

MYGO: Multi-Layer Depth and Occlusion Encoding for Geometry-Aware ControlNet

Yanhao Li
New York University Shanghai
yl111912@nyu.edu

Zhuoyan Yu
New York University Shanghai
zy3077@nyu.edu

Abstract

*Depth-conditioned ControlNets are widely used to control diffusion models with scene geometry, but the depth map is typically treated as a black-box condition and evaluated only through image quality. In this project, we ask a simple question: does the choice of depth encoding change how a depth ControlNet behaves? We propose **MYGO** (Multi-laYer Geometry Occlusion ControlNet), which replaces the standard single-channel depth input with an explicit multi-layer representation consisting of raw depth, ordinal depth layers, and an occlusion-boundary map. To measure geometric fidelity, we run a frozen monocular depth estimator (DepthAnything) on generated images and compare its predictions with the conditioning depth, reporting SILog, AbsRel, δ_1 , occlusion F1, and FID on a COCO-based dataset.*

Under a fixed 20k-step training budget, MYGO consistently improves FID and, over most training checkpoints, achieves comparable or slightly better geometry metrics and sharper occlusion edges than a 1-channel depth baseline. However, we also observe a new failure mode: with longer training, the multi-layer model can over-smooth textures and degrade depth structure. These results suggest that explicit multi-layer depth and occlusion encoding can make depth ControlNets more geometry-aware, but also introduce new stability–efficiency trade-offs that future work should explore.

Project code is available at: <https://github.com/JosiahWayne/MYGO>

1. Introduction

Controlling image generation with geometric signals such as depth has become a key ingredient for diffusion-based image synthesis. In particular, depth-conditioned ControlNets built on Stable Diffusion enable users to preserve scene layout while freely editing appearance [5, 7].

Most existing systems treat the depth map as a single-channel, black-box condition: a depth image is fed into the

ControlNet, and performance is evaluated only by visual quality or user preference. However, it is still unclear how the ControlNet actually uses this depth map, and whether alternative depth encodings could make the model more geometry-aware.

In this project we take a first step toward answering this question. We propose **MYGO** (Multi-laYer Geometry Occlusion ControlNet), which replaces the standard single-channel depth input with an explicit multi-layer representation. Concretely, MYGO uses raw depth, a set of ordinal depth layers, and an occlusion-boundary map as conditioning, exposing depth ordering and boundaries directly to the ControlNet instead of relying on implicit learning. This is inspired by ordinal depth encodings used in monocular depth estimation [1], but adapted to depth-guided diffusion.

To evaluate geometry, we use a frozen monocular depth estimator to predict depth from generated images and compare it with the conditioning depth, using a model such as DepthAnything [6]. This allows us to measure not only visual realism but also geometry-aware metrics such as SILog, AbsRel, δ_1 , and occlusion F1, alongside FID.

Our experiments on a subset of COCO with a fixed 20k-step training budget show that MYGO consistently improves FID and, over most training checkpoints, yields comparable or slightly better geometry metrics and sharper occlusion edges than a 1-channel depth baseline. At the same time, we observe a new failure mode: with overly long training, the multi-layer model can over-smooth textures and degrade depth structure, suggesting an interesting trade-off between stronger geometric supervision and training stability.

2. Related Work

2.1. Diffusion-based Image Editing and Control

Text-to-image diffusion models such as Stable Diffusion [5] enable high-fidelity image synthesis from text prompts. To gain spatial control, ControlNet [7] augments the U-Net backbone with condition-specific branches for edges, depth, pose, and other structural inputs. Most depth-conditioned

ControlNet variants simply take a single-channel depth map as input and evaluate performance primarily by visual quality.

2.2. Monocular Depth Estimation and Ordinal Representations

Monocular depth estimation has evolved from direct regression to discretized or ordinal formulations. Deep ordinal regression networks such as DORN [1] encode depth as ordered bins, which improves robustness and boundary sharpness compared to pure regression. These methods focus on predicting depth itself, rather than using depth as a control signal for generative models.

2.3. Depth-guided Diffusion Models

Depth-guided diffusion models combine depth maps with text prompts to preserve scene layout while allowing appearance editing, often via ControlNet-style architectures [7]. Some systems refine or normalize depth maps before feeding them into the model, but they typically still operate on a single depth channel and report only image-level metrics. To our knowledge, there is little systematic study of how alternative depth encodings (e.g., multi-layer or occlusion-aware representations) affect the behavior of depth ControlNets.

2.4. Geometry-aware Evaluation of Generative Models

Evaluation of generative models often relies on image fidelity metrics such as FID [2], which do not directly measure geometric consistency. Recent monocular depth estimators like DepthAnything [6] provide strong, generalizable depth predictions and can serve as a geometry proxy. In this work, we use a frozen DepthAnything model to estimate depth from generated images and compare it with the conditioning depth, enabling geometry-aware metrics such as SILog, AbsRel, δ_1 , and occlusion F1 in addition to FID.

3. Method

Our goal is to study how the choice of depth encoding affects the behavior of a depth-conditioned ControlNet. Given a text prompt y and a conditioning depth map $D \in [0, 1]^{H \times W}$, we replace the standard single-channel depth input with an explicit multi-layer representation that exposes depth ordering and occlusion boundaries. We then plug this representation into a Stable-Diffusion-style ControlNet and optionally add geometry-aware losses during training.

3.1. Depth-conditioned Diffusion with ControlNet

We build on a standard latent diffusion pipeline with a depth ControlNet. Let x denote the target RGB image and z its latent representation produced by the VAE encoder. The diffusion U-Net predicts noise $\epsilon_\theta(z_t, y, C(D), t)$ at timestep

t , conditioned on the text embedding y and a depth-based control signal $C(D)$.

In the *baseline* setting, the control signal is a single-channel depth map $D \in [0, 1]^{H \times W}$. In our method, MYGO, we instead construct a multi-channel tensor $C(D) \in \mathbb{R}^{(1+K+1) \times H \times W}$ that encodes raw depth, K ordinal depth layers, and an occlusion-boundary map, as described in Sec. 3.2. To accommodate this higher-dimensional condition, we patch only the input convolution inside `controlnet_cond_embedding`, replacing its `Conv2d` layer with a wider variant whose number of input channels C_{in} matches the dimensionality of the depth condition ($C_{\text{in}} = 1$ for the baseline and $C_{\text{in}} = 18$ for MYGO). The latent input convolution (on the 4-channel latent z_t) and all subsequent U-Net and ControlNet blocks are left unchanged, so any performance differences can be attributed to the depth encoding rather than changes in model capacity.

During training we use the standard noise prediction loss L_{diff} from the original diffusion model, optionally augmented with geometry-aware terms described in Sec. 3.3.

3.2. Multi-layer Depth and Occlusion Encoding

Given a normalized depth map $D \in [0, 1]^{H \times W}$, MYGO constructs a multi-layer representation

$$C(D) = [D, C^{\text{ord}}(D), B(D)] \in \mathbb{R}^{(1+K+1) \times H \times W},$$

where $C^{\text{ord}}(D)$ is a stack of K ordinal depth layers and $B(D)$ is an occlusion-boundary map.

Ordinal depth layers. We aim to expose coarse-to-fine depth ordering while avoiding channels that are almost always all-zero or all-one. Instead of regressing a single scalar depth, we discretize depth into K ordered “slices” using logarithmically spaced thresholds in depth.

Concretely, we first normalize D into $[0, 1]$ and define K thresholds $\{\tau_k\}_{k=1}^K$ that are uniformly spaced in log-space between 10^{-3} and 1. We then linearly rescale these thresholds back to $[0, 1]$ and construct binary ordinal channels by

$$C_k^{\text{ord}}(x) = \mathbf{1}[D(x) \geq \tau_k].$$

This concentrates more levels near the camera (where relative depth changes more quickly) while still covering the full depth range, and empirically reduces completely saturated channels.

Occlusion-boundary map. Depth discontinuities often correspond to occlusion boundaries, which are crucial for correctly ordering foreground and background. We estimate an occlusion map $B(D) \in [0, 1]^{H \times W}$ by applying

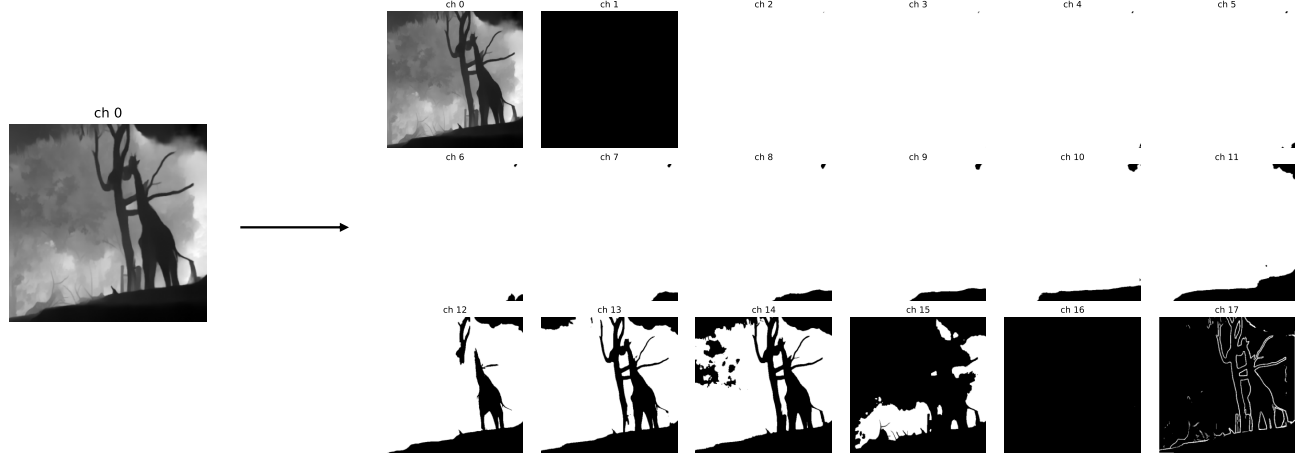


Figure 1. Left: baseline depth map. Right: MYGO’s 18 channels (raw depth, 16 ordinal layers, occlusion boundary).

Sobel filters in x and y , computing gradient magnitude, and normalizing it per-image:

$$g_x = \text{Sobel}_x(D), \quad g_y = \text{Sobel}_y(D), \quad m = \sqrt{g_x^2 + g_y^2}.$$

We then normalize m by its spatial maximum and threshold to obtain a binary boundary map:

$$\tilde{m} = \frac{m}{\max_x m(x) + \epsilon}, \quad B(D)(x) = \mathbf{1}[\tilde{m}(x) > \tau_{\text{ob}}],$$

with a fixed threshold τ_{ob} . This adds an explicit channel that highlights occlusion contours independently of the raw depth values.

Baseline vs. MYGO. The baseline ControlNet receives only the raw depth D , whereas MYGO receives $C(D)$ with $1 + K + 1$ channels. All other architectural components of the diffusion model and ControlNet are kept identical, isolating the effect of the depth encoding.

3.3. Geometry-aware Objectives

Beyond the standard diffusion loss L_{diff} , we consider geometry-aware losses that compare the conditioning depth D with depth recovered from the generated image. We use a frozen monocular estimator f_{DA} (DepthAnything V3) [3] to predict depth $\hat{D}$ from the generated image $\hat{x}$, and align it to the scale of D using a least-squares scale-and-shift fit:

$$D^* = s \hat{D} + t$$

with (s, t) chosen to minimize $\|s \hat{D} + t - D\|_2^2$.

Given D^* and D , we can define a scale-invariant log loss L_{silog} , as well as an occlusion-boundary loss. Let $B(D)$ and $B(D^*)$ denote the occlusion maps computed as above. We

Algorithm 1 MYGO condition construction from a depth map

Input: Normalized depth map $D \in [0, 1]^{H \times W}$, number of ordinal layers K

Output: Condition tensor $C(D) \in \mathbb{R}^{(1+K+1) \times H \times W}$

1. Compute K log-spaced thresholds $\{\tau_k\}_{k=1}^K$ between 10^{-3} and 1, and linearly rescale them to $[0, 1]$ to match the normalized depth range.
2. For each pixel x and each level k , define $C_k^{\text{ord}}(x) \leftarrow \mathbf{1}[D(x) \geq \tau_k]$.
3. Compute Sobel gradients g_x, g_y on D and gradient magnitude $m(x) = \sqrt{g_x(x)^2 + g_y(x)^2}$.
4. Normalize m by its spatial maximum to obtain $\tilde{m}(x)$ and threshold it to obtain the occlusion map $B(D)(x) = \mathbf{1}[\tilde{m}(x) > \tau_{\text{ob}}]$.
5. Concatenate channels: $C(D) \leftarrow [D, C^{\text{ord}}(D), B(D)]$.

treat boundary prediction as a binary detection problem and define

$$\text{F1}_{\text{occ}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}},$$

where TP, FP, and FN are computed over pixels using $B(D)$ as ground truth. The corresponding loss is

$$L_{\text{ob}} = 1 - \text{F1}_{\text{occ}}.$$

The overall training objective is

$$L = L_{\text{diff}} + \lambda_{\text{geo}} L_{\text{silog}} + \lambda_{\text{ob}} L_{\text{ob}},$$

where we set λ_{geo} and λ_{ob} to control the strength of geometry and occlusion supervision. In our main experiments, we focus on the effect of the multi-layer representation itself and primarily use L_{diff} (baseline) or $L_{\text{diff}} + \lambda_{\text{ob}} L_{\text{ob}}$ (MYGO-OB).

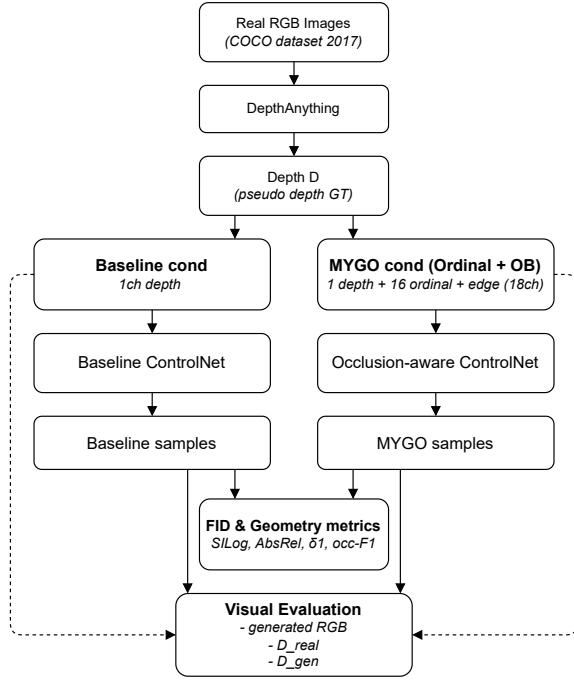


Figure 2. Overview of the training and evaluation pipeline.

4. Experiments

We empirically study how re-encoding a single depth map into multi-layer geometry and occlusion channels affects a depth-conditioned ControlNet. Unless otherwise noted, we compare a single-channel depth baseline against our MYGO variant under a matched training budget of 20k gradient steps.

4.1. Experimental Setup

Dataset and preprocessing. We build our training and evaluation data from a 50k-image subset of MS COCO 2017 [4] images containing diverse indoor and outdoor scenes. For each RGB image x we compute a monocular depth map D using a pretrained DepthAnything V3 model [3] and normalize it to $[0, 1]$. We treat this pseudo depth as both the control signal and the reference for evaluating geometric fidelity. The RGB images are resized to 512×512 and scaled to $[-1, 1]$, and the corresponding depth maps are resized to the same resolution. For MYGO, depth maps are further converted into multi-layer conditions using the procedure in Sec. 3.2.

Models and baselines. All experiments use Stable Diffusion v1.5 as the base latent diffusion model with a depth ControlNet backbone [5, 7]. The *Baseline* model re-

ceives a single-channel depth map $D \in [0, 1]^{H \times W}$. Our *MYGO* model replaces this with the multi-layer representation $C(D) = [D, C^{\text{ord}}(D), B(D)]$ described in Sec. 3.2, consisting of raw depth, K ordinal channels, and a binary occlusion-boundary channel. All other architectural components, including the text encoder, U-Net backbone, and VAE, are kept identical. This isolates the effect of the depth encoding.

Training details. We train both models with the same configuration: a global batch size of 32, learning rate 5×10^{-5} , mixed-precision (fp16) training, and the standard noise-prediction objective L_{diff} . For MYGO we optionally enable the geometry-aware losses defined in Sec. 3.3. In our main runs we set $\lambda_{\text{ob}} > 0$ and $\lambda_{\text{geo}} = 0$, focusing on occlusion-boundary supervision while keeping the depth regression purely implicit. We train for 20k gradient steps and save checkpoints at regular intervals. For fair comparison, we use the same random seed, the same shuffled data order, and the same fixed visualization batch across models.

Evaluation protocol. At evaluation time, we sample images with 15 diffusion steps and a classifier-free guidance scale of 7.5. For each checkpoint we generate a fixed set of images using held-out COCO captions and their associated depth maps. We then evaluate both visual realism and geometric consistency.

Geometry metrics. Given a generated image $\hat{x}$ and its conditioning depth D , we run the frozen DepthAnything V3 estimator [3] to obtain a predicted depth map $\hat{D}$. We align $\hat{D}$ to D by a global scale and shift and report standard depth metrics as in prior monocular depth estimation work [1, 6]: *SILog* (scale-invariant log RMSE), *AbsRel* (mean absolute relative error), and δ_1 (fraction of pixels whose relative error is below 1.25).

To capture occlusion quality, we derive binary occlusion maps from D and $\hat{D}$ using the Sobel-based operator introduced in Sec. 3.2 and compute an occlusion F1 score between the two maps.

All depth-related metrics are computed against pseudo-depth references produced by a frozen DepthAnything model and should therefore be interpreted as measures of consistency with this estimator rather than absolute geometric accuracy. Both the conditioning depth D and the predicted depth $\hat{D}$ are normalized to $[0, 1]$ and aligned via a global scale-shift fit prior to evaluation. This normalization, combined with the lack of metric ground-truth, can lead to atypical absolute values for relative error metrics such as *AbsRel* and δ_1 : in particular, relative errors are sensitive to near-zero reference depths, which can amplify error magnitudes even for visually plausible predictions. Consequently, we focus on relative comparisons under this fixed evaluation

protocol and treat these metrics as indicators of geometric consistency rather than absolute depth quality. Their absolute magnitudes are not directly comparable to standard benchmark reports on datasets with metric depth.

Image quality. To measure image-level realism, we compute Fréchet Inception Distance (FID) between generated images and the COCO reference set, using an ImageNet-pretrained Inception network. Unless otherwise noted, we report FID under the same sampling configuration and with 1k generated samples for all models.

5. Results

We now present quantitative and qualitative results comparing the single-channel depth baseline and our MYGO representation under the experimental setup from Sec. 4. Unless otherwise noted, both models are trained on the COCO subset for 20k steps with the same text prompts, depth conditions, and optimization hyperparameters.

5.1. Quantitative Results

We first report geometry-aware metrics computed by running the frozen DepthAnything estimator on generated images and comparing the predicted depth $\hat{D}$ (aligned by scale and shift) against the conditioning depth D , as described in Sec. 4.1. We measure SILog, AbsRel, and δ_1 for depth accuracy, and an occlusion F1 score derived from binary occlusion maps. Image-level realism is measured by Fréchet Inception Distance (FID) between generated images and the COCO reference set.

Because we are interested both in mid-training behavior and in the final checkpoint under a fixed training budget, we report metrics at two representative timesteps: a mid-training checkpoint at 14k steps, where the geometry gap is most visible, and the final 20k-step checkpoint. Table 1 summarizes the comparison.

Model	Steps	FID ↓	Occ F1 ↑	δ_1 ↑	SILog ↓	AbsRel ↓
Baseline	14k	85.4	0.396	0.313	3.78	3.18
MYGO	14k	68.6	0.414	0.342	4.14	3.01
Baseline	20k	84.0	0.408	0.319	4.02	3.17
MYGO	20k	74.3	0.404	0.319	3.42	3.40

Table 1. Main comparison between the single-channel depth baseline and MYGO at a mid-training checkpoint (14k steps) and the final checkpoint (20k steps) on the COCO subset. MYGO consistently improves FID and often improves δ_1 and occlusion F1 at mid-training, while SILog and AbsRel exhibit mixed behavior and the geometry gains shrink at the final checkpoint.

At 14k steps, MYGO achieves a substantially lower FID than the baseline (about a 20% relative reduction) and clearly higher δ_1 and occlusion F1, indicating better visual

quality and stronger depth/occlusion consistency. However, SILog and AbsRel are worse for MYGO at this checkpoint, suggesting that not all depth-error metrics benefit equally from the multi-layer encoding.

By the final 20k-step checkpoint, the picture becomes more mixed. MYGO still improves FID and SILog over the baseline (roughly 10–15% relative reductions), but the gains in δ_1 and occlusion F1 largely disappear: these metrics are nearly tied between the two models. AbsRel remains slightly worse for MYGO. Overall, MYGO delivers consistently better FID across training, tends to improve δ_1 and occlusion F1 at many checkpoints, but does not uniformly dominate the baseline on all depth metrics.

5.2. Training Dynamics

To better understand these effects, we analyze how metrics evolve over training. Figure 4 plots FID and the DepthAnything-based geometry metrics as a function of training steps for both the baseline and MYGO.

After roughly 5k steps, FID is consistently lower for MYGO than for the single-channel baseline, and this gap persists throughout training. For occlusion F1, MYGO is typically above the baseline at mid-training, but the curves converge towards the end of training, which matches the small difference we observe at 20k steps in Table 1.

Other depth-oriented metrics such as SILog, AbsRel, and δ_1 behave more subtly. In our logs, δ_1 is often higher for MYGO at many checkpoints, whereas SILog and AbsRel frequently cross between the two models and do not reveal a clear winner. This agrees with the mixed quantitative picture in Table 1: the multi-layer encoding helps in terms of some structural measures of geometry, but its impact on aggregate depth errors is not uniformly positive. Additional plots for SILog, AbsRel, and δ_1 are provided in Appendix A.

Extending training beyond 20k steps reveals an additional failure mode for MYGO. When we continue training to around 40k steps, geometry metrics begin to plateau or even degrade, and FID also stops improving, as shown in Figure 5. As discussed below, the generated images at these late checkpoints become noticeably over-smoothed, suggesting that stronger depth supervision and a richer condition space can make the model more sample-efficient early on, but may also amplify overfitting or collapse effects with prolonged training.

This suggests that stronger depth supervision and a richer condition space can make the model more sample-efficient early on, but may also amplify collapse effects in longer training.

5.3. Qualitative Analysis

We qualitatively compare generations from the baseline and MYGO using the same prompts and depth maps. Figure 3

Baseline



MYGO



Figure 3. **Qualitative comparison** between the baseline and MYGO. Each column corresponds to the same prompt. Rows show (top) generated images, (middle) conditioning depth, and (bottom) depth predicted from the generated image. MYGO produces more coherent structure and better aligns with the intended depth ordering.

shows examples at 13k steps. For each setting, we visualize: (1) the generated RGB image, (2) the conditioning depth, and (3) DepthAnything’s predicted depth.

For the baseline, global near/far relationships are often preserved, but fine-grained geometry is weak: foreground objects (e.g., people, umbrellas, cars) tend to bleed into the background, and occlusion contours are blurred. Depth predictions from the frozen estimator exhibit flattened or noisy regions around object boundaries, indicating that the generated images lack sharp geometric cues.

In contrast, MYGO generations show clearer layered structure. Occluding boundaries are sharper, silhouettes are more stable, and background planes (e.g., walls, build-

ings, sky) are better separated from foreground objects. DepthAnything’s predictions align more closely with the input depth, especially around occlusion edges, even when scalar depth metrics at the 20k-step checkpoint differ only modestly. This suggests that the multi-layer encoding is particularly helpful for capturing relative depth and boundary information that may not be fully reflected in the aggregate depth-error metrics.

At very late training checkpoints, MYGO sometimes produces overly smooth textures and “washed out” local details, and the predicted depth becomes less reliable. This collapse behavior is largely absent in the 1-channel baseline and highlights a stability–expressivity trade-off introduced

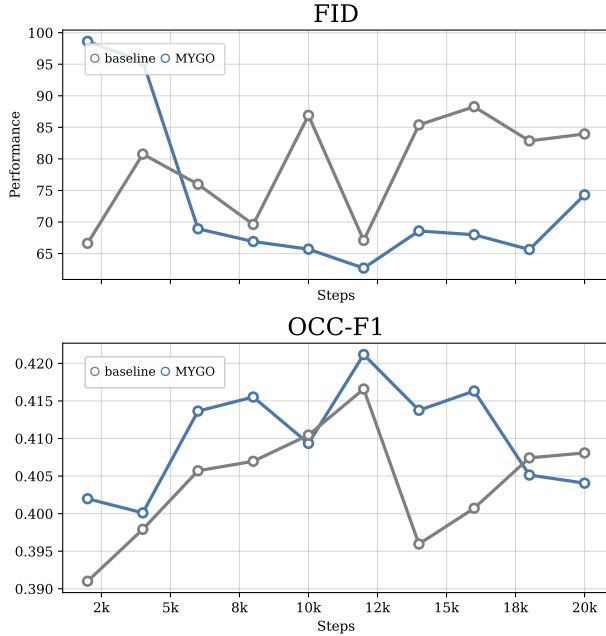


Figure 4. Training dynamics of FID and occlusion F1 for the single-channel baseline and MYGO. After roughly 5k steps, MYGO consistently achieves lower FID, and tends to obtain higher occlusion F1 during mid-training, while the gap shrinks towards the final checkpoint.

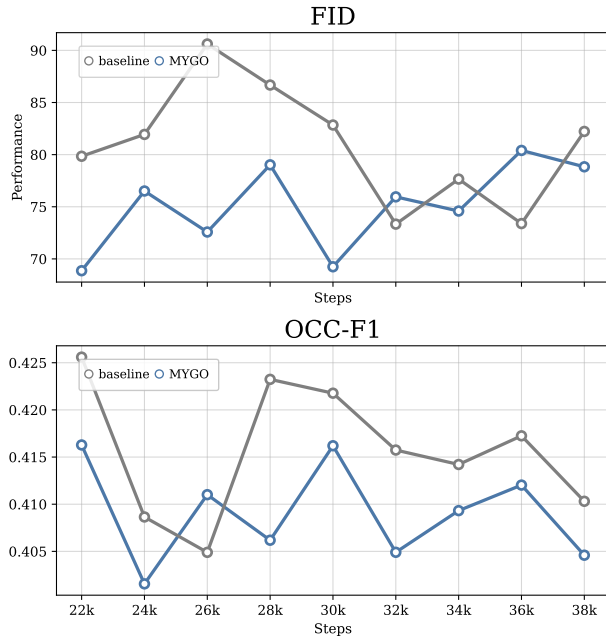


Figure 5. Same metrics with training steps extended to 38k steps.

by the multi-layer encoding: it can improve geometry and realism early in training, but requires careful regularization

or early stopping to avoid degeneration.

6. Conclusion

In this project, we asked a simple question: does the choice of depth encoding change how a depth ControlNet behaves? More specifically, if we provide more explicit geometric guidance, can the ControlNet learn to generate images with better quality?

We proposed MYGO, a multi-layer depth and occlusion representation that replaces the standard single-channel depth map with raw depth, ordinal depth layers, and an explicit occlusion-boundary map.

Using a frozen DepthAnything estimator to probe geometry, we showed that under a fixed 20k-step training budget, MYGO consistently improves FID and, across many checkpoints, achieves comparable or slightly better geometry metrics and sharper occlusion boundaries than a 1-channel baseline.

Qualitative results further indicate that the multi-layer representation encourages clearer foreground-background separation and more faithful depth ordering in generated images.

At the same time, we observed a new failure mode: when trained for too many steps, MYGO can over-smooth textures and degrade depth structure. This suggests that stronger geometry-aware conditioning can improve sample efficiency but may also make the model more sensitive to training length and optimization.

Overall, our results highlight that depth encoding is not a neutral design choice for ControlNet-style conditioning. Explicitly exposing ordinal layers and occlusion boundaries can make depth guidance more geometry-aware, but introduces new stability and compute trade-offs. Future work could explore alternative ordinal schemes (e.g., different K or learned binning), more robust occlusion losses, and extensions to other control modalities such as surface normals or multi-view geometry.

Author Contributions

Yanhao Li and Zhuoyan Yu jointly contributed to the design of the MYGO depth encoding and the initial setup of the MYGO codebase. Yanhao Li primarily ran experiments and analyzed results, while Zhuoyan Yu assisted with maintaining codebase and polishing the final report.

References

- [1] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4

- [2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 2
- [3] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 3, 4
- [4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 4
- [5] 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 (CVPR)*, pages 10684–10695, 2022. 1, 4
- [6] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 4
- [7] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023. 1, 2, 4

A. Additional Metric Curves

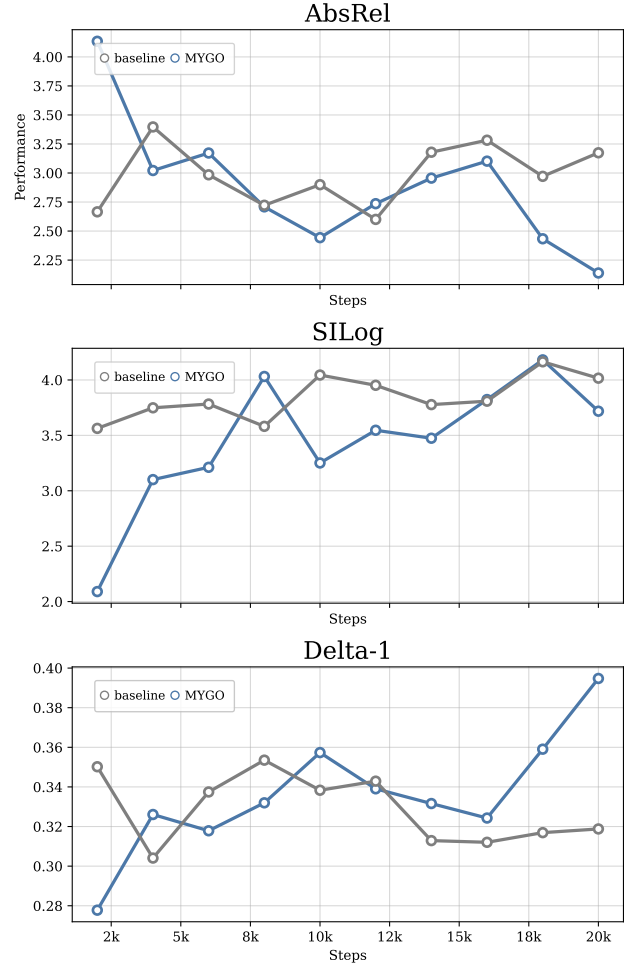


Figure 6. AbsRel, SILog, and δ_1 over training steps for the baseline and MYGO.