Scaling Properties of Diffusion Models for Perceptual Tasks

Zeeshan Patel
Rahul Ravishankar
Jathushan Rajasegaran
Jitendra Malik
Alexei (Alyosha) Efros, Ed.

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-38
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-38.html

May 1, 2025

Scaling Properties of Diffusion Models for Perceptual Tasks

by Zeeshan Patel

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Signé par :
Alexei Efros
D12772948BC0486...

Professor Alexei Efros
Research Advisor

4/18/2025

(Date)

* * * * * * *

Signed by:
Jitendra Malik
16D473A00244427...

Professor Jitendra Malik
Second Reader

4/21/2025

(Date)

Scaling Properties of Diffusion Models For Perceptual Tasks

Zeeshan Patel*, Rahul Ravishankar*, Jathushan Rajasegaran, Jitendra Malik
UC Berkeley
zeeshanp@berkeley.edu

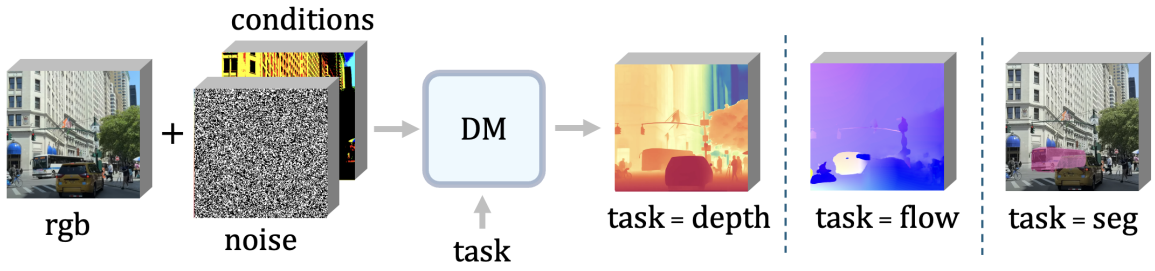


Figure 1. **A Unified Framework:** We fine-tune a pre-trained Diffusion Model (DM) for visual perception tasks. We take a RGB image, and a conditional image (i.e. next video frame, occlusion mask, etc.), along with the noised image of the ground truth prediction. Our model generates predictions for visual tasks such as depth estimation, optical flow prediction, and amodal segmentation, based on the conditional task embedding. We train a generalist model that can perform all three tasks with exceptional performance.

Abstract

In this paper, we argue that iterative computation with diffusion models offers a powerful paradigm for not only generation but also visual perception tasks. We unify tasks such as depth estimation, optical flow, and amodal segmentation under the framework of image-to-image translation, and show how diffusion models benefit from scaling training and test-time compute for these perceptual tasks. Through a careful analysis of these scaling properties, we formulate compute-optimal training and inference recipes to scale diffusion models for visual perception tasks. Our models achieve competitive performance to state-of-the-art methods using significantly less data and compute. We release code and models at scaling-diffusion-perception.github.io.

1. Introduction

Diffusion models have emerged as a powerful tool for generating images and videos with excellent scaling behaviors. In this paper, we present a unified framework to perform a variety of perceptual tasks — depth estimation, optical flow estimation, and amodal segmentation — with a single diffusion model, as illustrated in Fig. 1.

Previous works such as Marigold [20], FlowDiffuser [28], and pix2gestalt [31] demonstrate the potential of repurposing image diffusion models for various inverse vision tasks individually. Building on these prior works, we systematically explore the benefits of scaling across the axes of pre-training, fine-tuning, and test-time compute for image diffusion models. We establish scaling power laws for depth estimation and display their transferability to other perceptual tasks. Using these scaling laws, we formulate compute-optimal recipes for diffusion training and inference for multiple downstream perceptual tasks. Our work shows that a limited compute budget can be efficiently allocated for strong downstream performance.

Recent works in other fields have also focused on scaling test-time compute to enhance the capabilities of modern LLMs [7, 30]. We show that **increasing test-time compute compensates for the 3-4 orders of magnitude reduced pre-training budgets we use.** This redistribution allows us to achieve competitive results while using significantly less data and overall training compute, highlighting the tradeoff between training and test-time compute.

We scale test-time compute by exploiting the iterative and stochastic nature of diffusion. By increasing the number of denoising steps, allocating more compute to early denoising steps, and ensembling multiple denoised predictions, we consistently achieve higher accuracy on these perceptual tasks. Our results provide evidence on the benefits of scaling test-time compute for inverse vision problems under constrained compute budgets, bringing a new perspective to the conventional paradigm of training-centric scaling for generative models.

*Equal Contribution. Accepted to CVPR 2025.

2. Related Work

2.1. Generative Modeling

Generative modeling has been studied under various methods, including VAEs [21], GANs [14], Normalizing Flows [35], Autoregressive models [48], and Diffusion models [16, 43]. Denoising Diffusion Probabilistic Models (DDPMs) [16] have shown impressive scaling behaviors for many image and video generation models. Notable examples include Latent Diffusion Models [36], which enhance efficiency by operating in a compressed latent space, Imagen [38], which generates samples in pixel space with increasing resolution, and Consistency Models [45], which accelerate sampling while maintaining generation quality. Flow Matching [25, 27] employs training objectives inspired by optimal transport to model continuous vector fields that map data to target distributions, eliminating the discrete formulation of diffusion models.

2.2. Scaling Diffusion Models

Diffusion modeling has shown impressive scaling behaviors in terms of data, model size, and compute. Latent Diffusion Models [36] first showed that training with large-scale web datasets can achieve high quality image generation results with a U-Net backbone. DiT [32] explored scaling diffusion models with the transformer architecture, presenting desirable scaling properties for class-conditional image generation. Later, *Li et al.* [24] studied alignment scaling laws of text-to-image diffusion models. Recently, *Fei et al.* [11] trained mixture-of-experts DiT models up to 16B parameters, achieving high-quality image generation results. Upcycling can also help scale transformer models. *Komatsuzaki et al.* [22] used upcycling to convert a dense transformer-based language model to a mixture-of-experts model without pre-training from scratch. Similarly, EC-DiT [46] explores how to exploit heterogeneous compute allocation in mixture-of-experts training for DiT models through expert-choice routing and learning to adaptively optimize the compute allocated to specific text-image data samples.

2.3. Diffusion Models for Perception Tasks

Diffusion models have also been used for various downstream visual tasks such as depth estimation [8, 19, 39, 40, 54]. More recently, Marigold [20] and GeoWizard [13] displayed impressive results by repurposing pre-trained diffusion models for monocular depth estimation. Diffusion models with few modifications are used for semantic segmentation for categorical distributions [1–3, 17, 47, 52], instance segmentation [15], and panoptic segmentation [6]. Diffusion models are also used for optical flow [28, 40] and 3D understanding [18, 26, 33, 49, 50].

3. Generative Pre-Training

We first explore how to efficiently scale diffusion model pre-training. We pre-train diffusion models for class-conditional image generation using a diffusion transformer (DiT) backbone and follow the original model training recipe [32].

Starting with a target RGB image $I \in \mathbb{R}^{u \times u \times 3}$, where the resolution of the image is $u \times u$, our pretrained, frozen Stable Diffusion variational autoencoder [36] compresses the target to a latent $z_0 \in \mathbb{R}^{w \times w \times 4}$, where $w = u/8$. Gaussian noise is added at sampled time steps to obtain a noisy target latent. Noisy samples are generated as:

$$z_t = \sqrt{\alpha_t} \cdot z_0 + \sqrt{1 - \alpha_t} \cdot \epsilon_t \tag{1}$$

for timestep t. The noise is distributed as $\epsilon \sim \mathcal{N}(0, I)$, $t \sim \text{Uniform}(T)$, with $T = 1000$ and $\alpha_t := \prod_{s=1}^{t}(1 - \beta_s)$, with $\{\beta_1, \ldots, \beta_T\}$ as the variance schedule of a process. In the denoising process, the class-conditional DiT $f_\theta(\cdot)$, parameterized by learned parameters θ, gradually removes noise from z_t to obtain z_{t-1}. The parameters θ are updated by noising z_0 with sampled noise ϵ at a random timestep t, computing the noise estimate, and optimizing the mean squared loss between the generated noise and estimated noise in an n batch size sample. We formally represent this as the following minimization problem:

$$\theta^* = \arg\min_\theta \mathbb{E}_{t \sim U(\{1,\ldots,T\}), \epsilon \sim \mathcal{N}(0,I)} \left[\|\epsilon - f_\theta(z_t; t)\|^2 \right], \tag{2}$$

where θ^* are the DiT learned parameters and $f_\theta(z_t; t)$ is the DiT noise prediction for sample i.

3.1. Model Size

We pre-train six different dense DiT models as in Table 1, increasing model size by varying the number of layers and hidden dimension size. We use ImageNet-1K [37] as our pre-training dataset and train all models for 400k iterations with a fixed learning rate of $1e$-4 and a batch size of 256. Fig. 2 shows that larger models converge to lower loss with a clear power law scaling behavior. We show the train loss as a function of compute (in MACs), and our predictions indicate a power law relationship of $L(C) = 0.23 \times C^{-0.0098}$. Our pre-training experiments display the ease of scaling DiT with a small training dataset, which translates directly to efficiently scaling downstream model performance.

3.2. Mixture of Experts

We also pre-train Sparse Mixture of Experts (MoE) models [41], following the S/2 and L/2 model configurations in [10]. We use three different MoE configurations listed in Table 2, scaling the total parameter count by increasing hidden size, number of experts, layers, and attention heads. Each MoE block activates the top-2 experts per token and

Model	Params	Dimension	Heads	Layers
a1	14.8M	256	16	12
a2	77.2M	512	16	16
a3	215M	768	16	20
a4	458M	1024	16	24
a5	1.2B	1536	16	28
a6	1.9B	1792	16	32

Table 1. **Dense DiT Models:** We scale dense DiT model size by increasing hidden dimension and number of layers linearly while keeping number of heads constant following [34, 53].

Model	Active / Total	Dim	Heads	Layers
S/2-8E2A	71M / 199M	384	6	12
S/2-16E2A	71M / 369M	384	6	12
L/2-8E2A	1.0B / 2.8B	1024	16	24

Table 2. **MoE DiT Models:** We scale the MoE DiT models by increasing dimension size, number attention heads, layers, and experts following [10].

has a shared expert that is used by all tokens. To alleviate issues with expert balance, we use the proposed expert balance loss function from [10] which distributes the load across experts more efficiently. Sparse MoE pre-training allows for a higher parameter count while increasing throughput, making it more compute efficient than training a dense DiT model of the same size. We train our DiT-MoE models with the same training recipe as the dense DiT model using ImageNet-1K. Our approach enables training DiT-MoE models to increase model capacity without increasing compute usage by another order of magnitude, which would be required to train dense models of similar sizes.

4. Fine-Tuning for Perceptual tasks

In this section, we explore how to scale the fine-tuning of the pre-trained DiT models to maximize performance on downstream perception tasks. During fine-tuning, we utilize the image-to-image diffusion process from [20] and [4] as our training recipe. We pose all our visual tasks as conditional denoising diffusion generation. Give an RGB image $I \in \mathbb{R}^{u \times u \times 3}$ and its pair ground truth image $D \in \mathbb{R}^{u \times u \times 3}$, we first project them to the latent space, $i_0 \in \mathbb{R}^{w \times w \times 4}$ and $d_0 \in \mathbb{R}^{w \times w \times 4}$, respectively. We only add noise to the ground truth latent to obtain d_t and concatenate it with the RGB latent which results in a tensor $z_t = \{i_0, d_t\}$. The first convolutional layer of the DiT model is modified to match the doubled number of input channels, and its values are reduced by half to make sure the predictions are the same if the inputs are just RGB images [20]. Finally, we perform diffusion training by denoising the ground truth image. We ablate several fine-tuning compute scaling techniques on the monocular depth estimation task and report Absolute Relative error and Delta1 error. We transfer the best configurations from the depth estimation ablation study to fine-tune for other visual perception tasks.

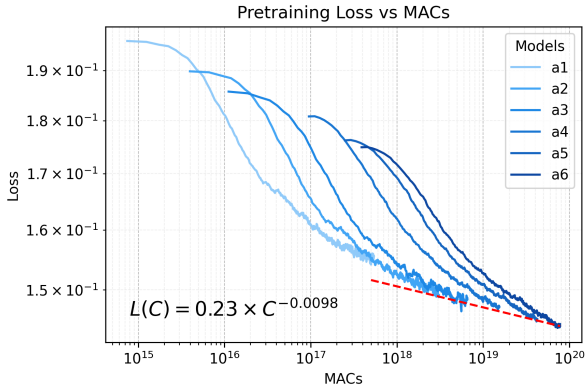


Figure 2. **Scaling at Model Size:** For generative pre-training of DiT models, we observe clear power law scaling behavior as we increase the model size.

4.1. Effect of Model Size

We fine-tune the pre-trained a1-a6 dense models on the depth estimation task to study the effect of model size. We scale model size as shown in as described in Section 3.1. Fig. 3 shows that larger dense DiT models predictably converge to a lower fine-tuning loss, presenting a clear power law scaling behavior. We plot the train loss and validation metrics as a function of compute (in MACs). Our fine-tuned model predictions show a power law relationship in both depth Absolute Relative error and Delta1 error. The power law relationship shows that scaling fine-tuning compute by increasing model size can provide significant gains on downstream tasks.

4.2. Effect of Pre-training Compute

We investigate the behavior of fine-tuning as we scale the number of pre-training steps for the DiT backbone. We train four models with the a4 configuration, only varying the number of pre-training steps. We then fine-tune these four models on the same depth estimation dataset. Fig. 4 displays the power law scaling behavior of the validation metrics for depth estimation as we increase DiT pre-training steps. Our experiments show that having stronger pre-trained representations has profound impact on model performance when scaling fine-tuning compute.

4.3. Effect of Image Resolution

The sequence length of each image also affects the total compute spent during training. For each forward pass, we can scale the amount of compute used by simply increasing the resolution of the image, which will increase the number

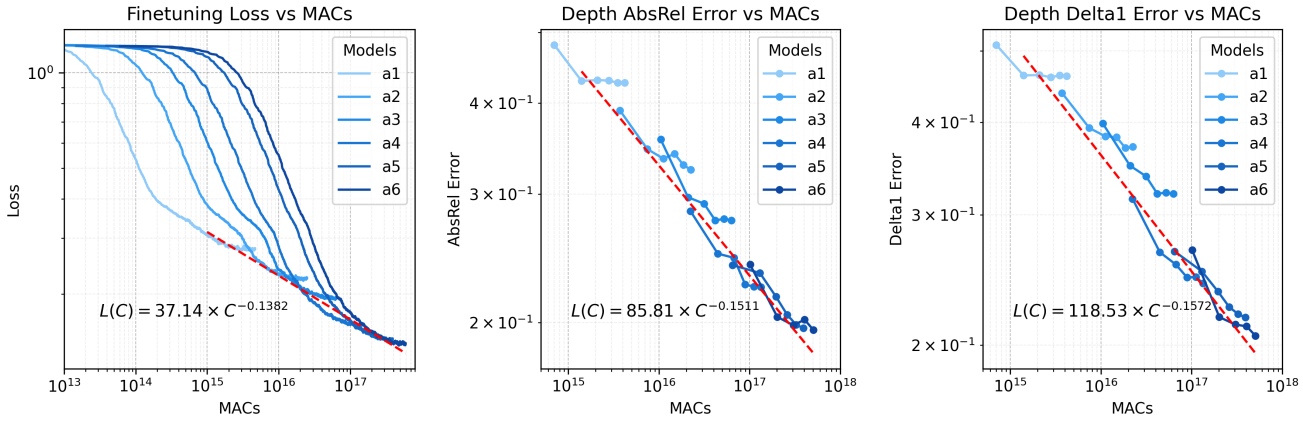


Figure 3. **Effect of Model Size:** We fine-tune a1-a6 models on the Hypersim dataset for 30K iterations with an exponential decay learning rate schedule from $3e$-5 to $3e$-7. We observe a strong correlation between the scaling laws of the fine-tuning loss and validation metrics.

of tokens in the image embedding. By increasing the number of tokens, we can increase the amount of information the model can learn from at training time to build stronger internal representations, which can in turn improve downstream performance. We use dense DiT-XL models with resolutions of 256×256 and 512×512 from [32] and we pre-train DiT-MoE L/2-8E2A models with 256×256 and 512×512 resolutions following the recipe in [10]. We then fine-tune each of these models with the corresponding resolution for the depth estimation task.

Fig. 5 displays that increasing image resolution to scale fine-tuning compute provides significant gains on downstream depth estimation performance. In our case, we effectively use $4\times$ the amount of tokens to represent each image, which also scales the total compute utilization by $4\times$.

4.4. Effect of Upcycling

Sparse MoE models are efficient options for increasing model capacity, but pre-training MoE models from scratch can be expensive. One way to alleviate this issue is Sparse MoE Upcycling [23]. Upcycling converts dense transformer models to MoE models by copying the MLP layer in each transformer block E times, where E is the number of experts, and adding a learnable router module to send each token to the top-k selected experts. The outputs of the selected experts are combined in a weighted sum at the end of each MoE block. We upcycle various dense DiT models after they are fine-tuned for depth estimation and then continue fine-tuning the upcycled model. Fig. 6 displays the scaling laws for upcycling, providing an average improvement of 5.3% on Absolute Relative Error and 8.6% on Delta1 error. Our results show that upcycling is an inexpensive and effective way to scale fine-tuning compute and significantly improves downstream performance.

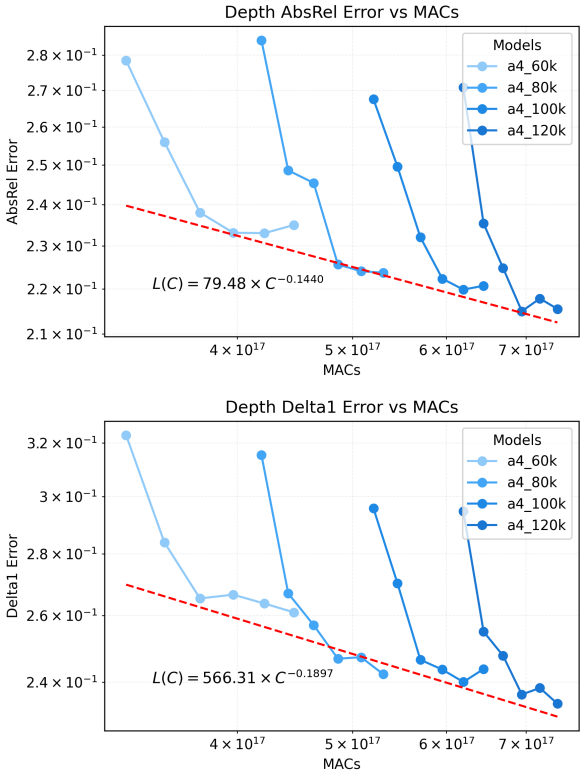


Figure 4. **Effect of Scaling Model Pre-training Compute on Depth Estimation:** (a) Depth Absolute Relative Error vs. MACs. (b) Depth Delta1 Error vs. MACs. We pre-train four a4 models with 60K, 80K, 100K, and 120K steps. These models are then fine-tuned for 30K steps on the Hypersim depth estimation dataset. We observe a clear power law as we increase the DiT pre-training compute across depth estimation validation metrics.

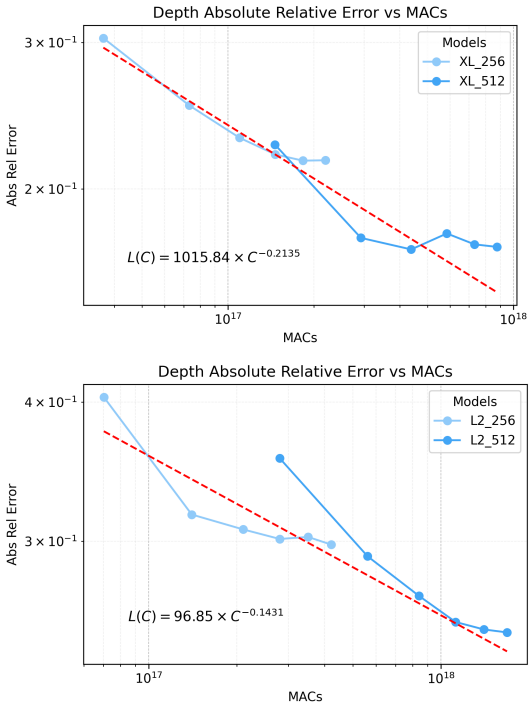


Figure 5. **Effect of Image Resolution.** We fine-tune DiT-XL and DiT-MoE L/2 models with resolutions of 256×256 and 512×512. We observe a power law when increasing image resolution during training. By scaling the number of tokens per image by $4\times$, we achieve strong performance on Depth Absolute Error, displaying the effect of increasing total dataset tokens for dense visual perception tasks such as depth estimation.

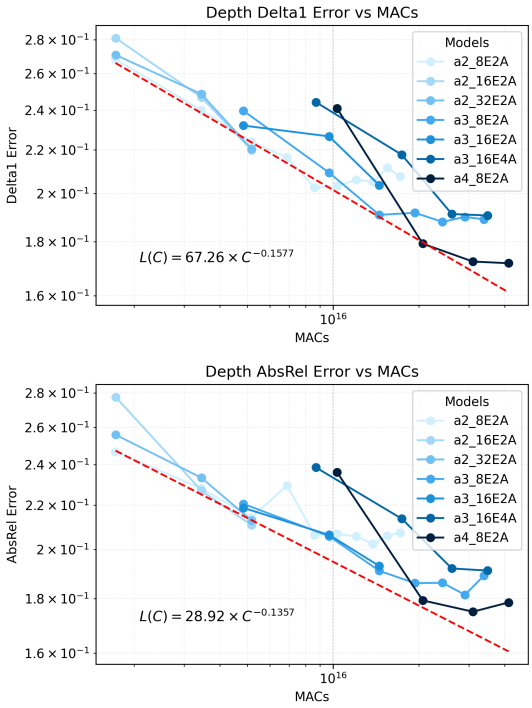


Figure 6. **Effect of Upcycling.** We upcycle a2, a3, and a4 models fine-tuned for depth estimation with a varying number of total/active model experts. We continue fine-tuning each upcycled model for 15K iterations on the Hypersim depth estimation dataset. We observe a clear scaling law in the validation metrics as we increase fine-tuning compute with upcycling. The upcycled models can also achieve equivalent or superior performance to our dense a5 and a6 checkpoints, each of which utilize more compute during pre-training and fine-tuning. Increasing the total model experts and total active experts can also improve the downstream performance.

5. Scaling Test-Time Compute

Scaling test-time compute has been explored for autoregressive Large Language Models (LLMs) to improve performance on long-horizon reasoning tasks [5, 9, 30, 42]. In this section, we show how to reliably improve diffusion model performance for perceptual tasks by scaling test-time compute. We summarize our approach in Fig. 7. We use the Stable-Diffusion VAE to encode the input image into latent space [36]. Then, we sample a target noise latent from a standard Gaussian distribution, which is iteratively denoised with DDIM [44] to generate the downstream prediction.

5.1. Effect of Scaling Inference Steps

The most natural way of scaling diffusion inference is by increasing denoising steps. Since the model is trained to denoise the input at various timesteps, we can scale the number of diffusion denoising steps at test-time to produce finer, more accurate predictions. This coarse-to-fine denoising paradigm is also reflected in the generative case, and we can take advantage of it for the discriminative case by increasing the number of denoising steps. In Fig. 8a, 8d, we observe that increasing the total test-time compute by simply increasing the number of diffusion sampling steps provides substantial gains in depth estimation performance. This shows that scaling the number of sampling steps is crucial to maximize the downstream performance of diffusion models trained for discriminative tasks.

5.2. Effect of Test-Time Ensembling

We also explore scaling inference compute with test-time ensembling. We exploit the fact that denoising different noise latents will generate different downstream predictions. In test-time ensembling, we compute N forward passes for each input sample and reduce the outputs through one of two methods. The first technique is naive ensembling where we use the pixel-wise median across

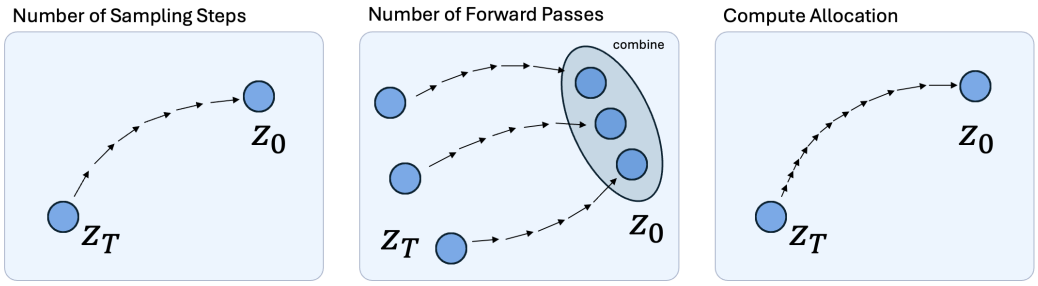


Figure 7. **Inference Scaling:** Diffusion models by design allow efficient scaling of test-time compute. First, we can simply increase the number of denoising steps to increase the compute spent at inference. Since we are estimating deterministic outputs, we can then initialize multiple noise latents and ensemble the predictions to get a better estimation. Finally, we can also reallocate our test-time compute budget for low and high frequency denoising by modifying the noise variance schedule.

all outputs. The second technique presented in Marigold [20] is median compilation, where we collect predictions $\{\hat{d}_1, \ldots, \hat{d}_N\}$ that are affine-invariant, estimate scale and shift parameters $\hat{s}_i$ and $\hat{t}_i$, and minimize the distances between each pair of scaled and shifted predictions $(\hat{d}'_i, \hat{d}'_j)$ where $\hat{d}' = \hat{d} \times \hat{s} + \hat{t}$. For each step, we take the pixel-wise median $m(x, y) = \text{median}(\hat{d'_1}(x, y), \ldots, \hat{d'_N}(x, y))$ to compute the merged depth m. Since it requires no ground truth, we scale ensembling by increasing N to utilize more test-time compute. Figs. 8b, 8e display the power law scaling behavior of test-time ensembling.

5.3. Effect of Noise Variance Schedule

We can also scale test-time compute by increasing compute usage at different points of the denoising process. In diffusion noise schedulers, we can define a schedule for the variance of the Gaussian noise applied to the image over the total diffusion timesteps T. Tuning the noise variance schedule allows for reorganizing compute by allocating more compute to denoising steps earlier or later in the noise schedule. We experiment with three different noise level settings for DDIM: linear, scaled linear, and cosine. Cosine scheduling from [29] linearly declines from the middle of the corruption process, ensuring the image is not corrupted too quickly as in linear schedules. Fig. 8c, 8f shows that the cosine schedule outperforms linear schedules for DDIM on depth estimation under a fixed compute budget.

6. Putting It All Together

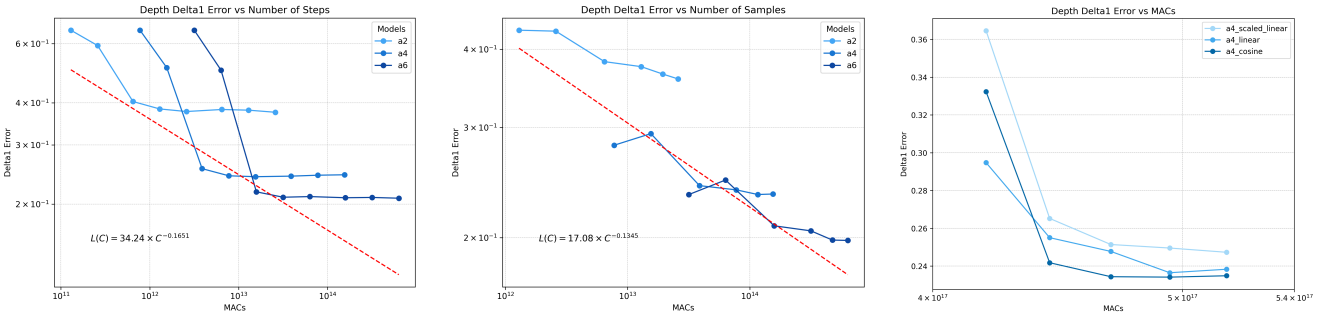
Using the lessons from our scaling experiments on depth estimation, we train diffusion models for optical flow prediction and amodal segmentation. We show that using diffusion models while considering efficient methods to scale training and test-time compute can provide substantial performance gains on visual perception tasks, achieving improved or similar performance as current state-of-the-art techniques. Our experiments provide insight on how to efficiently apply diffusion models for these visual perception tasks under limited compute budgets. Finally, we train a unified expert model, capable of performing all three visual perception tasks previously mentioned, displaying the generalizability. Our results show the effectiveness of our training and test-time scaling strategies, removing the need to use pre-trained models trained on internet-scale datasets to enable high-quality visual perception in diffusion models. Fig. 9 displays the predicted samples from our model.

6.1. Depth Estimation

We combine our findings from the ablation studies on depth estimation to create a model with the best training and inference configurations. We train a DiT-XL model from [32] on depth estimation data from Hypersim for 30K steps with a batch size of 1024, resolution of 512×512, and a learning rate exponentially decaying from $1.2e$-4 to $1.2e$-6. We use median compilation ensembling with a cosine noise variance schedule. From our scaling experiments, we found the optimal configuration for inference to be 200 denoising steps with $N = 5$ samples for ensembling. As shown in Table 3, our model achieves the same validation performance as Marigold on the Hypersim dataset and better performance on the ETH3D test set while being trained with lower resolution images and approximately **three orders of magnitude less pre-training data and compute**.

6.2. Optical Flow Prediction

Optical flow estimation involves predicting the motion of objects between consecutive frames in a video, represented as a dense vector field indicating pixel-wise displacement. We use a similar configuration as the depth estimation model for optical flow training. We train a DiT-XL model on the FlyingChairs dataset for 40K steps with batch size of 1024, resolution of 512×512, and learning rate exponentially decaying from $1.2e$-4 to $1.2e$-6. We compare our model's performance with other specialized optical flow prediction techniques in Table 4.



(a) Delta1 Error vs. Number of Steps (c) Delta1 Error vs. Number of Samples (e) Delta1 Error vs. Beta Schedules

(b) AbsRel Error vs. Number of Steps (d) AbsRel Error vs. Number of Samples (f) AbsRel Error vs. Beta Schedules

Figure 8. **Effect of Scaling Test-Time Compute and Different Noise Variance Schedules.** (a, b) Delta1 Error and Absolute Relative Error vs. Number of Sampling Steps measured at 1, 2, 5, 10, 20, 50, 100, 200 steps. (c, d) Delta1 Error and Absolute Relative Error vs. Number of Forward Passes measured at 1, 2, 5, 10, 20, 50 samples. (e, f) Delta1 Error and Absolute Relative Error with Different Variance Schedules measured at intervals of 5k training steps. We present different test-time scaling techniques and noise variance schedules on depth estimation metrics, highlighting power-law scaling and improved performance with optimized noise variance schedules.

Method	**Hypersim**		**ETH3D**		**NYUv2**		**ScanNet**		**DIODE**	
	AbsRel ↓	$\delta 1$ ↑	AbsRel ↓	$\delta 1$ ↑	AbsRel ↓	$\delta 1$ ↑	AbsRel ↓	$\delta 1$ ↑	AbsRel ↓	$\delta 1$ ↑
DiverseDepth	–	–	22.8	69.4	11.7	87.5	10.9	88.2	37.6	63.1
MiDaS	–	–	18.4	75.2	11.1	88.5	12.1	84.6	33.2	71.5
LeReS	–	–	17.1	77.7	9.0	91.6	9.1	91.7	27.1	76.6
Omnidata	–	–	16.6	77.8	7.4	94.5	7.5	93.6	33.9	74.2
HDN	–	–	12.1	83.3	6.9	94.8	8.0	93.9	24.6	78.0
DPT	–	–	7.8	94.6	9.8	90.3	8.2	93.4	18.2	75.8
Marigold	13.5	87.5	6.5	96.0	5.5	96.4	6.4	95.1	30.8	77.3
Ours	13.6	87.6	4.8	97.8	6.8	95.0	7.7	93.7	31.0	77.2

Table 3. **Depth Estimation Performance Comparison on Multiple Datasets.** We achieve state-of-the-art performance on the ETH3D dataset and competitive performance across all other benchmarks. Notably, we closely match the performance of Marigold across all datasets with significantly less training compute.

6.3. Amodal Segmentation

Amodal segmentation is the process of segmenting a complete object in an image, including the portions that are occluded or not directly visible, which can require higher-level reasoning for complex scenes. We provide the RGB input, a mask prompt of the visible portions of the object to segment, and a CLIP task embedding, which helps our model learn to complete the semantic object from the visible portions. This task is particularly challenging in scenes with significant occlusions or clutter, where accurate reconstruction often relies on contextual cues and prior knowledge of object shapes. We fine-tune a DiT-XL model on the pix2gestalt dataset [31] for 6K steps with a batch size of 4096, resolution of 256×256, and learning rate exponentially decaying from $1.2e$-4 to $1.2e$-6. We compare our model with other methods in Table 5, demonstrating competitive performance across a variety of occlusion levels.

Figure 9. **Depth Estimation, Optical Flow Estimation, and Amodal Segmentation Examples:** Each row showcases results from our models for different tasks. (a) Depth estimation, with relative scale and shift. (b) Optical flow, with scale and shift. (c) Amodal segmentation, where the model sees an RGB image and segmentation of the occluded object; the task is to predict the amodal image.

Method	FlyingChairs EPE ↓
DeepFlow	3.53
FlowNetS	2.71
FlowNetS+v	2.86
FlowNetS+ft	3.04
FlowNetC	2.19
FlowNetC+v	2.61
FlowNetC+ft	2.27
Ours (w/o ensembling)	3.45
Ours (w/ ensembling)	3.08

Table 4. **Optical Flow Comparison with Specialized Techniques.** We evaluate our optical flow model on the FlyingChairs validation set. Our model achieves similar end-point error as specialized methods, including DeepFlow [51] and FlowNet [12]. We train with significantly less data compared to other specialized methods, which use a several optical flow datasets. We generate predictions with and without test-time ensembling.

Method	COCO-A	P2G	MP3D
PCNet	81.35	–	–
PCNet-Sup	82.53	–	–
SAM	67.21	–	–
SD-XL Inpainting	76.52	–	–
pix2gestalt	82.9	88.7	61.5
Ours	82.9	88.6	63.9

Table 5. **Amodal Segmentation Performance (mIOU) Comparison Across Different Datasets.** This table compares mIOU performance across COCO-A, Pix2Gestalt, and MP3D datasets, showing the effectiveness of various methods. Our method is able to achieve competitive performance across all tasks, while training only on Pix2Gestalt.

6.4. One Model for All

We train a unified DiT-XL model for each of the different tasks. We train this model on a mixed dataset consisting of all three tasks. To train this generalist model, we modify the DiT-XL architecture by replacing the patch embedding layer with a separate `PatchEmbedRouter` module, which routes each VAE embedding to a specific input convolutional layer based perception task. This ensures the DiT-XL model is able to distinguish between the task-specific embeddings during fine-tuning. We use a similar training recipe as the previous experiments, using images with 512×512 resolution and a learning rate exponentially decaying from $1.2e$-4 to $1.2e$-6. Then, we upcycle the fine-tuned DiT-XL checkpoint to an DiT-XL-8E2A model, and continue fine-tuning for another 4K iterations. We display the generated predictions in Fig. 9, which exemplify the generalizability and transferability of our scaling techniques across a variety of perception tasks.

7. Conclusion

In our work, we examine the scaling properties of diffusion models for visual perception tasks. We explore various approaches to scale diffusion training, including increasing model size, mixture-of-experts models, increasing image resolution, and upcycling. We also efficiently scale test-time compute by exploiting the iterative nature of diffusion, which significantly improves downstream performance. Our experiments provide strong evidence of scaling, uncovering power laws across various training and inference scaling techniques. Finally, using our scaling laws, we train a generalist model that performs several perceptual tasks under a unified framework. We hope to inspire future work in scaling training and test-time compute for iterative generative paradigms such as diffusion for other tasks.

References

[1] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 2

[2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[3] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4175–4186, 2022. 2

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 3

[5] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. 5

[6] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 909–919, 2023. 2

[7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 1

[8] Yiqun Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. *arXiv preprint arXiv:2303.05021*, 2023. 2

[9] Karim El-Refai, Zeeshan Patel, Jonathan Pei, and Tianle Li. Swag: Storytelling with action guidance. 2024. 5

[10] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Jusnshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint*, 2024. 2, 3, 4

[11] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024. 2

[12] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015. 8

[13] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image, 2024. 2

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[15] Zhangxuan Gu, Haoxing Chen, and Zhuoer Xu. Diffusioninst: Diffusion model for instance segmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2730–2734. IEEE, 2024. 2

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[17] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021. 2

[18] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 867–876, 2022. 2

[19] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21741–21752, 2023. 2

[20] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 1, 2, 3, 6

[21] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[22] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*, 2022. 2

[23] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints, 2023. 4

[24] Hao Li, Yang Zou, Ying Wang, Orchid Majumder, Yusheng Xie, R Manmatha, Ashwin Swaminathan, Zhuowen Tu, Stefano Ermon, and Stefano Soatto. On the scalability of diffusion-based text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9400–9409, 2024. 2

[25] Y Lipman et al. Flow matching: Symmetrizing optimal transport and generative modeling. *arXiv preprint arXiv:2301.13003*, 2023. 2

[26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 2

[27] X Liu et al. Rectified flow: A unified approach for free-form generative models. *arXiv preprint arXiv:2209.07953*, 2022. 2

[28] Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Flowdiffuser: Advancing optical flow estimation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19167–19176, 2024. 1, 2

[29] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. 6

[30] OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. 1, 5

[31] Ege Ozguroglu, Ruoshi Liu, Dídac Surś, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 7

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4, 6

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[34] Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pretraining from videos. *arXiv preprint arXiv:2501.05453*, 2025. 3

[35] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 2

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 5

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 2

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[39] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 2

[40] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[41] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. 2

[42] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. 5

[43] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 5

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 2

[46] Haotian Sun, Tao Lei, Bowen Zhang, Yanghao Li, Haoshuo Huang, Ruoming Pang, Bo Dai, and Nan Du. Ec-dit: Scaling diffusion transformers with adaptive expert-choice routing, 2024. 2

[47] Haoru Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:8702–8716, 2022. 2

[48] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1747–1756, New York, New York, USA, 2016. PMLR. 2

[49] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 2

[50] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 2

[51] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 8

[52] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, pages 1336–1348. PMLR, 2022. 2

[53] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022. 3

[54] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5729–5739, 2023. 2

Scaling Properties of Diffusion Models For Perceptual Tasks

Supplementary Material

8. Scaling Power Law Derivation

We derive all scaling laws in our figures using an iterative method based on the convex hull. This procedure ensures the curve accurately captures the minimal envelope of the data, representing the scaling behavior of loss as a function of computational cost. The algorithm begins by aggregating all data points representing the relationship between the loss and MACs. We compute the convex hull of the aggregated points, which forms the smallest convex boundary enclosing all points. From this hull, the lower envelope is extracted. These lower hull points represent the minimal set of points along the loss vs. MACs curve, which define the primary trend. The scaling law is modeled as:

$$L(C) = a \times C^b, \tag{3}$$

where $L(C)$ is the loss/error, C represents the compute in MACs, and a, b are parameters to be optimized. The fitting process is initialized with reasonable guesses for these parameters and constraints to ensure the solution remains physically meaningful (e.g., non-negative losses). After fitting the initial curve to the lower hull points, the method identifies any data points that lie below the fitted curve. These points indicate regions where the current fit does not fully encapsulate the minimal envelope of the data. These points are added to the lower hull, and the convex hull is recalculated to include them. The fitting process is repeated iteratively until convergence, where either fewer than N_{max} points are found below the fitted curve or a maximum number of iterations is reached. This iterative process ensures the scaling law curve fully captures the trend defined by the lower envelope of the data. The final parameters a and b are determined after convergence, and the resulting curve represents the optimal scaling power law for the loss/error vs. compute relationship.

9. Noise Variance Schedule Visualization

During the denoising process, our mixture-of-experts generalist model refines depth latents from timestep $t = 1000$ to $t = 0$. At selected timesteps ($t \in \{1000, 800, 600, 400, 200, 0\}$), we project the current denoised depth latent into RGB space and compress the representation along the channel dimension to retrieve depth predictions.

To align these predictions with the ground truth, we apply least squares regression at each timestep to determine scaling and shifting parameters, γ and β, respectively. These parameters are used to scale and shift the predictions, ensuring consistency with the ground truth depth. Fig. 10 illustrates the progression of the denoising process, with the predictions approaching the ground truth as $t \to 0$.

10. Additional Results From Generalist Model

We visualize additional samples from our mixture-of-experts generalist model for depth estimation, optical flow estimation, and amodal segmentation in Fig. 11. Our model is able to generalize across the three tasks with accurate visual results, displaying the effectiveness of our scaling techniques to train a generalist diffusion model for perception.

11. Evaluation Metrics

We use a variety of metrics to evaluate our models. For depth estimation, we use Delta1 Accuracy and Absolute Relative Error metrics. The δ_1 accuracy measures the percentage of predicted depth values where the ratio between the prediction and ground truth (or its inverse) is within a threshold. The absolute relative error quantifies the mean of the absolute difference between the predicted and ground truth depths relative to the ground truth.

$$\delta_1 = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(\max\left(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}\right) < 1.25\right), \tag{4}$$

$$\text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\hat{D}_i - D_i\right|}{D_i}. \tag{5}$$

For optical flow estimation, we measure end-point error, which measures the average Euclidean distance between the predicted flow vectors and the ground truth flow vectors.

$$\text{EPE} = \frac{1}{N}\sum_{i=1}^{N} \left\|\hat{\mathbf{F}}_i - \mathbf{F}_i\right\|_2, \tag{6}$$

Finally, for amodal segmentation, we evaluate our model by computing the mIOU, calculated as the average IoU over all samples. IoU provides an intuitive measure of the overlap between the predicted segmentation and the ground truth, with values ranging from 0 (no overlap) to 1 (perfect overlap).

$$\text{IoU}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i} \tag{7}$$

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N} \text{IoU}_i \tag{8}$$

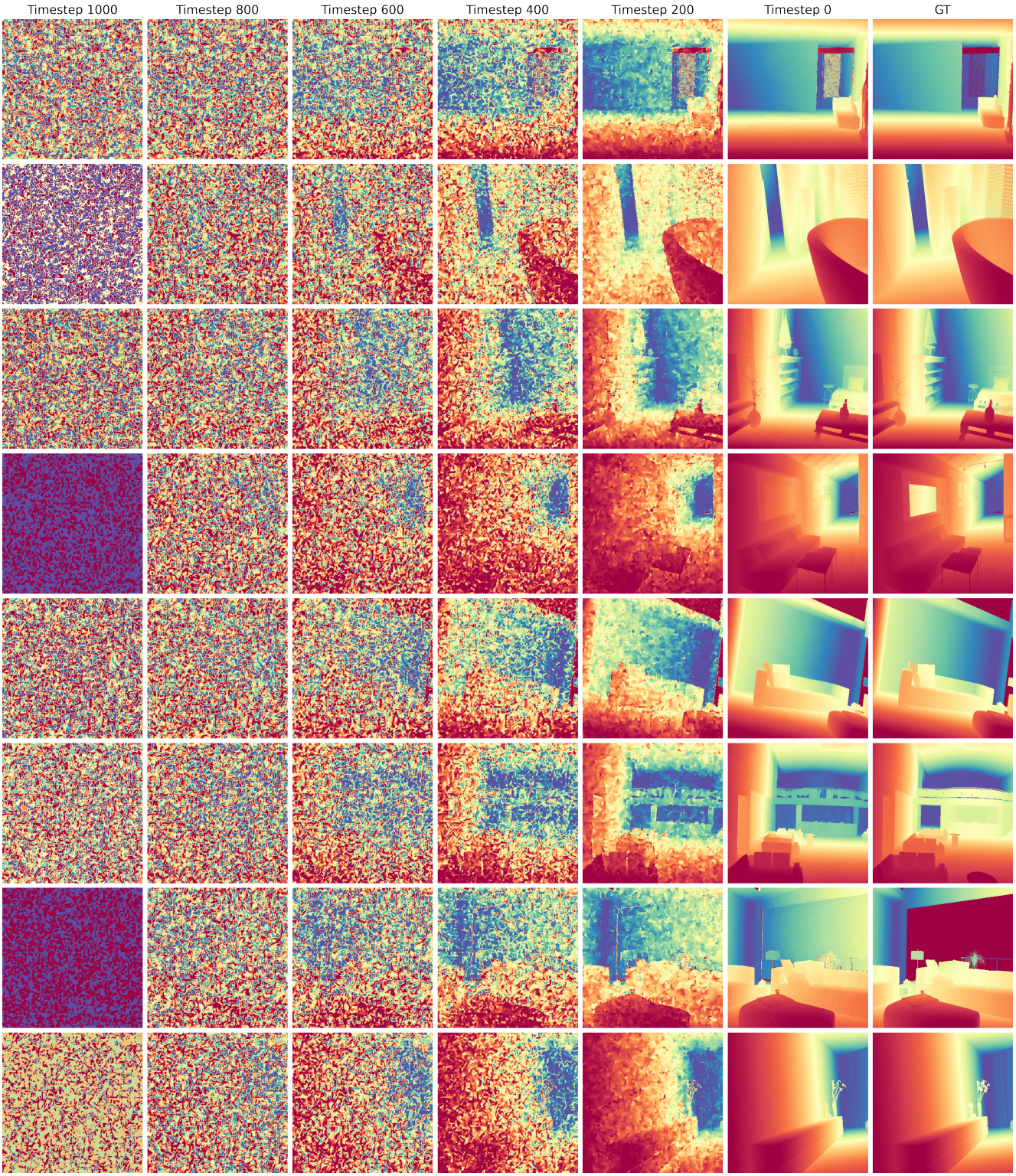


Figure 10. **Noise Variance Schedule Progression:** We project depth latents at uniform timesteps in the denoising process to show the predicted depth maps. The samples in this figure are generated from the Hypersim dataset.

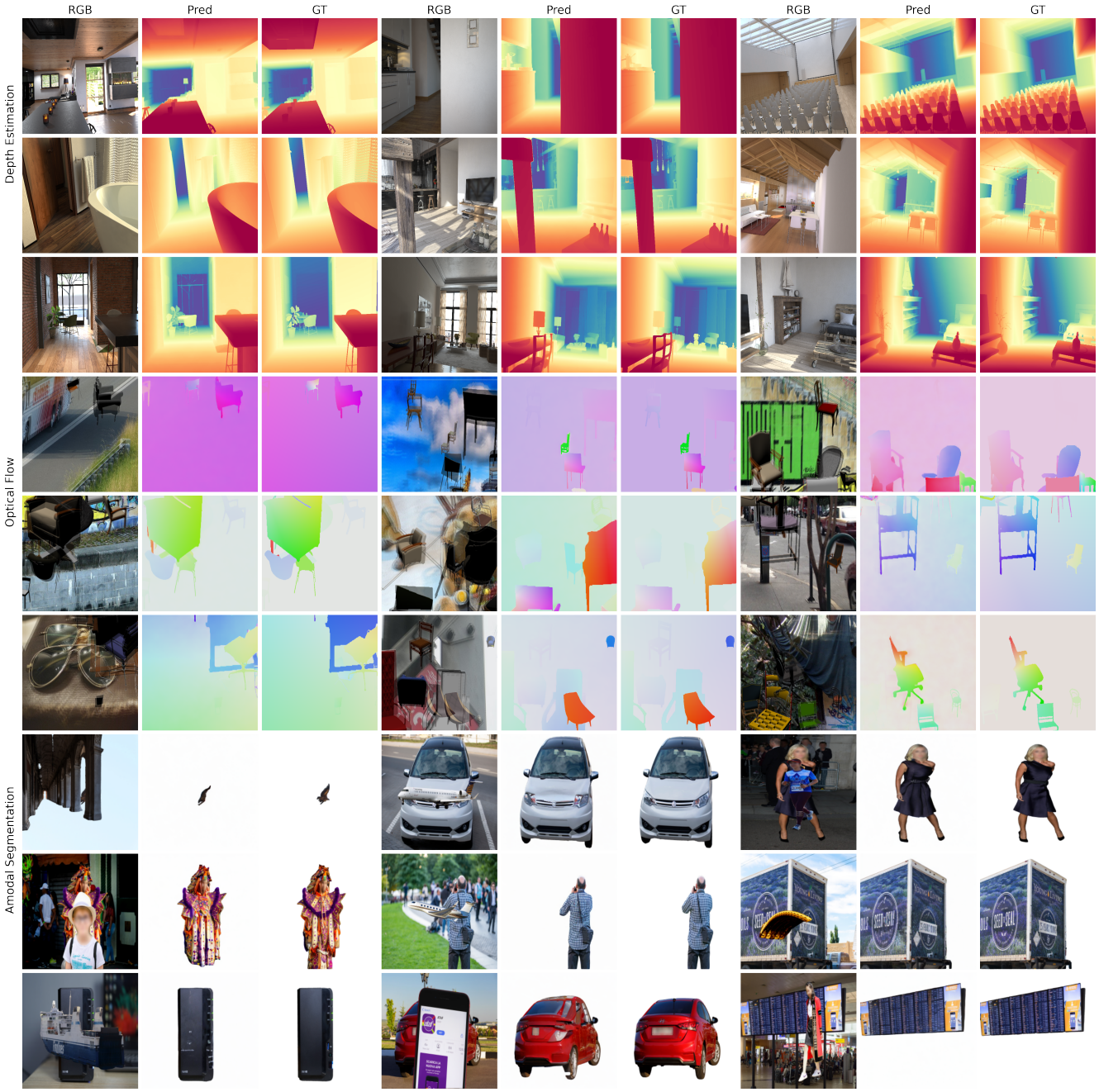


Figure 11. **Generalist Model Predictions**: We visualize additional samples generated from our mixture-of-experts generalist diffusion model. We generate the depth estimation samples from Hypersim, the optical flow estimation samples from FlyingChairs, and the amodal segmentation samples from Pix2Gestalt.