribution. This recipe is not incidental: the architecture provides the mechanism for scalable decoding, but the training recipe is what turns that mechanism into robust resolution generalization.

4.2 Latent Decoding Boosts Depth Across Domains

Protocol. We evaluate *relative* monocular depth estimation using the standard metrics ARel and δ_1 on NYUv2 [49], KITTI [20], ETH3D [48], and DIODE [51]. Following the Depth Anything V2 setting, we train only on synthetic data and directly predict relative depth. Our training pool combines Hypersim [45], VKITTI2 [6], Structured3D [68], TartanAir [57], BlendedMVG [62], and IRS [53]. The evaluation is performed zero-shot on the real benchmarks above. To improve training throughput on synthetic datasets, we optionally perform sequence-level deduplication using DINOv3 features, removing neighboring frames with similarity above 95%. We find that this step does not materially affect accuracy and is used only to reduce redundant samples and accelerate training.

Baselines. We compare against representative relative depth estimators that span both conventional supervised pipelines and recent large-scale pretraining. This includes MiDaS [44] and LeReS [64] as strong classical baselines, Omnidata [15] as a multi-task pretrained model, and the original DPT decoder as a canonical pyramidal pixel-decoding design. We further include recent foundation-style depth models, such as Marigold [29] and Depth Anything V2 [61], as high-capacity references. Most importantly, to isolate the effect of decoder design under a fixed backbone, we include a controlled comparison to DPT built on the same frozen DINOv3 ViT-7B encoder. In this setting, the encoder is identical and frozen, and only the decoder differs, so performance differences directly reflect the decoding paradigm rather than backbone capacity.

Results. Tab. 2 reports zero-shot relative depth estimation under a controlled setting with a frozen DINOv3 ViT-7B encoder, where only the decoder is trained. Beyond a single operating point, L-DPT forms a scalable decoder family (S/B/L) that traces a clear accuracy-efficiency frontier. As decoder capacity increases from 23M to 98M and 269M parameters, performance improves monotonically across datasets, showing that latent decoding scales predictably with capacity.

A key pattern is cross-domain robustness. While DPT (DINOv3) remains strong on canonical benchmarks (NYUv2 and KITTI), L-DPT consistently improves the hardest transfer benchmark, DIODE, across all model sizes. Notably, L-DPT S and B achieve substantially better DIODE accuracy than the much heavier DPT (DINOv3) decoder, despite using 10× and 2.3× fewer trainable parameters, respectively. At a comparable decoder budget, L-DPT L matches the best performance on NYUv2 and remains competitive on KITTI, while further improving ETH3D and DIODE. Together, these results indicate that moving decoding capacity off the pixel pyramid yields a more favorable trade-off, especially under domain shift.

Relative Depth	Parameters			NYUv2		KITTI		ETH3D		DIODE	
Methods	Enc.	Dec.	Train.	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑
MiDaS [44]	87M 🔥	19M 🔥	106M 🔥	11.1	88.5	23.6	63.0	18.4	75.2	33.2	71.5
LeReS [64]	89M 🔥	44M 🔥	133M 🔥	9.0	91.6	14.9	78.4	17.1	77.7	27.1	76.6
Omnidata [15]	108M 🔥	15M 🔥	123M 🔥	7.4	94.5	14.9	83.5	16.6	77.8	33.9	74.2
Marigold [29]	- 🔥	2.3B 🔥	2.3B 🔥	5.5	96.4	9.9	91.6	6.5	96.0	30.8	77.3
Zero-shot transfer (synthetic-only training)											
DPT (DA2)	1.1B 🔥	200M 🔥	1.3B 🔥	4.4	97.9	7.5	94.7	13.1	86.5	-	-
DPT (DINOv3)	7B ❄	230M 🔥	230M 🔥	**4.3**	98.0	**7.3**	**96.7**	5.4	97.5	25.6	82.2
L-DPT S (ours)	7B ❄	23M 🔥	23M 🔥	5.0	98.4	8.6	94.8	6.8	96.4	22.6	84.4
L-DPT B (ours)	7B ❄	98M 🔥	98M 🔥	4.9	98.4	8.2	95.5	6.0	97.0	22.2	84.9
L-DPT L (ours)	7B ❄	269M 🔥	269M 🔥	**4.3**	**98.2**	7.4	96.5	**5.2**	**97.6**	**22.0**	**84.5**
Mixed-domain supervision (w. training split)											
DPT (origin) [43]	328M 🔥	16M 🔥	344M 🔥	9.8	90.3	10.0	90.1	7.8	94.6	18.2	75.8
L-DPT S (ours)	7B ❄	23M 🔥	23M 🔥	4.9	98.7	7.7	96.0	6.2	97.0	16.7	86.3
L-DPT B (ours)	7B ❄	98M 🔥	98M 🔥	4.2	98.8	7.3	96.7	5.5	97.7	16.0	86.4
L-DPT L (ours)	7B ❄	269M 🔥	269M 🔥	**4.1**	**98.9**	**7.1**	**96.8**	**4.9**	**98.1**	**15.3**	**87.0**

Table 2: Relative depth estimation. We report zero-shot evaluated ARel (↓) and δ_1 (↑) on NYUv2, KITTI, ETH3D, and DIODE. All DINOv3-based entries share the same frozen ViT-7B encoder; only the decoder is trained, and we report encoder, decoder, and trainable parameter counts. L-DPT instantiates a scalable decoder family (S/B/L), tracing an accuracy-efficiency frontier: smaller decoders already improve cross-domain transfer (notably DIODE), and scaling decoder capacity further strengthens performance while remaining competitive on canonical benchmarks. The lower block reports *mixed-domain supervision* using dataset training splits as an upper-bound reference.

For completeness, we also report a *mixed-domain supervision* regime using training splits of the target datasets. Here, L-DPT continues to scale cleanly with decoder capacity and consistently improves across benchmarks, serving as an upper-bound sanity check that the latent decoding principle remains effective beyond the strict zero-shot setting.

Closest arbitrary-resolution baseline. InfiniDepth [65] is the closest concurrent method to our high-resolution setting, but it targets a complementary axis. InfiniDepth formulates depth as a continuous implicit field and queries arbitrary output pixels from a fixed-resolution encoder feature grid. In contrast, L-DPT addresses the decoder-side bottleneck of scaling dense prediction with input resolution: high-capacity feature integration and refinement are moved into latent decoding, followed by a lightweight pixel readout. As shown in Tab. 3, under the same DINOv3-L encoder, L-DPT-L substantially reduces inference memory and latency while matching or improving NYUv2, KITTI, and DIODE δ_1; InfiniDepth remains stronger on ETH3D in this setting. This suggests that implicit pixel querying and latent decoder-side scaling are complementary rather than mutually exclusive.

4.3 Latent Decoding Rivals Heavy Segmenter

Protocol. We evaluate semantic segmentation on COCO-Stuff 164k [8] and PASCAL VOC 2012 [16], reporting mIoU under a simple single-scale setting

Method	Encoder	S.S.	Mem.	Lat.	NYUv2	KITTI	ETH3D	DIODE
InfiniDepth [65]	DINOv3-L	RANSAC	10.2G	3.38s	97.8	95.8	**98.6**	85.5
L-DPT-L	DINOv3-L	IS	**5.7G**	**1.74s**	98.0	95.9	97.4	86.1
L-DPT-L	DINOv3-7B	IS	7.6G	1.99s	**98.9**	**96.8**	98.1	**87.0**

Table 3: Closest arbitrary-resolution depth comparison. We compare InfiniDepth and L-DPT under matched splits, masks, and δ_1. InfiniDepth uses its recommended RANSAC scale/shift estimation, while L-DPT uses the standard image-wise scale/shift estimation (IS). Memory and latency are measured on one NVIDIA H800.

Sem. Seg. (mIoU ↑)	Parameters			COCO-Stuff 164k		PASCAL VOC 2012	
Methods	Enc.	Dec.	Train.	Simple	TTA	Simple	TTA
Previous Best	- 🔥	- 🔥	1.5B/0.5B 🔥	53.7	53.7	-	90.0
DINOv3 Segmentor	7B ❄	927M 🔥	927M 🔥	53.8	54.0	90.1	90.4
L-DPT S (ours)	7B ❄	23M 🔥	23M 🔥	53.5	53.6	93.4	93.6
L-DPT B (ours)	7B ❄	98M 🔥	98M 🔥	53.8	54.0	93.8	93.8
L-DPT L (ours)	7B ❄	269M 🔥	269M 🔥	**54.0**	**54.2**	**94.0**	**94.1**

Table 4: Semantic segmentation. We report mIoU on COCO-Stuff 164k [8] and PASCAL VOC 2012 [16] under simple evaluation and lightweight TTA (horizontal flip), together with encoder/decoder/trainable parameter counts. The "DINOv3" baseline in the table refers to the ViT-Adapter + Mask2Former configuration reported in DINOv3, using the same frozen ViT-7B encoder with a much heavier pixel-space decoder. L-DPT instantiates a scalable decoder family (S/B/L), tracing a clear accuracy–efficiency frontier: mid-sized latent decoders already match the heavy baseline, larger ones surpass it, and all variants significantly improve VOC performance.

and a lightweight TTA setting. We follow the training protocol of DINOv3 for segmentation. On COCO-Stuff, we train the segmentation decoder from scratch on the target dataset. On VOC 2012, we fine-tune the COCO-Stuff pretrained decoder due to the limited size of the VOC dataset.

Baselines. We compare against the DINOv3 semantic segmentation baseline, combining Mask2Former [11] and ViT-Adapter [10], which represents a strong pyramidal pixel-decoding pipeline with a heavy task decoder. For additional context, we also include the "previous best" reference used in DINOv3: Mask2Former on COCO-Stuff, and *Rethinking Pre-training and Self-training* [69] on VOC 2012. As in our other experiments, we report encoder, decoder, and trainable parameter counts.

Results. Tab. 4 reports semantic segmentation results on COCO-Stuff 164k and VOC 2012 under a controlled setting with a frozen DINOv3 ViT-7B encoder, where only the decoder is trained. The DINOv3 baseline corresponds to the ViT-Adapter combined with the Mask2Former architecture, as reported in DINOv3, which uses a substantially heavier pixel-space decoder. In contrast, L-DPT forms a scalable latent-decoder family (S/B/L) that traces a clear accuracy-efficiency frontier: performance improves monotonically as decoder capacity increases. On COCO-Stuff, this frontier is particularly clean. L-DPT S remains highly competitive despite a dramatically smaller decoder, L-DPT B matches the DINOv3

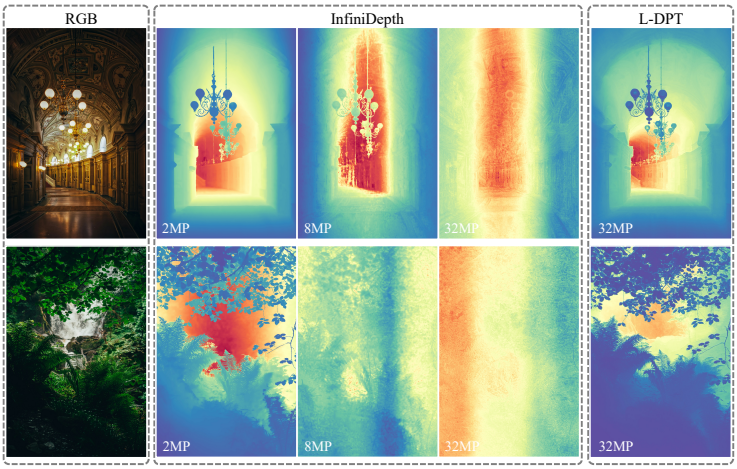

Fig. 5: Qualitative comparison with InfiniDepth. We visualize high-resolution relative-depth predictions across increasing input resolutions. InfiniDepth is evaluated at 2MP, 8MP, and 32MP, and compared with L-DPT at high resolution. The visualization complements the quantitative comparison in Tab. 3.

Method	ARel ↓	δ_1 ↑
Fixed-Schedule Fusion	9.67	91.77
Shared Selection	9.24	92.06
Stage-wise Adaptive Selection	**8.89**	**92.76**

Table 5: Injection mechanism. Ablation on Hypersim relative depth estimation. We train the model solely on the train split and report δ_1 (↑) and ARel (↓). Stage-wise adaptive selection outperforms progressive stage-bound injection and a shared selector, showing that repeatedly re-selecting encoder information across latent stages yields better depth prediction.

baseline with nearly 10× fewer decoder parameters, and L-DPT L further improves over it while still using a substantially smaller decoder. On VOC 2012, the gain is even more pronounced: all L-DPT variants significantly outperform the DINOv3 baseline, including the smallest 23M model. This pattern suggests that the improvement is architectural rather than attributable to decoder size alone. Strong segmentation does not require a heavy progressive pixel-space decoder when capacity and feature fusion are concentrated in latent decoding.

4.4 Adaptive Selection Boosts Accuracy

Latent decoding changes how feature fusion can be done. Classical pyramidal decoders enforce a fixed, progressive schedule, binding specific encoder stages to specific decoding stages. Once decoding is moved onto a single latent grid, this constraint disappears: multi-level encoder outputs can be treated as a shared feature pool, and each latent stage gains the privilege to *re-select* what it needs.

Tab. 5 isolates this choice on the Hypersim [45] validation set, with relative depth estimation held fixed across all other settings. Replacing progressive, stage-bound injection with stage-wise adaptive selection consistently improves

depth accuracy. Moreover, sharing a single selector across all stages underperforms stage-wise selection, indicating that re-compressing the feature pool is not redundant: different stages benefit from different mixtures of encoder semantics. Overall, adaptive selection turns feature fusion from a hand-designed schedule into a learned decision, yielding a tangible gain with minimal added complexity.

5 Related Work

Dense decoding. Modern dense prediction is realized through pixel-space dense decoders. A common pattern is pyramidal decoding, where multi-level features are progressively fused, upsampled, and refined to produce dense outputs [35,43]. This paradigm underlies depth estimation [37,58,63,66] and semantic segmentation [11,59], and remains the default interface for turning visual backbones into dense predictors. Recent systems [5,34,52,54,55,56] continue to adopt this interface. Works such as Depth Anything [33,60,61] and DINO [9,39,50] primarily strengthen the backbone, training recipe, or dense feature quality, while still relying on DPT-style decoders or downstream dense heads. This default inherits a shared limitation: dense prediction is realized through progressive computation on pixel-aligned feature maps, so decoder cost grows with the output grid. This motivates revisiting dense prediction from the opposite direction, by moving high-capacity decoding off the pixel grid and into latent space.

Dense prediction in latent space. A complementary line of work suggests that high-fidelity dense outputs need not be formed through pixel-space reconstruction. In generative modeling, latent-space methods such as VAE [31], latent diffusion models [46], and diffusion transformers [40] demonstrate that complex visual structure can be synthesized primarily from expressive latent representations, with pixel-space decoding attached as a lightweight final stage [67]. More directly related to vision analysis, recent methods [17,21,23,32,46] encode images into latent space and decode for dense prediction. Monocular depth estimation methods [28,29] further show that latent generative priors can be repurposed for dense prediction, indicating that the burden of modeling need not remain on a dense pixel pyramid. These works motivate our key intuition that high-capacity computation can live in latent space; however, rather than relying on generative sampling or diffusion-style generation, we revisit dense prediction itself as a deterministic latent decoding problem.

Resolution adaptation and extrapolation. Scaling dense predictors to larger grids is handled by resizing, tiling, sliding-window inference, or multi-scale evaluation [38], while more principled works study transformer extrapolation beyond training resolution [2,14]. Prior work improves the use of denser token grids through positional and attention-geometry designs, including RoPE [24,50], conditional positional encodings [12], directed attention [18], and attention biases [19,36]. Closest to our high-resolution depth setting, InfiniDepth [65] queries a continuous implicit depth field from a fixed-resolution encoder feature grid. L-DPT is complementary: it addresses the decoder-side bottleneck of input-resolution scaling by moving high-capacity feature integration into latent decod-

ing and keeping pixel rendering lightweight, making the same principle applicable beyond depth to pixel-aligned dense fields such as semantic segmentation.

6 Conclusion and Discussion

Pyramidal pixel decoding is an effective default for dense prediction, but it does not scale cleanly. Our pilot study exposes two walls: an efficiency wall, where memory and latency escalate sharply with resolution, and a capability wall, where more pixels stop helping and can even blur local geometry. We propose a different design principle: put capacity in latents and let pixels be the readout. L-DPT instantiates this principle with latent-space attention for high-capacity reasoning and a minimal pixel readout for rendering. Across relative depth estimation and semantic segmentation, L-DPT forms a scalable decoder family with a clear accuracy–efficiency frontier, and enables stable inference far beyond the training grid, up to 100 megapixels in our tests.

We also discuss *limitations and broader impacts as follows*:

- **Sparse decoding as a unifying direction.** L-DPT moves dense decoding off the pixel grid; a natural next step is to allocate computation even more selectively via sparse queries and adaptive readout, potentially unifying diverse dense fields under a common decoding interface. This direction could further reduce the need for resolution-dependent decoding and make high-resolution dense prediction more accessible in real systems.
- **Training at native resolution.** Our results show that resolution extrapolation can emerge from a fixed-resolution training regime with strong scale and coordinate augmentation, but fully native ultra-high-resolution training remains largely unexplored. Resolution-aware batching and readout budgets offer a path to make high-resolution behavior a first-class training objective, strengthening robustness for professional imaging, robotics, and mixed-reality workloads where resizing and tiling are undesirable.
- **Connections to latent generative modeling.** L-DPT shares a key premise with latent diffusion: high-fidelity structure can be produced by high-capacity latent modeling, with pixels as a rendering surface. This suggests opportunities to incorporate iterative latent refinement or diffusion/flow-style objectives for uncertainty-aware or multi-modal dense prediction, while keeping the decoder scalable and the pixel readout lightweight.

More broadly, we view latent decoding not as a specific network, but as a scalable paradigm shift for dense prediction. Pixels scale; dense decoding should not.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62422606), Hong Kong Research Grant Council General Research Fund (No. 17213925), and Shenzhen Loop Area Institute Grant (No. FPF10120250006). We would also like to thank Vladlen Koltun, Stephan R. Richter, Lars Mescheder, Aleksei Bochkovskii, Wei Dong, and Boyang Zheng for fruitful discussions.

References

1. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., Shulman, E.: ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In: NeurIPSW (2021)
2. Beyer, L., Izmailov, P., Kolesnikov, A., Caron, M., Kornblith, S., Zhai, X., Minderer, M., Tschannen, M., Alabdulmohsin, I., Pavetic, F.: Flexivit: One model for all patch sizes. In: CVPR (2023)
3. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023)
4. Birkl, R., Wofk, D., Müller, M.: Midas v3. 1–a model zoo for robust monocular relative depth estimation. arXiv preprint arXiv:2307.14460 (2023)
5. Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S.R., Koltun, V.: Depth pro: Sharp monocular metric depth in less than a second. In: ICLR (2025)
6. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
7. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
8. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
9. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
10. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. In: ICLR (2023)
11. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
12. Chu, X., Tian, Z., Zhang, B., Wang, X., Shen, C.: Conditional positional encodings for vision transformers. In: ICLR (2023)
13. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
14. Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A., Puigcerver, J., Geirhos, R., Alabdulmohsin, I.M., et al.: Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. NeurIPS (2023)
15. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV (2021)
16. Everingham, M., Winn, J.: The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep (2011)
17. Fu, X., Yin, W., Hu, M., Wang, K., Ma, Y., Tan, P., Shen, S., Lin, D., Long, X.: Geowizard: Unleashing the diffusion priors for 3D geometry estimation from a single image. arXiv:2403.12013 (2024), https://arxiv.org/abs/2403.12013
18. Fuller, A., Kyrollos, D., Yassin, Y., Green, J.: Lookhere: Vision transformers with directed attention generalize and extrapolate. NeurIPS (2024)
19. Fuller, A., Millard, K., Green, J.: Croma: Remote sensing representations with contrastive radar-optical masked autoencoders. NeurIPS (2023)
20. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. IJRR (2013)

21. Gui, M., Schusterbauer, J., Prestel, U., Ma, P., Kotovenko, D., Grebenkova, O., Baumann, S.A., Hu, V.T., Ommer, B.: Depthfm: Fast generative monocular depth estimation with flow matching. In: AAAI (2025)
22. Guo, G., Guo, Y., Yu, X., Li, W., Wang, Y., Gao, S.: Segment any-quality images with generative latent space enhancement. In: CVPR (2025)
23. He, J., Li, H., Yin, W., Liang, Y., Li, L., Zhou, K., Zhang, H., Liu, B., Chen, Y.C.: Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In: ICLR (2025)
24. Heo, B., Park, S., Han, D., Yun, S.: Rotary position embedding for vision transformer. In: ECCV. Springer (2024)
25. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
26. Hu, M., Yin, W., Zhang, C., Cai, Z., Long, X., Chen, H., Wang, K., Yu, G., Shen, C., Shen, S.: Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. TPAMI (2024)
27. Jain, J., Li, J., Chiu, M.T., Hassani, A., Orlov, N., Shi, H.: Oneformer: One transformer to rule universal image segmentation. In: CVPR (2023)
28. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: CVPR (2024)
29. Ke, B., Qu, K., Wang, T., Metzger, N., Huang, S., Li, B., Obukhov, A., Schindler, K.: Marigold: Affordable adaptation of diffusion-based image generators for image analysis. In: CVPR (2024)
30. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
31. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
32. Lee, H.Y., Tseng, H.Y., Lee, H.Y., Yang, M.H.: Exploiting diffusion prior for generalizable dense prediction. In: CVPR. pp. 7861–7871 (2024), https://arxiv.org/abs/2311.18832
33. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Zhao, Y., Peng, S., Guo, H., Zhou, X., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. In: ICLR (2026)
34. Lin, H., Peng, S., Chen, J., Peng, S., Sun, J., Liu, M., Bao, H., Feng, J., Zhou, X., Kang, B.: Prompting depth anything for 4k resolution accurate metric depth estimation. In: CVPR (2025)
35. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
36. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: CVPR (2022)
37. Liu, Z., Cheng, K.L., Wang, Q., Wang, S., Ouyang, H., Tan, B., Zhu, K., Shen, Y., Chen, Q., Luo, P.: Depthlab: From partial to complete. arXiv preprint arXiv:2412.18153 (2024)
38. Miangoleh, S.M.H., Dille, S., Mai, L., Paris, S., Aksoy, Y.: Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In: CVPR. pp. 9685–9694 (2021)
39. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve,

G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. TMLR (2024)
40. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
41. Piccinelli, L., Sakaridis, C., Yang, Y.H., Segu, M., Li, S., Abbeloos, W., Van Gool, L.: Unidepthv2: Universal monocular metric depth estimation made simpler. TPAMI (2025)
42. Piccinelli, L., Yang, Y.H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: Unidepth: Universal monocular metric depth estimation. In: CVPR (2024)
43. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV (2021)
44. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. TPAMI (2020)
45. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: ICCV (2021)
46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
47. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
48. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: CVPR (2017)
49. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
50. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. arXiv preprint arXiv:2508.10104 (2025)
51. Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F.Z., Daniele, A.F., Mostajabi, M., Basart, S., Walter, M.R., et al.: Diode: A dense indoor and outdoor depth dataset. arXiv preprint arXiv:1908.00463 (2019)
52. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025)
53. Wang, Q., Zheng, S., Yan, Q., Deng, F., Zhao, K., Chu, X.: Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. arXiv preprint arXiv:1912.09678 (2019)
54. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: CVPR (2025)
55. Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J.: Moge-2: Accurate monocular geometry with metric scale and sharp details. arXiv preprint arXiv:2507.02546 (2025)
56. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR (2024)
57. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. In: IROS (2020)
58. Wen, B., Trepte, M., Aribido, J., Kautz, J., Gallo, O., Birchfield, S.: Foundationstereo: Zero-shot stereo matching. In: CVPR (2025)

59. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: NeurIPS (2021)
60. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: CVPR (2024)
61. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. In: NeurIPS (2024)
62. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: CVPR (2020)
63. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: ICCV (2023)
64. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3d scene shape from a single image. In: CVPR (2021)
65. Yu, H., Lin, H., Wang, J., Li, J., Wang, Y., Zhang, X., et al.: Infinidepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. In: CVPR (2026)
66. Yuan, W., Gu, X., Dai, Z., Zhu, S., Tan, P.: Neural window fully-connected crfs for monocular depth estimation. In: CVPR (2022)
67. Zavadski, D., Kalšan, D., Rother, C.: Primedepth: Efficient monocular depth estimation with a stable diffusion preimage. In: ACCV (2024)
68. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In: ECCV (2020)
69. Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.V.: Rethinking pre-training and self-training. In: NeurIPS (2020)