arXiv:2303.17559v2 [cs.CV] 13 May 2023

DDP: Diffusion Model for Dense Visual Prediction

Yuanfeng Ji[1*], Zhe Chen[3*], Enze Xie[2†], Lanqing Hong[2], Xihui Liu[1],
Zhaoqiang Liu[2], Tong Lu[3], Zhenguo Li[2], Ping Luo[1]
[1]The University of Hong Kong [2]Huawei Noah's Ark Lab [3]Nanjing University

https://github.com/JiYuanFeng/DDP

Abstract

We propose a simple, efficient, yet powerful framework for dense visual predictions based on the conditional diffusion pipeline. Our approach follows a "noise-to-map" generative paradigm for prediction by progressively removing noise from a random Gaussian distribution, guided by the image. The method, called DDP, efficiently extends the denoising diffusion process into the modern perception pipeline. Without task-specific design and architecture customization, DDP is easy to generalize to most dense prediction tasks, e.g., semantic segmentation and depth estimation. In addition, DDP shows attractive properties such as dynamic inference and uncertainty awareness, in contrast to previous single-step discriminative methods. We show top results on three representative tasks with six diverse benchmarks, without tricks, DDP achieves state-of-the-art or competitive performance on each task compared to the specialist counterparts. For example, semantic segmentation (83.9 mIoU on Cityscapes), BEV map segmentation (70.6 mIoU on nuScenes), and depth estimation (0.05 REL on KITTI). We hope that our approach will serve as a solid baseline and facilitate future research.

1. Introduction

Dense prediction tasks are the foundation of computer vision research, including a wide range of perceptual tasks such as semantic segmentation [21, 99], depth estimation [31, 70, 74], and optical flow [29, 31]. These tasks require correctly predicting the discrete labels or continuous values for all pixels in the image, which provides detailed contextual understanding and enables various applications.

Numerous methods have rapidly improved the result of perception tasks over a short period of time. In general terms, these methods can be divided into two paradigms: *discriminative-based* [30, 96, 85, 18] and *generative-based*

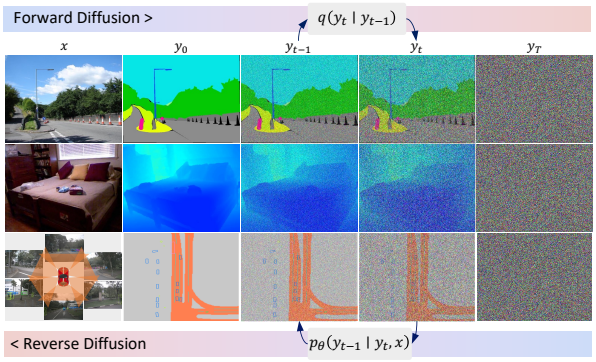


Figure 1. **Conditional diffusion pipeline for dense visual predictions**. Specifically, a conditional diffusion model is employed, where q is the forward diffusion process and p_θ is the inverse process. The framework iteratively transforms the noise sample $\boldsymbol{y}_T$, drawn from a standard Gaussian distribution, into the desired target prediction $\boldsymbol{y}_0$ under the guidance of the input image $\boldsymbol{x}$.

[84, 34, 39, 46, 88]. The former approach, which directly learns the mapping between input-output pairs and predicts in a single forward step, has become the current de-facto choice due to its simplicity and efficiency. Whereas, generative models aim at modeling the underlying distribution of the data, conceptually having a greater capacity to handle challenging tasks. However, they are often restricted by complex architecture customization as well as various training difficulties [67, 42, 6].

These challenges have been largely addressed by the diffusion and score-based models [35, 71, 75]. The solutions, based on *denosing diffusion process*, are conceptually simple: they apply a continuous diffusion process to transform data into noise and generate new samples by simulating the time-reversed diffusion process. These methods now enable easy training and achieve superior results on various generative tasks [57, 65, 63, 60]. Witnessing these great successes, there has been a recent surge of interest to introduce diffusion models to dense prediction tasks, including semantic segmentation [1, 14, 82, 81] and depth estimation [68]. However, these methods simply transfer the heavy

∗ Equal contribution.
† Corresponding author.

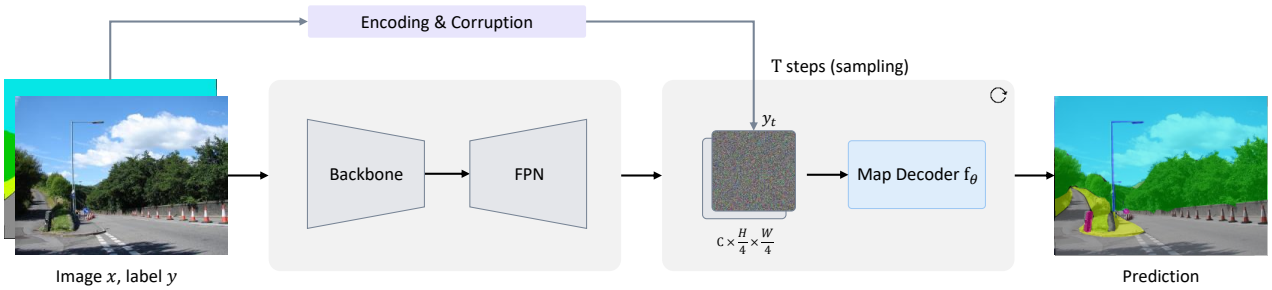


Figure 2. **The proposed DDP framework**. The image encoder extracts feature representation from the input image $\boldsymbol{x}$ as the condition. The map decoder takes the noisy map $\boldsymbol{y}_t$ as input and produces the denoised prediction under the guidance. During training, the noisy map $\boldsymbol{y}_t$ is constructed by adding Gaussian noise to the encoded ground truth. In inference, the noisy map $\boldsymbol{y}_t$ is randomly sampled from the Gaussian distribution and iteratively refined to obtain the desired prediction $\boldsymbol{y}_0$.

frameworks from image generation tasks to dense prediction, resulting in low efficiency, slow convergence, and suboptimal performance.

In this paper, we introduce a general, simple, yet effective diffusion framework for dense visual prediction. Our method named as DDP, which extends the denoising diffusion process into the modern perception pipeline effectively (see Figure 2). During training, the Gaussian noise controlled by a noise schedule [58] is added to the encoded ground truth to obtain the noisy maps. Then these noisy maps are fused with the conditional features from the image encoder, *e.g.*, Swin Transformer [52]. Finally, these fused features are fed to a lightweight map decoder to produce the predictions without noise. At the inference phase, DDP generates predictions by reversing the learned diffusion process, which adjusts a noisy Gaussian distribution to the learned map distribution under the guidance of the test images (see Figure 1).

Compared to previous cumbersome diffusion perception models [82, 81, 68], DDP decouples the image encoder and map decoder. The image encoder runs only once, while the diffusion process is performed only in the lightweight decoder head. With this efficient design, our proposed method can easily be applied to modern perception tasks. Furthermore, unlike previous single-step discriminative models, DDP is capable of performing iterative inference multiple times using the shared parameters and exhibits the following appealing properties: (1) dynamic inference to trade off computation and prediction quality and (2) natural awareness of the prediction uncertainty.

We evaluate DDP on three representative dense prediction tasks, including semantic segmentation, BEV map segmentation, and depth estimation, using six popular datasets (ADE20K [99], Cityscapes [21], nuScenes [7], KITTI [31], NYU-DepthV2 [70], and SUN RGB-D [74]). Our experimental results demonstrate that DDP significantly outperforms existing state-of-the-art methods. Specifically, on ADE20K, DDP achieves 46.1 mIoU with a single sampling step, which is significantly better than UperNet [83] and K-Net [95]. On nuScenes, DDP yields an mIoU of 70.3, which is clearly better than the BEVFusion [54] baseline that achieves an mIoU of 62.7. Furthermore, by increasing the sampling steps, DDP can achieve even higher performance on both ADE20K and nuScenes, reaching anmIoU of 47.0 and 70.6, respectively. Moreover, the gains are more versatile for different model architectures as well as model sizes. DDP achieves 83.9 mIoU on Cityscapes with the ConvNeXt-L backbone and produces a leading REL of 0.05 on KITTI with the Swin-L backbone.

Overall, our contributions in this work are three-fold.

- We formulate the dense visual prediction tasks as a general conditional denoising process, with simple yet highly effective designs.
- Our "noise-to-map" generative paradigm offers several appealing properties, such as the ability to perform dynamic inference and uncertain awareness.
- We conduct extensive experiments on three representative tasks with six diverse benchmarks. The results demonstrate that our method, which we refer to as DDP, achieves competitive performance when compared to previous discriminative methods.

2. Related Work

Diffusion Model. Diffusion [35, 71] and score-based generative models [73] have been particularly successful as generative models and achieve impressive results across various modalities, including images [60, 66, 27, 57, 25, 25], video [36, 37], audio [43], and biomedical [2, 77, 69, 22]. Given the notable achievements of diffusion models in these respective domains, leveraging such models to develop generation-based perceptual models would prove to be a highly promising avenue to push the boundaries of perceptual tasks to newer heights.

Dense Prediction. The perception of real-world scenes via pixel-by-pixel classification or regression is commonly formulated as dense prediction tasks, such as semantic segmentation [21, 99], depth estimation [31, 70, 74], and optical flow [29, 31]. Numerous methods have emerged and achieved tremendous progress, and these advances can be roughly divided to: multi-scale feature aggregation [9, 10, 83], high-capacity backbone [85, 97, 61] and powerful decoder head [76, 95, 19, 40]. In this paper, as shown in Figure 1, which differs from previous discriminative-based methods, we explore a generative "noise-to-map" paradigm for general dense prediction tasks.

Diffusion Models for Dense Prediction. With the recent success of diffusion models in generation tasks, there has been a noticeable rise in interest to incorporate them into dense visual prediction tasks. Several pioneering works [82, 1, 81, 14, 68, 12] attempted to apply the diffusion model to visual perception tasks, *e.g.* image segmentation or depth estimation task. For example, Wolleb *et al.* [81] explore the diffusion model for medical image segmentation. Pix2Seq-D [14] applies the bit diffusion model [16] for panoptic segmentation. Our concurrent work DepthGen [68] involves diffusion pipeline to the task of depth estimation. For all the diffusion models listed above, one or two parameter-heavy convolutional U-Nets [64] are adopted, leading to low efficiency, slow convergence, and sub-optimal performance. In this work, as illustrated in Figure 2, we introduce a simple yet effective diffusion framework, which extends the denoising diffusion process into the modern perception pipeline while maintaining accuracy and efficiency.

3. Methodology

3.1. Preliminaries

Dense Prediction. The objective of dense prediction tasks is to predict discrete labels or continuous values, denoted as $\boldsymbol{y}$, for every pixel present in the input image $\boldsymbol{x} \in \mathbb{R}^{3\times h\times w}$.

Conditional Diffusion Model. The conditional diffusion model, which is an extension of the diffusion model [35, 71, 75], belongs to the category of likelihood-based models inspired by non-equilibrium thermodynamics. The conditional diffusion model assumes a forward noising process by gradually adding noise to the data sample, which is defined as:

$$q\left(\boldsymbol{z}_t \mid \boldsymbol{z}_0\right) = \mathcal{N}\left(\boldsymbol{z}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right), \tag{1}$$

which transforms the data sample $\boldsymbol{z}_0$ to a latent noisy sample $\boldsymbol{z}_t$ for $t \in \{0, 1, \ldots, T\}$. The constants $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s = \prod_{s=0}^{t}\left(1-\beta_s\right)$ and β_s represents the noise schedule [58, 35]. During training, the reverse process model $f_\theta\left(\boldsymbol{z}_t, \boldsymbol{x}, t\right)$ is trained to predict $\boldsymbol{z}_0$ from $\boldsymbol{z}_t$ under the guidance of condition $\boldsymbol{x}$ by minimizing the training objective function (*i.e.*, l_2 loss). At the inference stage, predicted data sample $\boldsymbol{z}_0$ is reconstructed from a random noise $\boldsymbol{z}_T$ with the model f_θ, conditional input $\boldsymbol{x}$, and a translation rule [35, 72] in a markovian way, *i.e.*, $\boldsymbol{z}_T \rightarrow \boldsymbol{z}_{T-\Delta} \rightarrow \ldots \rightarrow \boldsymbol{z}_0$, which can be formulated as:

$$p_\theta\left(\boldsymbol{z}_{0:T} \mid \boldsymbol{x}\right) = p\left(\boldsymbol{z}_T\right) \prod_{t=1}^{T} p_\theta\left(\boldsymbol{z}_{t-1} \mid \boldsymbol{z}_t, \boldsymbol{x}\right). \tag{2}$$

In this paper, our goal is to solve dense prediction tasks via the conditional diffusion model. In our setting, the data samples are the ground truth map $\boldsymbol{z}_0 = \boldsymbol{y}$, and a neural network f_θ is trained to predict $\boldsymbol{z}_0$ from random noise $\boldsymbol{z}_t \sim \mathcal{N}(0, \mathbf{I})$ conditioned on the corresponding image $\boldsymbol{x}$.

3.2. Architecture

Since the diffusion model generates samples progressively, it requires multiple runs of the model in the inference stage. Previous methods [82, 68, 81] apply the model f_θ in multiple steps on the raw image $\boldsymbol{x}$, which significantly increases the computational overhead. To alleviate this issue, we separate the entire model into two parts: image encoder and map decoder, as shown in Figure 2. The image encoder forwards only once to extract the feature map from the input image $\boldsymbol{x}$. Then the map decoder employs it as the condition rather than the raw image $\boldsymbol{x}$, to gradually refine the prediction from the noisy map $\boldsymbol{y}_t$.

Image Encoder. The image encoder receives the raw image $\boldsymbol{x}$ as input and generates multi-scale features at 4 different resolutions. Subsequently, these multi-scale features are fused using the FPN [51] and aggregated by a 1×1 convolution. The produced feature map, with the resolution of $256 \times \frac{h}{4} \times \frac{w}{4}$, is employed as the condition for the map decoder. In contrast to the previous methods [1, 82, 68], DDP is able to work with modern network architectures such as ConvNext [53] and Swin Transformer [52].

Map Decoder. The map decoder f_θ takes as input the noisy map $\boldsymbol{y}_t$ and the feature map from the image encoder via concatenation and performs a pixel-by-pixel classification or regression. Following the common practice [18, 100, 93] in modern perception pipelines, we simply stack six layers of deformable attention as the map decoder. Compared to previous works [1, 82, 68, 14, 81] that use the parameter-intensive U-Nets, our map decoder is lightweight and compact, allowing efficient reuse of the shared parameters during the multi-step reverse diffusion process.

3.3. Training

During training, we first construct a diffusion process from the ground truth $\boldsymbol{y}$ to the noisy map $\boldsymbol{y}_t$ and then train the model to reverse this process. The training procedure

Algorithm 1: DDP Training

```
def train(images, maps):
  """images: [b, 3, h, w], maps: [b, 1, h, w]"""
  img_enc = image_encoder(images) # encode image
  map_enc = encoding(maps) # encode gt
  map_enc = (sigmoid(map_enc) * 2 - 1) * scale
  # corrupt gt
  t, eps = uniform(0, 1), normal(mean=0, std=1)
  map_crpt = sqrt(alpha_cumprod(t)) * map_enc +
             sqrt(1 - alpha_cumprod(t)) * eps
  # predict and backward
  map_pred = map_decoder(map_crpt, img_enc, t)
  loss = objective_func(map_pred, maps)
  return loss
```

Algorithm 2: DDP Sampling

```
def sample(images, steps, td=1):
  """steps: sample steps, td: time difference"""
  img_enc = image_encoder(images)
  map_t = normal(0, 1) # [b, 256, h/4, w/4]
  for step in range(steps):
    # time intervals
    t_now = 1 - step / steps
    t_next = max(1 - (step + 1 + td) / steps, 0)
    # predict map_0 from map_t
    map_pred = map_decoder(map_t, img_enc, t_now)
    # estimate map_t at t_next
    map_t = ddim(map_t, map_pred, t_now, t_next)
  return map_pred
```

for DDP is provided in Algorithm 1 (for more details please refer to Appendix A).

Label Encoding. Standard diffusion models assume continuous data, which makes them a convenient choice for regression tasks with continuous values (*e.g.*, depth estimation). However, existing studies [14, 16] show that they are unsuitable for discrete labels (*e.g.*, semantic segmentation). Therefore, we explore several encoding strategies for the discrete labels, including: (1) One-hot encoding, which represents categorical labels as binary vectors of 0 and 1; (2) Analog bits encoding [14], which first converts discrete integers into bit strings, and then casts them as real numbers; (3) Class embedding, which uses a learnable embedding layer to project discrete labels into a high-dimensional continuous space, with a sigmoid function for normalization. For all of these strategies, we normalize and scale the range of encoded labels within $[-\text{scale}, +\text{scale}]$, as shown in Algorithm 1. Notably, the scaling factor scale controls the signal-to-noise ratio (SNR) [14, 13], which is an important hyper-parameter for diffusion models. We compare these strategies in Table 5a and find class embedding work best. More discussions are in Section 4.5.

Map Corruption. We add Gaussian noise to corrupt the encoded ground truth, obtaining the noisy map $\boldsymbol{y}_t$. As shown in Equation (1), the intensity of corruption noise is controlled by α_t, which adopts the monotonically decreasing schedule for α_t in different time steps $t \in [0, 1]$. Different noise scheduling strategies, including cosine schedule [58] and linear schedule [35], are compared and discussed in Section 4.5. We found that the cosine schedule usually worked best in our benchmark tasks.

Objective Function. Standard diffusion models are trained with l_2 loss, which is reasonable for dense prediction tasks, but we found that adopting a task-specific loss works better for supervision, *e.g.*, cross-entropy loss for semantic segmentation, sigloss for depth estimation.

3.4. Inference

Given a test image as condition input, the model starts with a random noise map sampled from a Gaussian distribution and gradually refines the prediction, we summarize the inference procedure in Algorithm 2.

Sampling Rule. We choose the DDIM update rule [72] for the sampling. In each sampling step t, the random noise $\boldsymbol{y}_T$ or the predicted noisy map $\boldsymbol{y}_{t+1}$ from the last step is fused with the conditional feature map, and sent to the map decoder f_θ for map prediction. After getting the predicted result of the current step, we compute the noisy map $\boldsymbol{y}_t$ for the next step using the reparameterization trick. Following [15, 14, 12], we use the asymmetric time intervals (controlled by a hyper-parameter td) during the inference stage, and $td = 1$ works best in our method.

Sampling Drift. As displayed in Figure 3a, we empirically observe that the model performance improves in a few sampling steps and then declines slightly as the number of steps increases. Similar observations can also be found in [12, 11, 68]. This performance decline can be attributed to the "sampling drift" challenge, which refers to the discrepancy between the distribution of training and sampling data. During training, the model is trained to inverse the noisy ground truth map, while during testing, the model is inferred to remove noise from its "imperfect" prediction, which drifts away from the underlying corrupted distributions. This drift becomes pronounced with smaller time steps t, owing to the compounded errors, and is further intensified when a sample deviates more substantially from the distribution of ground truth [24].

To verify our hypothesis, in the last 5k iterations of training, we construct $\boldsymbol{y}_t$ using the model's prediction rather than the ground truth. The approach transforms the training target to remove the added noise on its own predictions, thereby aligning the data distribution of training and testing. We name this approach "*self-aligned denoising*." As revealed in Figure 3a, this approach tends to produce satura-

tion instead of performance degradation. Our findings suggest that incorporating the diffusion process into perception tasks could enhance efficacy compared to image generation (*e.g.*, about 50 DDIM steps for image generation). In other words, the proposed DDP can improve efficiency (*e.g.*, satisfied results in 3 iterative steps) while retaining the benefits of the diffusion model. More discussions can be found in Appendix A.

Multiple Inference. By virtue of the multi-step sampling procedure, our method supports dynamic inference, which has the flexibility to trade compute for prediction quality. Besides, it naturally enables the assessment of the reliability and uncertainty of model predictions.

4. Experiment

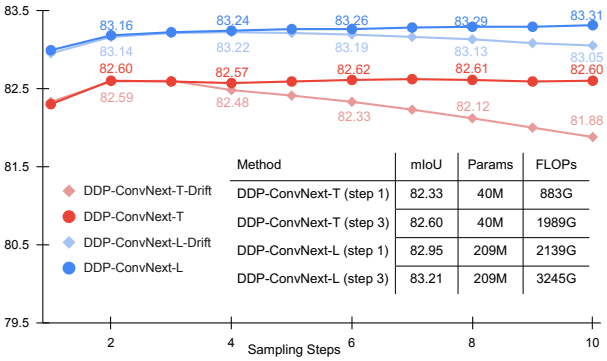
We first present the appealing properties of our DDP, followed by empirical evaluations of its performance against leading methods on several representative tasks, including semantic segmentation, BEV map segmentation, and monocular depth estimation. Finally, we provide ablation studies on the DDP components. Due to space limitations, more implementation details and experimental results are provided in Appendix B and Appendix C, respectively.

4.1. Main Properties

We explore and show properties of DDP in Figure 3 using the default setting in Section 4.2. With such a multi-step sampling procedure, we have the flexibility to trade computational cost for prediction quality. Furthermore, the stochastic sampling process allows the computing of pixel-wise uncertainty maps of the prediction.

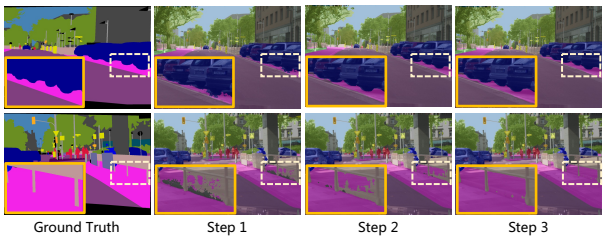
Dynamic Inference. We evaluate DDP with ConvNext-T and ConvNext-L backbones by increasing their sampling steps from 1 to 10. The results are presented in Figure 3a. It can be seen that the DDP can continuously improve its performance by using more sampling steps. For example, DDP with ConvNext-T shows an increase from 82.33 mIoU (1 step) to 82.60 mIoU (3 steps), and we visualize the inference trajectory in Figure 3b. In comparison to the previous single-step method, our approach boasts the flexibility to balance computational cost against accuracy. This means our method can be adapted to different trade-offs between speed and accuracy under various scenarios without the need to retrain the network.

Uncertainty Awareness. In addition to the performance gains, the proposed DDP can naturally provide uncertainty estimates. In the multi-step sampling process, we can simply count the pixels where the predicted result of each step differs from the result of the previous step, and finally, we simply normalize this change count map to 0-1 and obtain an uncertainty map. In comparison, DDP is naturally and easily capable of estimating uncertainty, whereas previous methods [55, 33] require complicated modeling such as Bayesian networks.

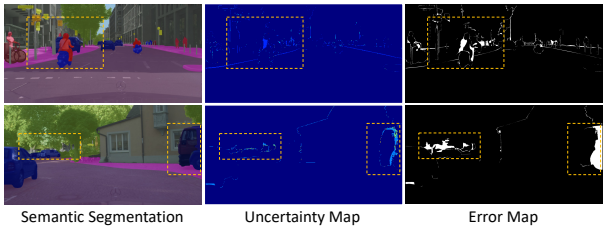


Method	mIoU	Params	FLOPs
DDP-ConvNext-T (step 1)	82.33	40M	883G
DDP-ConvNext-T (step 3)	82.60	40M	1989G
DDP-ConvNext-L (step 1)	82.95	209M	2139G
DDP-ConvNext-L (step 3)	83.21	209M	3245G

(a) **Dynamic inference**. The results of multiple inference on Cityscapes.



(b) **Inference trajectory**. Predicted mask results on different time steps.



(c) **Uncertainty awareness**. High response areas in the uncertainty map indicate high estimated uncertainty and are highly positively correlated with white areas in the error map, which indicate misclassified points. Zoom in for better visualization.

Figure 3. **DDP enjoys two appealing properties:** dynamic inference to trading off computation and prediction quality and the natural awareness of the prediction uncertainty.

4.2. Semantic Segmentation

Datasets. We evaluate the proposed DDP using two widely used datasets: ADE20K [99] and Cityscapes [21]. ADE20K is a large-scale scene parsing dataset with over 20,000 images, and Cityscapes is a street scene dataset with high-quality pixel-level annotations for 5,000 images.

Settings. In the training phase, following common practices [80, 17, 85, 79], the crop size is set to 512×512 for ADE20K, and 512×1024 for Cityscapes. We optimize our DDP models using the AdamW [56] optimizer, with an initial learning rate of 6×10^{-5} and a weight decay of 0.01. All models are trained for 160k iterations and compared fairly

Method	Backbone	#Param	FLOPs	mIoU	+MS
UperNet [83]	Swin-T	60M	236G	44.5	45.8
Region Rebalance [23]	Swin-T	60M	236G	45.0	46.5
MaskFormer [19]	Swin-T	42M	55G	46.7	48.8
Mask2Former [18]	Swin-T	47M	74G	47.7	49.6
K-Net [95]	Swin-T	73M	256G	45.8	46.3
SenFormer [5]	Swin-T	144M	179G	46.0	46.4
Non-diffusion Baseline	Swin-T	35M	111G	44.9	46.1
DDP (step 1)	Swin-T	40M	113G	46.1	47.6
DDP (step 3)	Swin-T	40M	252G	47.0	47.8
UperNet [83]	Swin-S	81M	259G	47.6	49.5
DDP (step 1)	Swin-S	61M	136G	48.4	49.7
DDP (step 3)	Swin-S	61M	276G	48.7	49.7
UperNet [83]	Swin-B	121M	297G	48.1	49.7
DDP (step 1)	Swin-B	99M	173G	49.2	50.8
DDP (step 3)	Swin-B	99M	312G	49.4	50.8
UperNet [83]	Swin-L†	234M	411G	52.1	53.5
DDP (step 1)	Swin-L†	207M	285G	53.1	54.4
DDP (step 3)	Swin-L†	207M	425G	53.2	54.4

Table 1. **Semantic segmentation on ADE20K val set.** We report single-scale (SS) and multi-scale (MS) mIoU. The FLOPs are measured with 512×512 inputs. Backbones pre-trained on ImageNet-22K are marked with †.

with previous non-diffusion methods.

Results on ADE20K. Table 1 presents the semantic segmentation performance of DDP on ADE20K [99], which shows that our method consistently outperforms many representative methods [83, 23, 95, 5] and the non-diffusion baseline across different backbones. For instance, when using Swin-T [52] as the backbone, our DDP (step 1) yields a promising result of 46.1 mIoU, surpassing the non-diffusion baseline (DDP *w/o* diffusion process) by 1.2 points (46.1 *vs.* 44.9). Moreover, our DDP (step 3) can further enhance the performance to 47.0 mIoU, attaining a remarkable gain of 0.9 points by multi-steps of denoising diffusion. With the Swin-L backbone, our DDP (step 3) achieves the best performance of 53.2 mIoU, which is 1.1 points (53.2 *vs.* 52.1) better than UperNet with comparable FLOPs. These results suggest that our DDP not only achieves a performance gain but also offers more flexibility than previous methods.

Results on Cityscapes. We compare our DDP with various representative models on Cityscapes [21] in Table 2, such as Segmenter [76], SETR [97], SegFormer [85], DiversePatch [32], and Mask2Former [18], and so on. As shown, we conduct extensive experiments based on ConvNeXt [53] and Swin [52] with different model sizes. When using ConvNeXt-L† as the backbone, our DDP (step 1) produces a competitive result of 82.95 mIoU, and it can be further boosted to 83.21 mIoU (step 3). This phenomenon was also observed when taking Swin-T as the backbone, and the mIoU increased from 80.96 to 81.24 through additional 2 sampling steps. These experimental results demonstrate the scalability of our methodology, which can be applied to different model structures of arbitrary size. Moreover, once again, the experimental results show that DDP achieves progressive improvements through multi-step denoising diffusion while keeping comparable computational overhead.

Method	Backbone	#Param	FLOPs	mIoU	+MS
Segmenter [76]	ViT-L†	333M	2685G	79.10	81.30
SETR-PUP [97]	ViT-L†	318M	2955G	79.34	82.15
StructToken [50]	ViT-L†	364M	2913G	80.05	82.07
OCRNet [91, 92]	HRFormer-B	56M	2240G	81.90	82.60
SegFormer-B5 [85]	MiT-B5	85M	1448G	82.25	83.48
DiversePatch [32]	Swin-L†	234M	3190G	82.70	83.60
Mask2Former [18]	Swin-L†	216M	2113G	83.30	84.30
DDP (step 1)	Swin-T	39M	885G	80.96	82.25
DDP (step 3)	Swin-T	39M	1992G	81.24	82.46
DDP (step 1)	Swin-S	61M	1067G	82.17	83.06
DDP (step 3)	Swin-S	61M	2174G	82.41	83.21
DDP (step 1)	Swin-B	99M	1357G	82.37	83.36
DDP (step 3)	Swin-B	99M	2464G	82.54	83.42
DDP (step 1)	ConvNext-T	40M	883G	82.33	83.00
DDP (step 3)	ConvNext-T	40M	1989G	82.60	83.15
DDP (step 1)	ConvNext-S	62M	1059G	82.37	83.38
DDP (step 3)	ConvNext-S	62M	2166G	82.69	83.58
DDP (step 1)	ConvNext-B	100M	1340G	82.59	83.47
DDP (step 3)	ConvNext-B	100M	2447G	82.78	83.49
DDP (step 1)	ConvNext-L†	209M	2139G	82.95	83.76
DDP (step 3)	ConvNext-L†	209M	3245G	83.21	83.92

Table 2. **Semantic segmentation on Cityscapes val set.** We report single-scale (SS) and multi-scale (MS) mIoU. The FLOPs are measured with 1024×2048 inputs. Backbones pre-trained on ImageNet-22K are marked with †.

Discussion. The original intention of DDP is to design a diffusion-based general framework for various dense prediction tasks. Although its segmentation performance is slightly lower than its specialized counterpart Mask2Former [18], it remains highly competitive and has several attractive features. How to design a segmentation-specific diffusion framework to achieve better performance than Mask2Former is left for future research.

4.3. BEV Map Segmentation

Dataset. We conduct our experiments of BEV map segmentation on the nuScenes [7] dataset. It is a large-scale autonomous driving perception dataset, which includes over 1000 urban road scenes covering different time periods and weather conditions in two cities, Boston and Singapore.

Settings. We further verify the DDP framework on the BEV map segmentation task. Specifically, we equip our method with the representative method BEVFusion [54], where we directly replace its segmentation head with the proposed map decoder for the diffusion process. We follow evaluation protocol from [54] and compare the results with state-of-the-art methods [86, 89, 54, 4]. We report the IoU of 6 background classes, including drivable space (Dri), pedestrian crossing (Ped), walk-way (Wal), stop line (Sto), car-parking area (Car), and lane divider (Div), and use the

Method	Modality	Dri	Ped	Wal	Sto	Car	Div	Mean
OFT [62]	C	74.0	35.3	45.9	27.5	35.9	33.9	42.1
LSS [59]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT [98]	C	74.3	36.8	39.9	25.8	35.0	29.4	40.2
M^2BEV [86]	C	77.2	-	-	-	-	40.5	-
BEVFusion [54]	C	81.7	54.8	58.4	47.4	50.7	46.4	56.6
X-Align [4]	C	82.4	55.6	59.3	49.6	53.8	47.4	58.0
DDP (step 1)	C	83.2	58.5	61.6	52.4	51.1	48.9	59.3
DDP (step 3)	C	83.6	58.3	61.8	52.3	51.4	49.2	**59.4**
PointPainting [78]	C+L	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP [89]	C+L	76.1	48.7	57.0	36.9	33.0	42.2	49.0
BEVFusion [54]	C+L	85.5	60.5	67.6	52.0	57.0	53.7	62.7
X-Align [4]	C+L	86.8	65.2	70.0	58.3	57.1	58.2	65.7
DDP (step 1)	C+L	89.3	69.5	74.8	62.5	63.5	62.3	70.3
DDP (step 3)	C+L	89.4	69.8	75.0	63.0	63.8	62.6	**70.6**

Table 3. **BEV map segmentation on nuScenes val set.** We report the IoU of 6 background classes and the mean IoU. "C" and "L" denotes the camera modality and LiDAR modality, respectively.

mean IoU as the primary evaluation metric. Other training settings are kept the same as [54] for fair comparisons.

Results. We show the results of our BEV map segmentation experiments in Table 3, which exhibit the superior performance of our approach, over existing state-of-the-art methods. Specifically, in the camera-only scenario, our DDP (step 1) attains a 59.3 mIoU score on the nuScenes validation dataset, which surpasses the previous best method X-Align [4] by 1.3 mIoU (59.3 *vs.* 58.0). By iteratively refining the output of the model, DDP (step 3) sets a new state-of-the-art record of 59.4 mIoU solely based on camera modality. In the multi-modality setting, we improve the segmentation results of our DDP (step 1) to 70.3 mIoU by combining LiDAR information, significantly higher than the current state-of-the-art methods [54, 4] by at least 4.6 mIoU. Remarkably, this performance can be further enhanced to a maximum of 70.6 mIoU by leveraging the benefits of iterative denoising diffusion. In summary, these results demonstrate that DDP can be easily generalized to other tasks and obtain performance gains, proving the effectiveness and generalization of our approach.

4.4. Depth Estimation

Datasets. We evaluate the depth estimation performance of DDP on three prominent datasets, namely KITTI [31], NYU-DepthV2 [70], and SUN RGB-D [74]. (1) The KITTI dataset encompasses stereo image pairs and corresponding ground truth depth maps for outdoor scenes captured by a car-mounted camera. Following common practices [28, 48], we use about 26K left-view images for training and 697 images for testing. (2) The NYU dataset contains RGB-Depth images for indoor scenes captured at a resolution of 640×480. Similar to prior research [48], the model is trained on 24K train images and evaluated on the reserved 652 images. (3) The SUN RGB-D dataset is a vast collection of around 10K indoor images. We employ it to evaluate the generalization abilities of our NYU pre-trained models. The results on KITTI are shown in the main paper, while others will be provided in the supplementary material.

Settings. We incorporate the DDP model into the codebase developed by [48] for depth estimation experiments. We excluded the discrete label encoding module as the task requires continuous value regression. All experimental settings are the same as [48] for a fair comparison.

Metrics. Typically, the evaluation of depth estimation methods employs the following metrics: accuracy under threshold ($\delta_i < 1.25^i, i = 1, 2, 3$), mean absolute relative error (REL), mean squared relative error (SqRel), root mean squared error (RMSE), root mean squared log error (RMSE log), and mean log10 error (log10).

Results. Table 4 shows the depth estimation results on the KITTI dataset. We compare the proposed DDP models with state-of-the-art depth estimators. Specifically, we choose DepthFormer [48] and DepthGen [68] as our main competitors, in which DepthFormer is a strong counterpart and achieved leading performance, while DepthGen is a concurrent work of ours and is also a diffusion-based depth estimator. As we can observe, although the performance on this benchmark tends to be saturated, our DDP models still outperform all the competitors with clear margins in most metrics, such as REL, SqRel, and RMSE. For instance, equipped with Swin-L†, our method achieves a state-of-the-art RMSE log of 0.076 by 3 steps of denoising diffusion. Compared with the concurrent diffusion-based model [68], we find that: (1) DDP outperforms DepthGen with clear margins, particularly in regards to the RMSE↓ metric (2.072 *vs.* 2.985), which can be contributed by the equipped advanced pipeline design (e.g., Swin Transformer *vs.* U-Net). (2) DDP is more lightweight and efficient compared to DepthGen, as the denoising diffusion process occurs solely on the decoder head, whereas with DepthGen, the process occurs on the entire model.

4.5. Ablation Study

We conduct ablation studies on the ADE20K semantic segmentation. All models are trained using our DDP with Swin-T [52] backbone for 160k iterations. Other settings are the same as the settings in Section 4.2.

Label Encoding. Since the labels of semantic segmentation are discrete, we need to encode them first. As shown in Table 5a, here we study the effect of three different strategies. For each of them, we search the optimal scaling factor. The results show that class embedding is a better strategy to encode semantic labels than one-hot and analog bits [14].

Signal Scale. As shown in Table 5b, we search for the best scaling factor for the class embedding strategy. As can be seen, when we use a larger scaling factor than 0.01, the

Method	Backbone	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	SqRel $\downarrow$	RMSE $\downarrow$	RMSE log $\downarrow$
DORN [30]	ResNet-101	0.932	0.984	0.994	0.072	0.307	2.727	0.120
VNL [90]	ResNeXt-101	0.938	0.990	0.998	0.072	-	3.258	0.117
BTS [44]	DenseNet-161	0.956	0.993	0.998	0.059	0.245	2.756	0.096
TransDepth [87]	ResNet-50 + ViT-B	0.956	0.994	**0.999**	0.064	0.252	2.755	0.098
DPT [61]	ResNet-50 + ViT-B	0.959	0.995	**0.999**	0.062	-	2.573	0.092
AdaBins [3]	EfficientNet-B5 + Mini-ViT	0.964	0.995	**0.999**	0.058	0.190	2.360	0.088
DepthFormer [48]	ResNet-50 + Swin-T	0.966	0.995	**0.999**	0.056	0.177	2.252	0.086
DepthFormer [48]	ResNet-50 + Swin-L†	**0.975**	**0.997**	**0.999**	0.052	0.158	2.143	0.079
BinsFormer [49]	Swin-L†	0.974	**0.997**	**0.999**	0.052	0.151	2.098	0.079
DepthGen (step 8)* [68]	Efficient U-Net	0.953	0.991	0.998	0.064	0.356	2.985	0.100
DDP (step 3)	Swin-T	0.969	0.996	**0.999**	0.054	0.168	2.172	0.083
DDP (step 3)	Swin-S	0.970	0.996	**0.999**	0.053	0.167	2.171	0.082
DDP (step 3)	Swin-B†	0.973	**0.997**	**0.999**	0.051	0.155	2.119	0.078
DDP (step 3)	Swin-L†	**0.975**	**0.997**	**0.999**	**0.050**	**0.148**	**2.072**	**0.076**

Table 4. **Depth estimation on the KITTI val set.** Backbones pre-trained on ImageNet-22K are marked with †. We report the performance of DDP with 3 diffusion steps. The best and second-best results are bolded or underlined, respectively. $\downarrow$ means lower is better, and $\uparrow$ means higher is better. * denotes best results of our concurrent work [68].

Type	mAcc	mIoU
analog bits	57.6	46.2
onehot	56.8	46.2
embedding	**58.4**	**47.0**

(a) **Label encoding**. We find class embedding works best.

Scale	mAcc	mIoU
0.001	56.6	45.4
0.01	**58.4**	**47.0**
0.02	57.5	46.8
0.04	56.8	45.9
0.1	55.0	44.0

(b) **Scaling factor**. The best scaling factor is 0.01.

Type	mAcc	mIoU
cosine	**58.4**	**47.0**
linear	56.3	45.1

(c) **Noise schedule**. Cosine works best.

L	mAcc	mIoU	#Param
1	56.1	44.5	2.4M
2	56.5	45.0	3.6M
4	57.2	45.7	6.0M
6	**58.4**	**47.0**	**8.4M**
12	55.7	46.0	15.6M

(d) **Decoder depth** L. Six blocks work best.

Step	mIoU	FLOPs	FPS
1	45.8	256G	18
1	46.1	113G	19
2	46.8	182G	15
3	**47.0**	**252G**	**13**
4	46.8	322G	11

(e) **Accuracy *vs.* Efficiency**. Yellow denotes K-Net [95].

Table 5. **DDP ablation experiments** with Swin-T [52] on ADE20K semantic segmentation. We report the performance with 3 sampling steps in (a), (b), (c), and (d). If not specified, the default settings are: the label encoding strategy is class embedding, the scaling factor is set to 0.01, the noise schedule is cosine, and the map decoder has a depth of 6. Default settings are marked in gray.

performance degraded significantly. This is because using a larger scaling factor, more easy cases are reserved with the same time step t. In addition, we found the best scaling factor (*i.e.*, 0.01) for class embedding is typically smaller than analog bits [14] and one-hot (*i.e.*, 0.1).

Noise Schedule. As shown in Table 5c, we compare the effectiveness of the cosine schedule [58] and linear schedule [35] in DDP for semantic segmentation, and find that the model using the cosine schedule achieves notably better performance (47.0 *vs.* 45.1). This is attributed to the cosine schedule's mechanism of simulating the realistic scenario of gradually weakening signal influence, which prompts the model to learn stronger denoising capabilities, in contrast to the simple linear schedule.

Decoder Depth. We study the effect of decoder depth in Table 5d and observe that the map decoder requires a suitable depth. Initially, the model accuracy improves as the depth increases, but eventually decreases. Therefore, we finally adopted a map decoder with 6 blocks, which only has 8.4M parameters. Overall, the map decoder is lightweight and efficient, compared with representative methods K-Net [95] (41.5M) and UperNet [83] (31.5M).

Accuracy *vs.* Efficiency. We show the dynamic trade-off of DDP between accuracy and efficiency in Table 5e. Compared with the representative discriminative method K-Net [95], DDP yields a better mIoU when using only one sampling step, with fewer FLOPs and higher FPS. When adopting three sampling steps, the performance is further boosted to 47.0 mIoU, while maintaining comparable FLOPs and FPS. These results show that DDP can iteratively infer multiple times with reasonable time cost.

5. Conclusion

This paper introduced DDP, a simple, efficient, yet powerful framework for dense visual predictions based on conditional diffusion. It extends the denoising diffusion process into modern perception pipelines, without requiring architectural customization or task-specific design. We demonstrate DDP's effectiveness through state-of-the-art or competitive performance on three representative tasks and six diverse benchmarks. Moreover, it additionally exhibits multiple inference and uncertainty awareness, which contrasts with previous single-step discriminative methods. These results indicate that DDP can serve as an important baseline

for future research in dense prediction tasks. One potential drawback of DDP is its non-negligible additional computational cost for multi-step inference. Besides, while DDP has demonstrated excellent improvement on several benchmark datasets for dense visual prediction tasks, further research is necessary to determine its efficacy in other domains.

Acknowledgement. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 1, 3

[2] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022. 2

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, pages 4009–4018, 2021. 8, 14

[4] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *WACV*, pages 3287–3297, 2023. 6, 7

[5] Walid Bousselham, Guillaume Thibault, Lucas Pagano, Archana Machireddy, Joe Gray, Young Hwan Chang, and Xubo Song. Efficient self-ensemble framework for semantic segmentation. *arXiv preprint arXiv:2111.13280*, 2021. 6

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Biggan: Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 1

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2, 6, 14

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 14

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 3

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 3

[11] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022. 4

[12] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 3, 4

[13] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 4

[14] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. *arXiv preprint arXiv:2210.06366*, 2022. 1, 3, 4, 7, 8

[15] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 4

[16] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *ICLR*, 2023. 3, 4

[17] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *ICLR*, 2023. 5

[18] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 1, 3, 6

[19] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34:17864–17875, 2021. 3, 6

[20] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 13

[21] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 3, 5, 6

[22] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022. 2

[23] Jiequan Cui, Yuhui Yuan, Zhisheng Zhong, Zhuotao Tian, Han Hu, Stephen Lin, and Jiaya Jia. Region rebalance for long-tailed semantic segmentation. *arXiv preprint arXiv:2204.01969*, 2022. 6

[24] Giannis Daras, Yuval Dagan, Alexandros G Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *arXiv preprint arXiv:2302.09057*, 2023. 4

[25] Giannis Daras and Alexandros G Dimakis. Multiresolution textual inversion. *arXiv preprint arXiv:2211.17115*, 2022. 2

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 13

[27] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 2

[28] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014. 7, 14

[29] 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. *arXiv preprint arXiv:1504.06852*, 2015. 1, 3

[30] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018. 1, 8, 14

[31] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013. 1, 2, 3, 7

[32] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753*, 2021. 6

[33] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *ICRA*, pages 87–93. IEEE, 2020. 5

[34] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, pages 2755–2764, 2017. 1

[35] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2, 3, 4, 8

[36] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2

[37] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2

[38] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *ECCV*, pages 581–597, 2020. 14

[39] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 1

[40] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. *arXiv preprint arXiv:2211.06220*, 2022. 3

[41] Pan Ji, Runze Li, Bir Bhanu, and Yi Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *ICCV*, pages 12787–12796, 2021. 14

[42] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1

[43] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, pages 491–507, 2020. 2

[44] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 8, 14

[45] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. Structdepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *ICCV*, pages 12663–12673, 2021. 14

[46] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, pages 8300–8311, 2021. 1

[47] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022. 15

[48] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *arXiv preprint arXiv:2203.14211*, 2022. 7, 8, 14

[49] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 8

[50] Fangjian Lin, Zhanhao Liang, Junjun He, Miao Zheng, Shengwei Tian, and Kai Chen. Structtoken: Rethinking semantic segmentation with structural prior. *arXiv preprint arXiv:2203.12612*, 2022. 6

[51] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[52] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2, 3, 6, 7, 8

[53] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 3, 6

[54] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 2, 6, 7, 14

[55] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020. 5

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5, 13

[57] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1, 2

[58] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171, 2021. 2, 3, 4, 8

[59] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210, 2020. 7

[60] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2

[61] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021. 3, 8, 14

[62] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 7

[63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 3

[65] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[66] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *TPAMI*, 2022. 2

[67] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29:2234–2242, 2016. 1

[68] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J. Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 1, 2, 3, 4, 7, 8

[69] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022. 2

[70] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. 1, 2, 3, 7

[71] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 1, 2, 3

[72] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3, 4

[73] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2

[74] Shuran Song, Jianxiong Xiao, Li Guo, and Xiaogang Yang. Sun rgb-d: A rgb-d scene understanding benchmark suite. *CVPR*, 2015. 1, 2, 3, 7, 14

[75] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 33:12438–12448, 2020. 1, 3

[76] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 3, 6

[77] Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022. 2

[78] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, pages 4604–4612, 2020. 7

[79] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022. 5

[80] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 5

[81] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *MIDL*, pages 1336–1348, 2022. 1, 2, 3

[82] Junde Wu, Huihui Fang, Yu Zhang, Yehui Yang, and Yanwu Xu. Medsegdiff: Medical image segmentation with diffusion probabilistic model. *arXiv preprint arXiv:2211.00611*, 2022. 1, 2, 3

[83] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 2, 3, 6, 8

[84] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, pages 1369–1378, 2017. 1

[85] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34, 2021. 1, 3, 5, 6

[86] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M$\hat{2}$ bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 6, 7

[87] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *ICCV*, pages 16269–16279, 2021. 8, 14

[88] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017. 1

[89] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *NeurIPS*, 34:16494–16507, 2021. 6, 7

[90] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, pages 5684–5693, 2019. 8, 14

[91] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190, 2020. 6

[92] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution vision transformer for dense prediction. *NeurIPS*, 34, 2021. 6

[93] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[94] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 15

[95] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *NeurIPS*, pages 10326–10338, 2021. 2, 3, 6, 8

[96] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 1

[97] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. 3, 6

[98] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, pages 13760–13769, 2022. 7

[99] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 1, 2, 3, 5, 6, 13

[100] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. 2020. 3

Algorithm 3: DDIM Update

```
def alpha_cumprod(t, ns=0.0002, ds=0.00025):
 """cosine noise schedule"""
 n = torch.cos((t + ns) / (1 + ds)
    * math.pi / 2) ** -2
 return -torch.log(n - 1, eps=1e-5)

def ddim(map_t, map_pred, t_now, t_next):
  """
  estimate x at t_next with DDIM update rule.
  """
  α_now = alpha_cumprod(t_now)
  α_next = alpha_cumprod(t_next)
  map_enc = encoding(map_pred)
  map_enc = (sigmoid(map_enc) * 2 - 1) * scale
  eps = 1/sqrt(1-α_now) * (map_t - sqrt(α_now) * map_enc)
  map_next = sqrt(α_next) * x_pred + sqrt(1 - α_now) * eps
  return map_next
```

Algorithm 4: DDP Self-aligned Denoising

```
def train(images, maps):
  """
  images: [b, 3, h, w], maps: [b, 1, h, w]
  """
  img_enc = image_encoder(images)
  map_t = normal(mean=0, std=1)
  map_pred = map_decoder(map_t, img_enc, t=1)
  # encode map_pred
  map_enc = encoding(map_pred.detach())
  map_enc = (sigmoid(map_enc) * 2 - 1) * scale
  # corrupt the map_enc
  t, eps = uniform(0, 1), normal(mean=0, std=1)
  map_crpt = sqrt(alpha_cumprod(t)) * map_enc +
             sqrt(1 - alpha_cumprod(t)) * eps
  # predict
  map_pred = map_decoder(map_crpt, img_enc, t)
  loss = objective_func(map_pred, maps)
  return loss
```

A. Diffusion Model

A.1. Algorithm details

As a supplement to Algorithm 1 and Algorithm 2 described in the main paper, we provide the implementation details in Algorithm 3 for better clarity. Additionally, we introduce the implementation of the "*self-aligned denoising*" procedure in Algorithm 4, used in the last 5K iteration training to address the sampling drift problem (see Section 3.4). We provide an example in Figure 4 to illustrate the gap between the training and inference denoising targets.

A.2. More Discussions

As illustrated in Figure 3a, diffusion models for perceptual tasks tend to reach a saturation point within the first few steps, usually between 3-5 steps, making additional diffusion less advantageous. This is in contrast to the requirements of generative models for image generation, where multiple iterations over many steps (from 10 to 50) are often necessary. Intuitively, in generative tasks such as image generation, the goal is to produce complete and high-quality results by progressively incorporating more information at each time step, thus gradually accumulating and improving the overall result. Therefore, it may take more time steps to reach convergence in order to fully accumulate the necessary information. In perceptual tasks, such as semantic segmentation and object detection, the process from image to label is a gradual reduction of information, and critical information sufficient to make a decision needs to be obtained in only a few steps. Therefore, further diffusion has a limited role in improving the accuracy of predictions, leading to an early peak within three to five steps. In short, the diffusion process in a perception task can make decisions by accumulating the most important information. Therefore, DDP can achieve high accuracy in perception tasks with minimal computational cost.

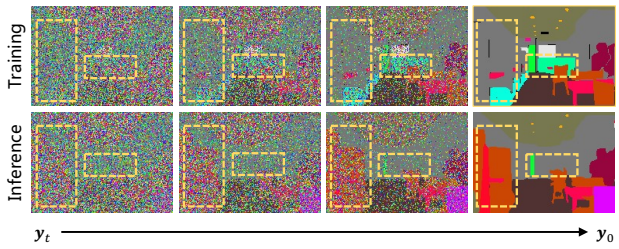


Figure 4. **Sampling drift**. Denoising targets differ from the training process and inference process.

B. Implementation Details

B.1. Semantic Segmentation

ADE20K. We conduct the experiments of ADE20K [99] semantic segmentation based on MMSegmentation [20]. In the training phase, the backbone is initialized with the ImageNet [26] pre-trained weights. We optimize our DDP models using AdamW [56] optimizer with an initial learning rate of 6×10^{-5}, and a weight decay of 0.01. The learning rate is decayed following the polynomial decay schedule with a power of 1.0. Besides, we randomly resize and crop the image to 512×512 for training, and rescale to have a shorter side of 512 pixels during testing. All models are trained for 160k iterations with a batch size of 16 and compared fairly with previous discriminative-based and non-diffusion methods.

Cityscapes. The Cityscape dataset includes 5000 high-resolution images, which contain 2,975 training images, 500 validation images, and 1525 testing samples. The images are captured from 50 different cities in Germany, cov-

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMS $\downarrow$	$\log_{10} \downarrow$
Chen et al.	0.757	0.943	0.984	0.166	0.494	0.071
Yin et al. [90]	0.696	0.912	0.973	0.183	0.541	0.082
BTS [44]	0.740	0.933	0.980	0.172	0.515	0.075
AdaBins [3]	0.771	0.944	0.983	0.159	0.476	0.068
DepthFormer [48]	0.815	0.970	0.993	0.137	0.408	0.059
DDP (step 3)	**0.825**	**0.973**	**0.994**	**0.128**	**0.397**	**0.056**

Table 6. **Depth estimation on the SUN RGB-D dataset.** We report the result of the model trained on the NYU-DepthV2 dataset and tested on the SUN RGB-D dataset without fine-tuning.

ering various environments such as highways, city centers, and suburbs. Similar to ADE20K, during training, we load the ImageNet pre-trained weights and employ the AdamW optimizer. Following common practice, we randomly resize and crop the image to 512×1024 for training, and take the original images of 1024×2048 for testing. We Other hyper-parameters are kept the same as our ADE20K experiments.

B.2. BEV Map Segmentation

nuScenes. We conduct our experiments of BEV map segmentation on nuScenes [7], a large-scale multi-modal dataset for 3D detection and map segmentation. The dataset is split into 700/150/150 scenes for training/validation/testing. It contains data from multiple sensors, including six cameras, one LIDAR, and five radars. For camera inputs, each frame consists of six views of the surrounding environment at the same timestamps. We resize the input views to 256×704 and voxelize the point cloud to 0.1m. Our evaluation metrics align with [54] and report the IoU of 6 background classes, including drivable space, pedestrian crossing, walk-way, stop line, car-parking area, and lane divider, and use the mean IoU as the primary evaluation metric. We adopt the image and LiDAR data augmentation strategies from [8] for training. AdamW is utilized with a weight decay of 0.01 and a learning rate of 5e-5. We take overall 20 training epochs on 8 A100 GPUs with a batch size of 32. Other training settings are kept the same as [54] for fair comparisons.

B.3. Depth Estimation

KITTI. The KITTI depth estimation dataset is a widely used benchmark dataset for monocular depth estimation with a depth range from 0-80m. The stereo images of the dataset have a resolution of 1242×375, while the corresponding GT depth map has a low density of 3.75% to 5.0%. Following the standard Eigen training/testing split [28], we use around 26K left view images for training and 697 frames for testing. We incorporate the DDP model into the codebase developed by [48] for KITTI depth estimation experiments. We excluded the discrete label encoding module as the task requires continuous value regression All experimental settings are the same as [48] for a fair comparison.

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$
StructDepth [45]	0.817	0.955	0.988	0.140	0.534	0.060
MonoIndoor [41]	0.823	0.958	0.989	0.134	0.526	-
DORN [30]	0.828	0.965	0.992	0.115	0.509	0.051
BTS [44]	0.885	0.978	0.994	0.110	0.392	0.047
DAV [38]	0.882	0.980	0.996	0.108	0.412	-
TransDepth [87]	0.900	0.983	0.996	0.106	0.365	0.045
DPT-Hybrid [61]	0.904	0.988	**0.998**	0.110	0.357	0.045
AdaBins [3]	0.903	0.984	0.997	0.103	0.364	0.044
DepthFormer [48]	**0.921**	0.989	**0.998**	0.096	0.339	0.041
DDP (step 3)	**0.921**	**0.990**	**0.998**	**0.094**	**0.329**	**0.040**

Table 7. **Depth estimation on the NYU-DepthV2 val set.** We report the performance of DDP with 3 diffusion steps. The best and second-best results are bolded or underlined, respectively. $\downarrow$ means lower is better, and $\uparrow$ means higher is better.

NYU-DepthV2. The NYU-DepthV2 is an indoor scene dataset that consists of RGB and depth images captured at a resolution of 640×480 pixels. The dataset contains over 1,449 pairs of aligned indoor scenes, captured from 464 different indoor areas. We train DDP using image pairs with a resolution of 320×240 and with varying depths up to approximately 10 meters. Following previous work, we evaluate the results on the predefined center cropping by [28]. To be fair, all experimental configurations were aligned with the previous method [48].

SUN RGB-D. We use this dataset [74] to evaluate generalization. To be specific, we assess the performance of our NYU pre-trained models on the official test set, which includes 5,050 images, without any additional fine-tuning. The maximum depth is restricted to 10 meters. Please note that this dataset is solely intended for evaluation purposes and is not utilized for training.

C. Experimental Results

In Table 7, we provide the depth estimation performance of DDP on the NYU-V2 dataset, in addition, in Table 6, we provide the generalization performance results of DDP on the SUN-RGBD dataset.

D. Visualization

Figure 5 and Figure 6 visualize the "multiple inference" property of DDP on the validation sets of Cityscapes and ADE20K, respectively. These inference trajectories show that DDP can enhance its performance continuously and produce smoother segmentation maps by using more sampling steps. Figure 7 presents the BEV map segmentation results of DDP (step 3) with the ground truths and multi-view images. Figure 8 and Figure 9 compare the generated depth estimation results of DDP (step 3) with the ground truths on the validation sets of KITTI and NYU-DepthV2, respectively. These results indicate that our method can be easily generalized to most dense prediction tasks.

E. More Applications

E.1. Combine DDP with ControlNet

Setup. It has been found that compared to the previous single-shot model, DDP can achieve more continuous and semantic consistency prediction results. To demonstrate the benefits of this pixel clustering property, we combined DDP with the recently popular segmentation mask condition generation model: ControlNet. We followed the official implementation of ControlNet for all hyperparameters, including input resolution and DDIM sampling steps.

Implementation ControlNet [94] improves upon the original Stable Diffusion (SD) model by adding extra conditions, which is done by incorporating a conditioning network. In the mask-conditional ControlNet, the map generated by the segmentation model is used as input for image synthesis. The original segmentation model was adopted from Uniformaer-S [47] with UperNetHead, which has 52M parameters and achieves 47.6 mIoU (ss) on the ADE20K dataset. To make a fair comparison, we replaced the original segmentation model in the mask-conditional ControlNet with DDP using the Swin-T backbone, which has 40M parameters and achieves 47.0 mIoU (ss) on the ADE20K dataset. Note that all results were obtained with the default prompt.

Results We select images from the PEXEL website https://www.pexels.com/ for testing in different scenarios. The results from the original ControlNet and the combination of DDP with ControlNet are shown in Figure 10. ControlNet is designed to achieve fine-grained, controllable image generation, our experiments show that DDP can produce more consistent results and has advantages in various scenarios. Moreover, when combined with DDP, ControlNet produces visually satisfying and well-composed results, surpassing those of the original ControlNet. Our experimental results suggest that DDP has great potential to improve cooperation with other types of foundation models.

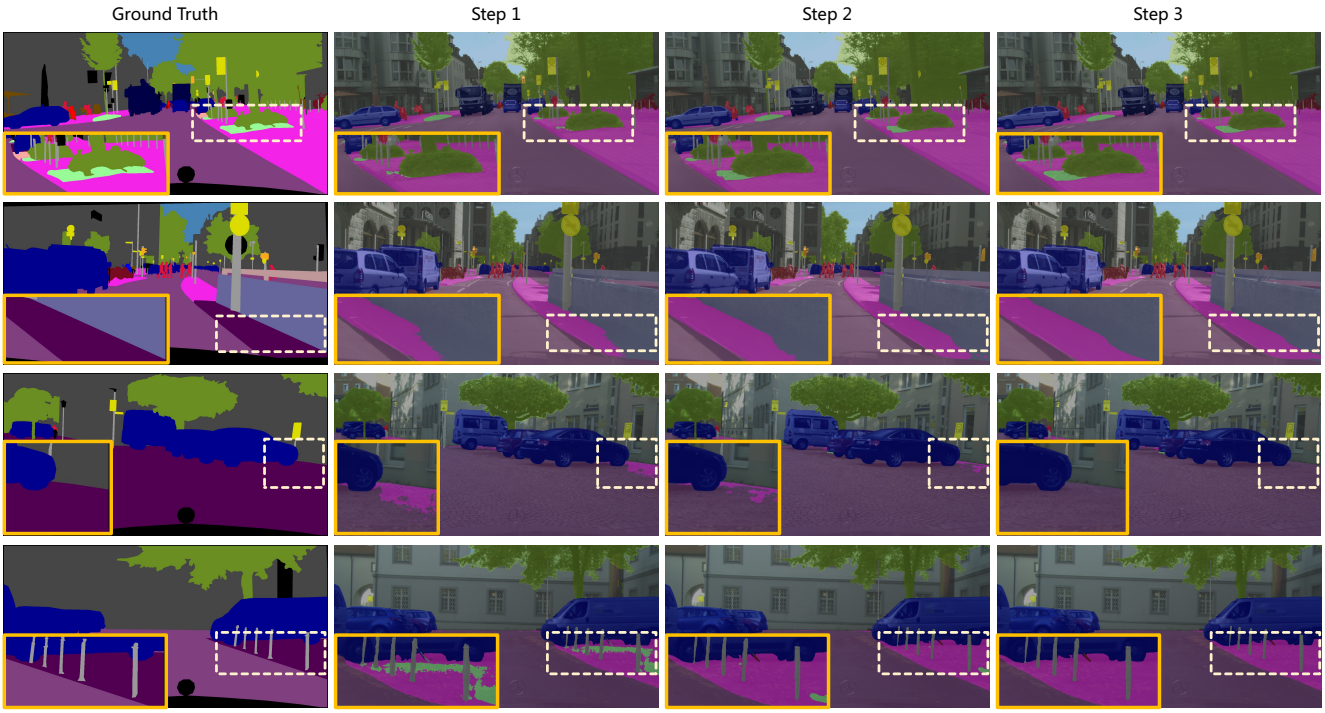


Figure 5. **Visualization of multiple inference on Cityscapes val set.**

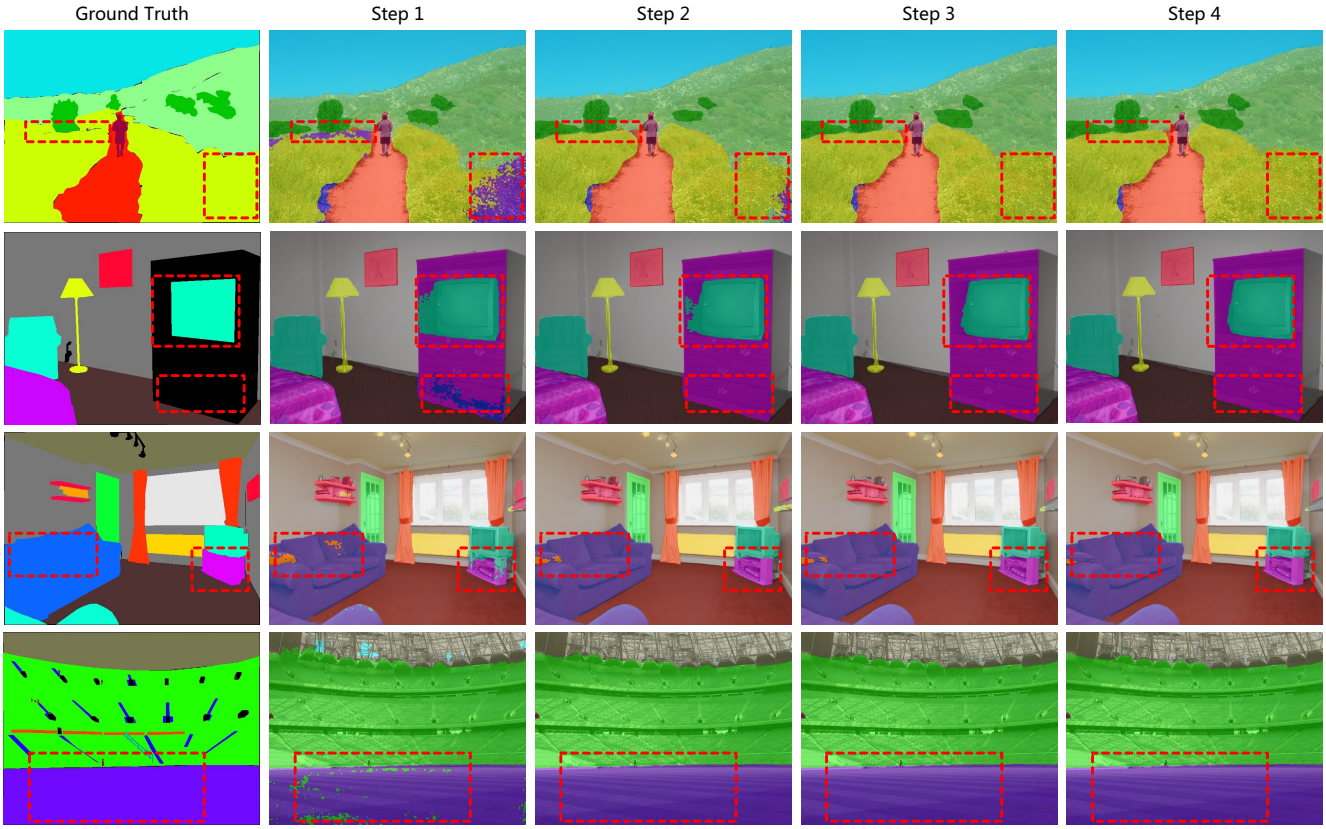


Figure 6. **Visualization of multiple inference on ADE20K val set.**

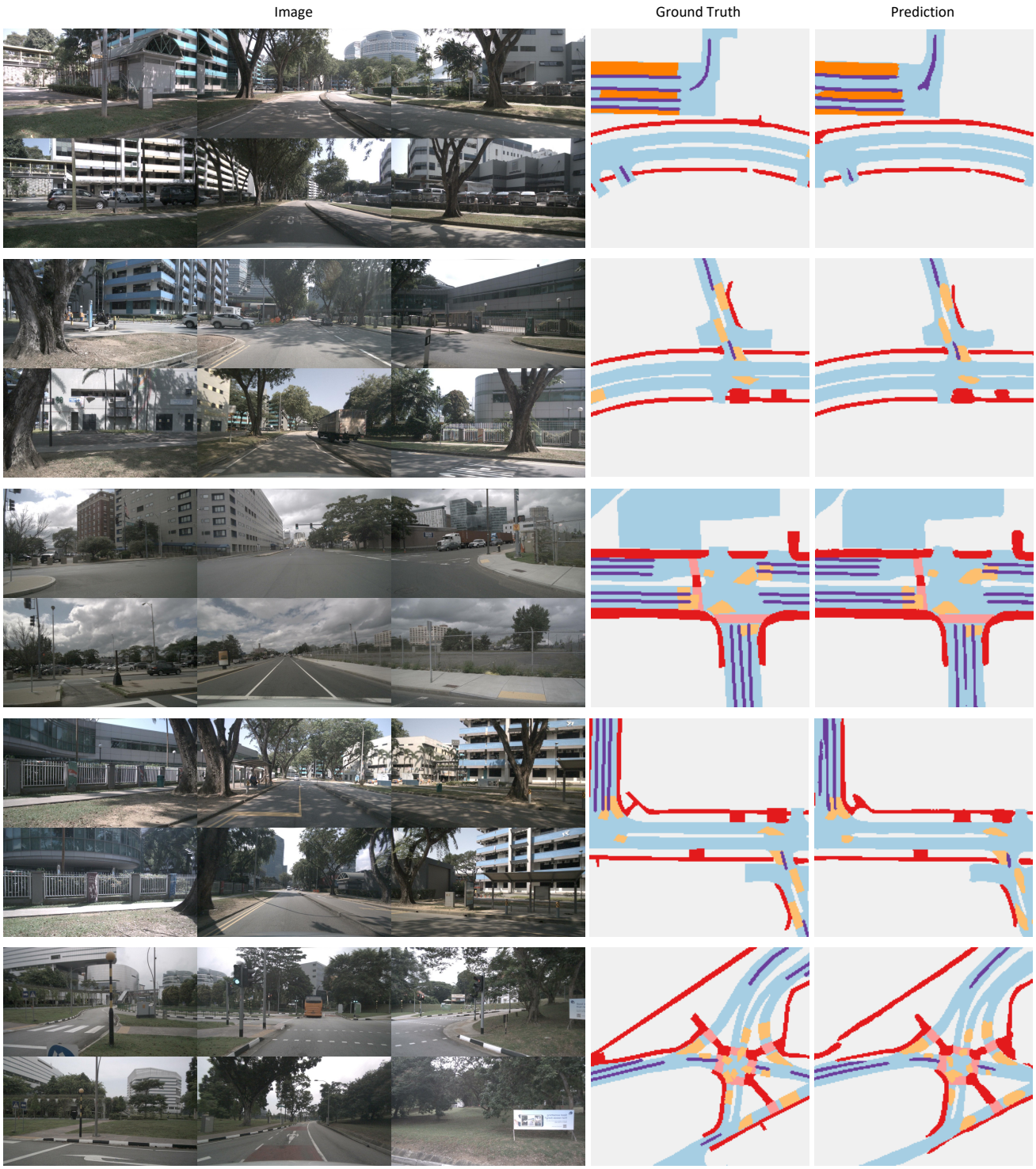


Figure 7. **Visualization of predicted BEV map segmentation results on nuScenes val set.**

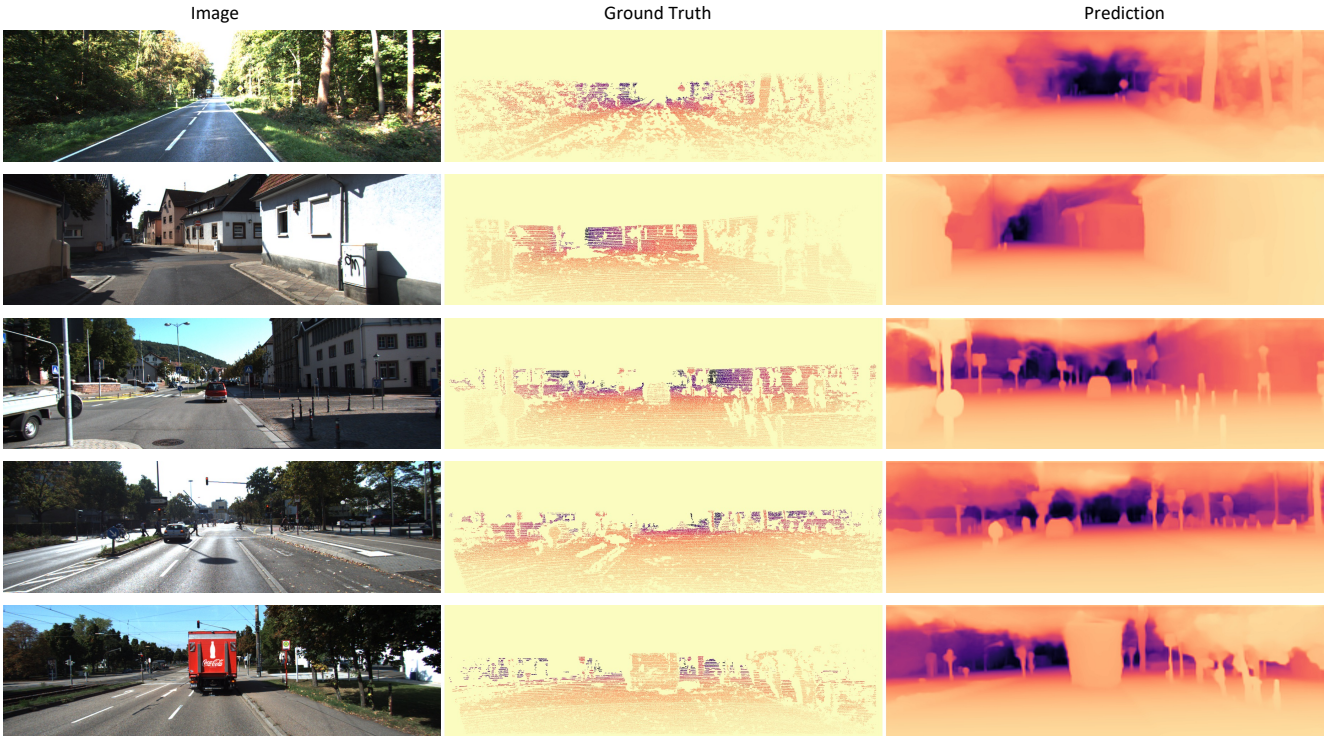


Figure 8. **Visualization of predicted depth estimation results on KITTI val set.**

Figure 9. **Visualization of predicted depth estimation results on NYU-DepthV2 val set.**

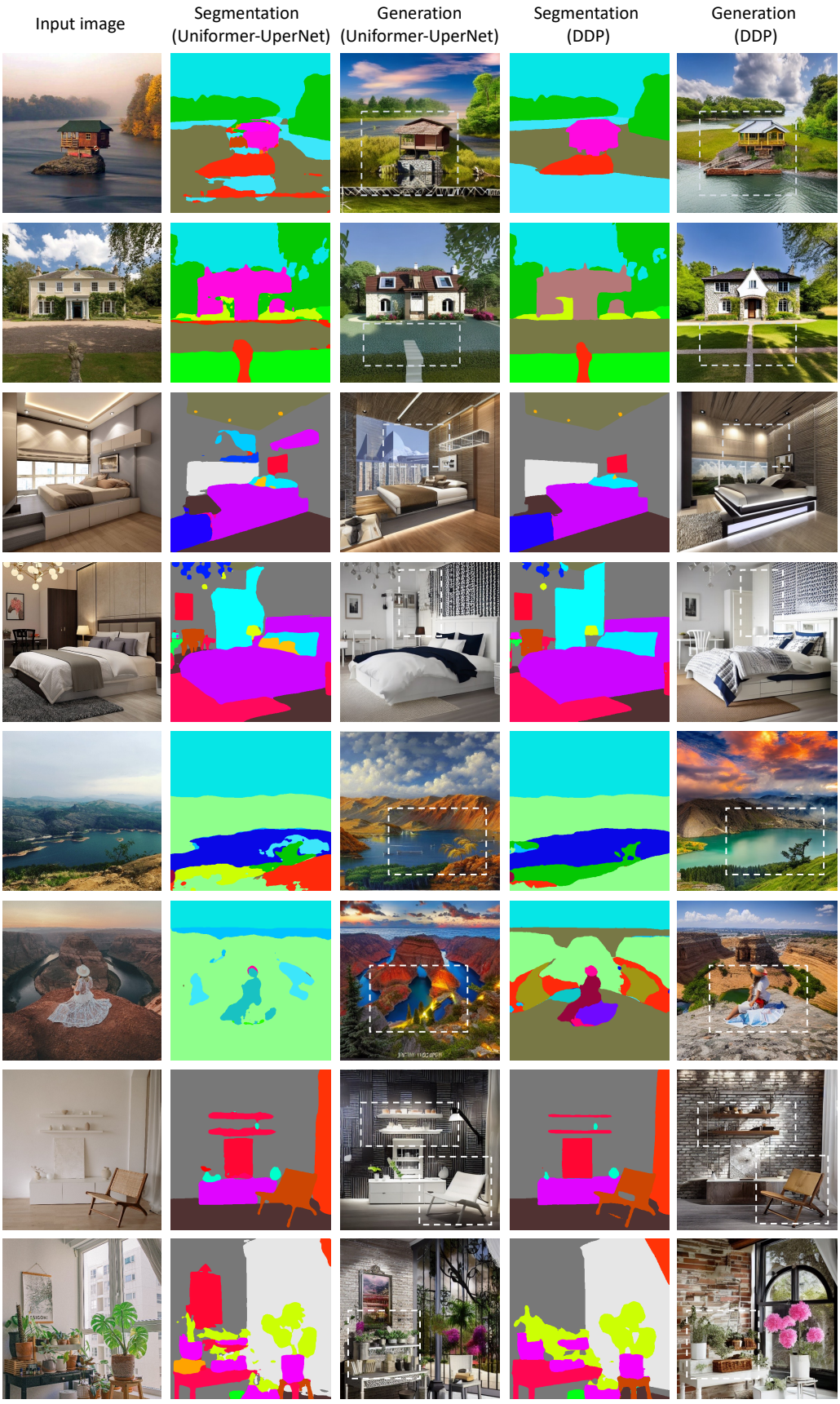


Figure 10. **Control Stable Diffusion with Semantic Map**, the Uniformer-UnperNet, and DDP segmentation models are used to predict segmentation maps as condition input. All results were achieved using the default prompt.