Improving Controllable Generation: Faster Training and Better Performance via x_0-Supervision

Supplementary Material

Sec. A provides additional implementation details regarding our experiments and the computation of the mAUCC metric. Sec. B describes how to analytically convert the various diffusion and flow-matching supervision signals into an x_0 estimate. Sec. C comments on the convergence curves for T2I-Adapter. Sec. D offers insights into the suboptimal performance of the original v-ControlNet and ϵ-OminiControl approaches. Finally, in Sec. E, we present further qualitative results illustrating the performance of our x_0-supervision for controllable generation.

A. Further experimental details

A.1. Controllable Generation Tasks

We evaluated our proposed x_0-supervision for controllable generation on several control modalities and previous frameworks so as to assess its effectiveness.

All training hyperparameters are set to the same values as in the original papers, with the exception of the ControlNet batch size, which is set to 8 in contrast to 32 for depth, 256 for semantic segmentation, 32 for Canny edge, and 18 for human pose in the original work. We generate images using the DDIM algorithm with 50 sampling steps and a guidance scale of 7.5. The same configuration is used for T2I-Adapter. For OminiControl, we use 28 sampling steps with a guidance scale of 3.5.

Depth. MultiGen-20M is used as the depth control dataset. We train the selected methods using the different supervision signals. Experiments are performed with images at a resolution of 512×512.

Semantic Segmentation. The ADE20K [20] dataset is used as the segmentation control dataset. We use Florence-2 [18] to generate image captions. Experiments are performed with images at a resolution of 512×512.

Canny Edge. We use the ADE20K dataset, annotated with Florence-2. In order to extract the control images, we apply the Canny edge detector to the semantic segmentation images to focus on object boundary edges. The aim is to ensure a robust evaluation protocol, as inner edges are highly sensitive to hysteresis thresholds.

Pose. We utilize the MS-COCO keypoints dataset [6]. We resize the images to a resolution of 512×512 and adjust the keypoint and bounding box coordinates accordingly. We retain only images where the bounding box covers at least 2% of the total image area and which contain between 1 and 6 visible persons. Additionally, for each image, we use the first of the 5 captions provided in MS-COCO. We follow the same training and sampling procedures as described above.

Box+Text grounding. We utilize the dataset provided by the authors of GLIGEN [5]. It comprises a combination of GoldG [3], SBU [10], CC3M [14], and Object365 [13]. Further details can be found in the original paper. For training, we perform 200k iterations with a batch size of 64. For evaluation, we use the MS-COCO subset from [17]. The template *"a <object>"* is used for object captions, where *<object>* is replaced by the actual class name of the object. These captions are further embedded using the CLIP text encoder, as in the original paper. We generate images using the PLMS sampler [8] with 50 sampling steps and a guidance scale of 7.5, following the procedure described in the original paper. YOLO detection scores are computed image-wise and then averaged.

Box+Text+Image grounding. We use the same dataset as described above, following the original paper. We train for 200k iterations with a batch size of 64. The same MS-COCO subset is used for evaluation. In addition to object caption embeddings, object image embeddings are computed following the same procedure as in the original work. Finally, the same hyperparameters are used for sampling.

A.2. mAUCC definition and implementation

In this section, we provide additional justifications to the definition of **mAUCC**. The area under the control fidelity curves is a reasonable way to assess the convergence speed since it faithfully represents the cumulative performance gains for a given training horizon. One natural choice is then to report **AUCC**@T_{max}, where T_{max} is the number of steps used for training. However, this is very sensitive to the choice of a T_{max}. In particular, for large values of T_{max}, the contribution of the initial steps can be asymptotically washed away, resulting in a artificially low variance between different supervision. This may happen if a performance plateau is reached early. To reduce this sensitivity, we advocate for a metric averaging over multiple horizons. However, this implicitly up-weights initial steps. Formally, defining $b_i = \int_{t_i}^{t_{i+1}} m_s \, d_s$ with $t_0 = 0 <$

Control	Training set	Validation set	Metric
Depth	MultiGen-20M [4] train	MultiGen-20M [4] val	RMSE
Semantic Segmentation	ADE20K [20] train	ADE20K [20] val	mIoU
Canny Edge	ADE20K [20] train	ADE20K [20] val	F1
Pose	MS-COCO [6] train filtered	MS-COCO [6] val filtered	Average Precision (AP)
Box+Text grounding	GoldG [3]+SBU [10]+CC3M [14]+Object365 [13]	MS-COCO [6] subset [17]	Average Precision (AP)
Box+Text+Image grounding	GoldG [3]+SBU [10]+CC3M [14]+Object365 [13]	MS-COCO [6] subset [17]	Average Precision (AP)

Table 1. **Datasets used for experiments**

$t_1 < \cdots < t_n$ different training horizons, we can also write **mAUCC** $= \sum_{i=0}^{n-1} w_i b_i$, with $w_i = \frac{1}{n}\sum_{j=i+1}^{n} \frac{1}{t_j}$, and **AUCC**@$t_n = \sum_{i=0}^{n-1} \frac{1}{t_n} b_i$. Thus, while **AUCC**@$T_{max}$ gives a uniform weight to all b_i, **mAUCC** emphasizes initial steps since w_i is decreasing. This is in fact desired in our context for assessing convergence speed. For completeness we additionally report the values of **AUCC**@T_{max} on Tabs. 3 and 4 for completeness. We can see that the two metrics have similar trends.

We also provide the implementation of the mean Area Under the Convergence Curve (mAUCC) in Algorithm 1. As explained in the main paper, to obtain normalized values, we integrate the normalized convergence curves over $[0, 1]$. To do so the metric are divided by their maximum achievable value, which 255 for the RMSE and 100 for the others. We reporte the mAUCC on a $[0, 100]$ scale on the result tables.

Method	Base model	Batch size	Training steps
ControlNet [19]	Stable Diffusion 1.5 [11]	8	200k
T2I-Adapter [9]	Stable Diffusion 1.5 [11]	8	200k
OminiControl [16]	FLUX.1 [2]	8	40k
GLIGEN [5]	Stable Diffusion 1.4 [11]	64	200k

Table 2. **Methods used for experiments**

Algorithm 1 Python code for mAUCC

```
import numpy as np
import scipy

def calculate_aucc(convergence_curve):
    # Computes the area using the trapezoidal rule
    N = len(convergence_curve)
    steps = np.arange(N) / N
    return scipy.integrate.trapezoid(
        performance_curve,
        steps
    )

def calculate_mean_aucc(
    convergence_curve,
    thresholds=None
):
    # Computes the average AUCC over multiple horizons
    if thresholds is None:
        thresholds = np.linspace(
            0.25, 1, int(np.round((1 - 0.25)/.05)) + 1,
            endpoint=True
        )
    AUCCs = []
    for ti in thresholds:
        max_steps = int(len(performance_curve) * ti)
        curve_trunc = convergence_curve[:max_steps]
        AUCC_at_ti = calculate_aucc(curve_trunc)
        AUCCs.append(AUCC_at_ti)
    mAUCC = np.mean(AUCCs)
    return mAUCC
```

B. Conversion formulas

We describe the way to convert the different objectives of diffusion and flow matching to an x_0 estimate. The derivations are mostly reported from existing works [7, 12]. Our main contribution is to analyse their impact on controllable generation and propose a method for improving both the training speed and the final performance.

Although flow matching relies on Ordinary Differential Equations (ODEs) and diffusion modeling relies on Stochastic Differential Equations (SDEs), during training they are equivalent and the main difference is the learning objective. In particular, the two paradigms are unified under the framework of stochastic interpolants [1], they both use the one-sided interpolant:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon \tag{1}$$

where α_t and σ_t are respectively decreasing and increasing time-dependent functions on $[0, T]$, such that $\alpha_0 = \sigma_T = 1$ and $\alpha_T = \sigma_0 = 0$. Note that in the flow matching literature, this time convention is reversed. That is, α_t and σ_t are increasing and decreasing such that $\alpha_0 = \sigma_T = 0$ and $\alpha_T = \sigma_0 = 1$. But, for notation consistency, we will use the first convention so that x_0 is the clean image and x_T the fully noised image. The main distinction between the two paradigms comes from how α_t and σ_t are instantiated and which objective is learned, each objective giving a different parameterization. Despite this variety of objectives, they can be converted into one another. As a matter of fact, the different learning losses can be seen as a particular weighting of the x_0 supervision loss:

$$\mathcal{L}_\theta^p = \mathbb{E}_{t,\epsilon,x_0}\left[w_t^p \|x_0 - x_\theta(x_t, t)\|_2^2\right] \tag{2}$$

where w_t^p is the weighting for the parameterization p. A summary of the conversion formulas and their equivalent loss re-weightings are provided in Tab. 5. We now describe the derivations for the different parameterizations in details.

Supervision	Depth		Semantic Seg		Canny Edge		Pose	
	AUCC@T_{max} ↓	mAUCC↓	**AUCC**@T_{max} ↑	mAUCC↑	**AUCC**@T_{max} ↑	mAUCC↑	**AUCC**@T_{max} ↑	mAUCC↑
ControlNet (Diffusion)								
ϵ-ControlNet	15.97	17.70	30.61	25.19	9.24	8.09	44.85	35.86
v-ControlNet	18.56	21.89	20.39	13.48	8.41	6.93	36.20	22.72
x_0-ControlNet	**15.01**	**15.98**	**35.60**	**31.52**	**9.71**	**8.76**	**48.63**	**42.19**
T2I-Adapter (Diffusion)								
ϵ-T2I-Adpater	21.77	22.44	19.00	17.39	3.73	3.61	15.66	12.98
v-T2I-Adapter	25.94	26.52	12.51	11.54	2.66	2.62	6.02	4.44
x_0-T2I-Adapter	**19.53**	**19.67**	**22.01**	**21.07**	**4.51**	**4.47**	**23.00**	**21.45**
OminiControl (Flow Matching)								
u-OminiControl	**13.38**	**13.47**	41.37	**39.13**	**15.05**	14.40	43.87	34.54
ϵ-OminiControl	15.45	16.79	29.28	23.76	11.15	9.71	26.78	17.57
x_0-OminiControl	13.58	13.74	**41.39**	38.69	14.99	**14.41**	**49.28**	**42.07**

Table 3. **Comparison between the different supervision signals on different spatially-aligned control modalities and methods.** The curves are smoothed with EMA weight 0.9 before computing **AUCC**@T_{max} and **mAUCC**.

Supervision	Box+Text		Box+Text+Image	
	AUCC@T_{max} ↑	mAUCC↑	**AUCC**@T_{max} ↑	mAUCC↑
GLIGEN (Diffusion)				
ϵ-GLIGEN	14.61	8.28	9.92	6.15
v-GLIGEN	2.07	1.54	3.34	9.56
x_0-GLIGEN	**18.38**	**18.38**	**11.49**	**8.07**

Table 4. **Comparison between the different supervision signals on different non-spatially-aligned control modalities.** The curves are smoothed with EMA weight 0.9 before computing **AUCC**@T_{max} and **mAUCC**.

Parameterization	Conversion	Loss re-weighting
ϵ	$\frac{1}{\alpha_t}$, $-\frac{\sigma_t}{\alpha_t}$	$\frac{\sigma_t^2}{\alpha_t^2}$
v	α_t, $-\sigma_t$	$\left(\frac{\alpha_t^2}{\sigma_t^2}+1\right)^{-1}$
u	$-\frac{\dot{\sigma}_t}{\dot{\alpha}_t\sigma_t-\alpha_t\dot{\sigma}_t}$, $\frac{\sigma_t}{\dot{\alpha}_t\sigma_t-\alpha_t\dot{\sigma}_t}$	$\left(\frac{\sigma_t}{\dot{\alpha}_t\sigma_t-\alpha_t\dot{\sigma}_t}\right)^2$

Table 5. **Conversion formulas from different predictors to an x_0-predictor**. In the second column, (a_t, b_t) corresponds to $x_\theta(x_t, t) = a_t x_t + b_t f(x_t, t)$ where f is the predictor. In the third column are the equivalent re-weighting factors of the initial losses.

B.1. Diffusion

In diffusion, α_t and σ_t are indirectly defined by reasoning on the forward diffusion process.

ϵ-parameterization. Using the relation Eq. (1) one can write:

$$x_0 = \frac{x_t - \sigma_t \epsilon}{\alpha_t} \text{ and } \epsilon = \frac{x_t - \alpha_t x_0}{\sigma_t} \tag{3}$$

hence the loss of an ϵ-predictor can be written as:

$$\mathcal{L}_\theta^\epsilon = \mathbb{E}_{t,\epsilon,x_0}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right], \tag{4}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\|\frac{1}{\sigma_t}(x_t - \alpha_t x_0) - \frac{1}{\sigma_t}(x_t - \alpha_t x_\theta(x_t, t))\|_2^2\right] \tag{5}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\frac{\alpha_t^2}{\sigma_t^2}\|x_0 - x_\theta(x_t, t)\|_2^2\right] \tag{6}$$

hence $w_t^\epsilon = \frac{\alpha_t^2}{\sigma_t^2}$, which is the signal-to-noise ratio (SNR).

v-parameterization. [12] defines the v-parameterization, also known as the Variance Preserving (VP) schedule, in the case where $\alpha_t^2 + \sigma_t^2 = 1$. Under this assumption, the interpolant is expressed in terms of $\phi_t = \arctan(\sigma_t/\alpha_t)$:

$$x_\phi = \alpha_\phi x_0 + \sigma_\phi \epsilon = \cos(\phi)x_0 + \sin(\phi)\epsilon \tag{7}$$

hence v is defined as:

$$v_\phi = \frac{dx_\phi}{d\phi} = \cos(\phi)\epsilon - \sin(\phi)x_0 \tag{8}$$

Therefore, v is an angular velocity. We can then derive x_0 from v as follows:

$$x_0 = \frac{\cos(\phi)\epsilon - v_\phi}{\sin(\phi)} \tag{9}$$

$$= \frac{\cos(\phi)\frac{x_\phi - \cos(\phi)x_0}{\sin(\phi)} - v_\phi}{\sin(\phi)} \tag{10}$$

$$= \frac{\cos(\phi)x_\phi - \cos^2(\phi)x_0 - \sin(\phi)v_\phi}{\sin^2(\phi)} \tag{11}$$

$$\sin^2(\phi)x_0 = \cos(\phi)x_\phi - \cos^2(\phi)x_0 - \sin(\phi)v_\phi \tag{12}$$

$$\left[\cos^2(\phi) + \sin^2(\phi)\right]x_0 = \cos(\phi)x_\phi - \sin(\phi)v_\phi \tag{13}$$

$$x_0 = \cos(\phi)x_\phi - \sin(\phi)v_\phi \tag{14}$$

hence, coming back to α_t and σ_t, we can summarize by writing:

$$x_0 = \alpha_t x_t - \sigma_t v_t \text{ and } v_t = \alpha_t \epsilon - \sigma_t x_0 \tag{15}$$

Using these expressions, the loss of a v-predictor can be written as:

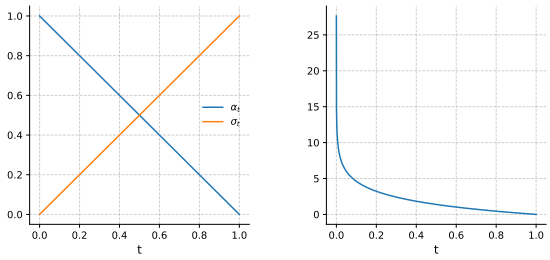
$$\mathcal{L}_\theta^v = \mathbb{E}_{t,\epsilon,x_0}\left[\|v_t - v_\theta(x_t,t)\|_2^2\right], \tag{16}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\|\frac{1}{\sigma_t}(\alpha_t x_t - x_0) - \frac{1}{\sigma_t}(\alpha_t x_t - x_\theta(x_t,t))\|_2^2\right] \tag{17}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\frac{1}{\sigma_t^2}\|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{18}$$

hence, $w_t^v = \frac{1}{\sigma_t^2} = \frac{\alpha_t^2+\sigma_t^2}{\sigma_t^2} = \frac{\alpha_t^2}{\sigma_t^2} + 1$, which is referred to as SNR+1 weighting in [12], where the authors used it in the context of diffusion model distillation.

B.2. Flow matching



(a) OT noise schedule. (b) $\log w_t^u$.

Figure 1. **OT noise schedule used in flow matching and the corresponding weighting incurred by x_0-supervision in log scale.**

In flow matching, the goal is to predict the velocity field u governing the probability flow ODE [15]:

$$\frac{dX_t}{dt} = u_t(X_t) \tag{19}$$

Equation (1) describes the conditional flow $\psi_t(x_0,\epsilon) : t \mapsto \alpha_t x_0 + \sigma_t \epsilon$, and one has:

$$u_t(x) = \mathbb{E}_{x_0,\epsilon}\left[u_t(X_t|x_0,\epsilon)|X_t = x\right] \tag{20}$$

$$= \mathbb{E}_{x_0,\epsilon}\left[\dot{\psi}_t(x_0,\epsilon)\middle|X_t = x\right] \tag{21}$$

$$= \mathbb{E}_{x_0,\epsilon}\left[\dot{\alpha}_t x_0 + \dot{\sigma}_t \epsilon | X_t = x\right] \tag{22}$$

$$= \mathbb{E}_{x_0,\epsilon}\left[\dot{\alpha}_t x_0 + \dot{\sigma}_t \frac{X_t - \alpha_t x_0}{\sigma_t}\middle| X_t = x\right] \tag{23}$$

$$= \mathbb{E}_{x_0,\epsilon}\left[\frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\sigma_t} x_0 + \frac{\dot{\sigma}_t}{\sigma_t} x \middle| X_t = x\right] \tag{24}$$

$$= \frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\sigma_t} \underbrace{\mathbb{E}_{x_0,\epsilon}\left[x_0|X_t = x\right]}_{x_0\text{-predictor}} + \frac{\dot{\sigma}_t}{\sigma_t} x \tag{25}$$

where the $\cdot$ denotes the time derivative and Eq. (23) uses the fact that $\epsilon = \frac{X_t - \alpha_t x_0}{\sigma_t}$. In practice, one trains a network to predict the conditional velocity field $u_t(\cdot|x_0,\epsilon)$ by optimizing the conditional flow matching loss:

$$\mathcal{L}_\theta^u = \mathbb{E}_{t,x_0,\epsilon}\left[\|u_t(x_t|x_0,\epsilon) - u_\theta(x_t,t)\|_2^2\right] \tag{26}$$

$$= \mathbb{E}_{t,x_0,\epsilon}\left[\left(\frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\sigma_t}\right)^2 \|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{27}$$

where Eq. (27) uses the relation between the velocity and x_0 in Eqs. (24) and (25). Hence, we get $w_t^u = \left(\frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\sigma_t}\right)^2$. This can be re-written as follows:

$$w_t^u = \left(\alpha_t\left(\frac{\dot{\alpha}_t}{\alpha_t} - \frac{\dot{\sigma}_t}{\sigma_t}\right)\right)^2 \tag{28}$$

$$= \alpha_t^2\left(\frac{d\log\alpha}{dt}(t) - \frac{d\log\sigma}{dt}(t)\right)^2 \tag{29}$$

$$= \alpha_t^2\left(\frac{d}{dt}\left(\log\frac{\alpha}{\sigma}\right)(t)\right)^2 \tag{30}$$

$$= \frac{\alpha_t^2}{4}\left(\frac{d\log\text{SNR}}{dt}(t)\right)^2 \tag{31}$$

Although theoretically $\left(\frac{d\log\text{SNR}(t)}{dt}\right)^2$ diverges as $t \to T$, this issue can be mitigated in practice. Overall, w_t^u is very small near T compared to its values near 0. Figure 1 illustrates this for the Optimal Transport (OT) path, *i.e.*, where $\alpha_t = 1 - t$ and $\sigma_t = t$. The over-weighting of the high-SNR region relative to the low-SNR region causes the network to focus less on the early denoising steps during training.

C. Convergence curves for T2I-Adapter

The convergence curves for T2I-Adapter are provided on Fig. 2. We can see that x_0-supervision consistently improves convergence on all selected tasks.

D. Analysis of the results for converting to v/ϵ

In our experiments, we observed that v-ControlNet and ϵ-OminiControl do not achieve optimal performance. We show in this section that this can be explained by the implied weightings. Formally, in the case of ϵ-to-v, we obtain:

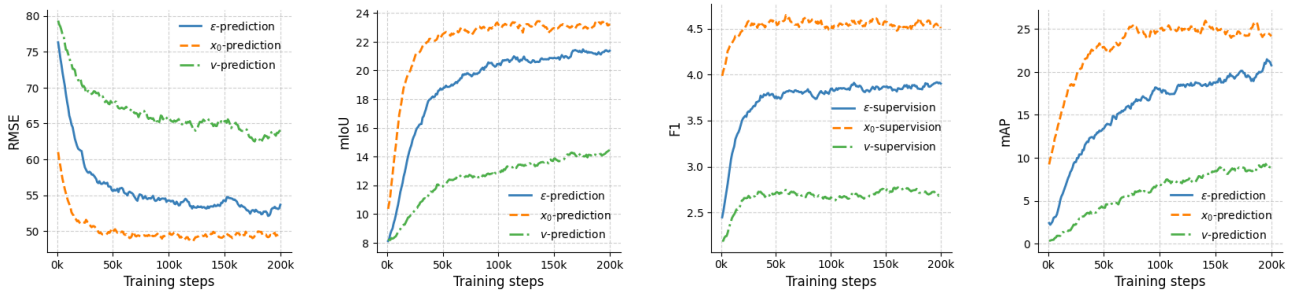
$$\mathcal{L}_\theta^{\epsilon\to v} = \mathbb{E}_{t,\epsilon,x_0}\left[\|v_t - (\alpha_t\epsilon_\theta(x_t,t) - \sigma_t x_0)\|_2^2\right] \tag{32}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\|(\alpha_t\epsilon - \sigma_t x_0) - (\alpha_t\epsilon_\theta(x_t,t) - \sigma_t x_0)\|_2^2\right] \tag{33}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\alpha_t^2\|\epsilon - \epsilon_\theta(x_t,t)\|_2^2\right] \tag{34}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\alpha_t^2 w_t^\epsilon\|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{35}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\frac{\alpha_t^4}{\sigma_t^2}\|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{36}$$



(a) T2I-Adapter Depth. (b) T2I-Adapter Segmentation. (c) T2I-Adapter Canny Edge. (d) T2I-Adapter Pose.

Figure 2. **Convergence curves for T2I-Adapter on different tasks.** We use an EMA weight of 0.9 to smooth the curves. We can notice that the convergence is faster with x_0-supervision.

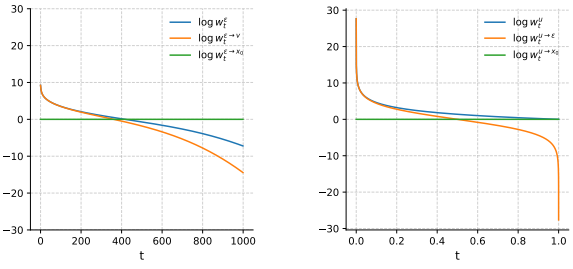
and for the case of u-to-ϵ we get:

$$\mathcal{L}_\theta^{u\to\epsilon} = \mathbb{E}_{t,\epsilon,x_0}\left[\|\epsilon - \frac{\alpha_t}{\alpha_t\dot{\sigma}_t - \dot{\alpha}_t\sigma_t}(u_\theta(x_t,t) - \frac{\dot{\alpha}_t}{\alpha_t}x_t)\|_2^2\right] \tag{37}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\left(\frac{\alpha_t}{\alpha_t\dot{\sigma}_t - \dot{\alpha}_t\sigma_t}\right)^2 \|u_t(x_t|x_0,\epsilon) - u_\theta(x_t,t)\|_2^2\right] \tag{38}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\left(\frac{\alpha_t}{\alpha_t\dot{\sigma}_t - \dot{\alpha}_t\sigma_t}\right)^2 w_t^u\|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{39}$$

$$= \mathbb{E}_{t,\epsilon,x_0}\left[\frac{\alpha_t^2}{\sigma_t^2}\|x_0 - x_\theta(x_t,t)\|_2^2\right] \tag{40}$$



(a) Diffusion log-weightings. (b) Flow matching log-weightings.

Figure 3. **Log-weighting functions.** In Fig. 3a, we plot the log-weightings incurred by the x_0-supervision loss when using ϵ, v, and x_0 as supervision signals for SD-based control methods. In Fig. 3b, we plot the corresponding weightings when using u, ϵ, and x_0 as supervision signals for OminiControl. In both cases, we observe that lower convergence speed and performance are related to suboptimal weighting of the initial denoising steps (near T). In the case of u-supervision in particular, the weighting converges to 1 near T, making it less detrimental than ϵ-supervision. However, since its weighting near 0 is significantly higher, the training gradients focus more on these steps, which are less critical for the final image's control fidelity than the initial denoising steps.

hence both result in worse weightings that penalizes low SNRs. Therefore resulting in a slower convergence and worse performances in chosen training horizons. This is illustrated on Fig. 3.

E. Qualitative results

On Figs. 4 to 17, we provide some visual results on the control fidelity. Overall, with x_0-supervision, we either maintain or significantly improve the control fidelity.

References

[1] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *CoRR*, abs/2303.08797, 2023. 2

[2] Black Forest Labs. Flux: Official inference repository for flux.1 models, 2024. Accessed: 2024-11-12. 2

[3] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10955–10965. IEEE, 2022. 1, 2

[4] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VII*, pages 129–147. Springer, 2024. 2

[5] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: open-set grounded text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22511–22521. IEEE, 2023. 1, 2

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects

in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755. Springer, 2014. 1, 2

[7] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. 2

[8] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 1

[9] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 4296–4304. AAAI Press, 2024. 2

[10] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1143–1151, 2011. 1, 2

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022. 2

[12] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 2, 3, 4

[13] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8429–8438. IEEE, 2019. 1, 2

[14] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics, 2018. 1, 2

[15] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 4

[16] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14940–14950, 2025. 2

[17] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 6232–6242. IEEE, 2024. 1, 2

[18] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 4818–4829. IEEE, 2024. 1

[19] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3813–3824. IEEE, 2023. 2

[20] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

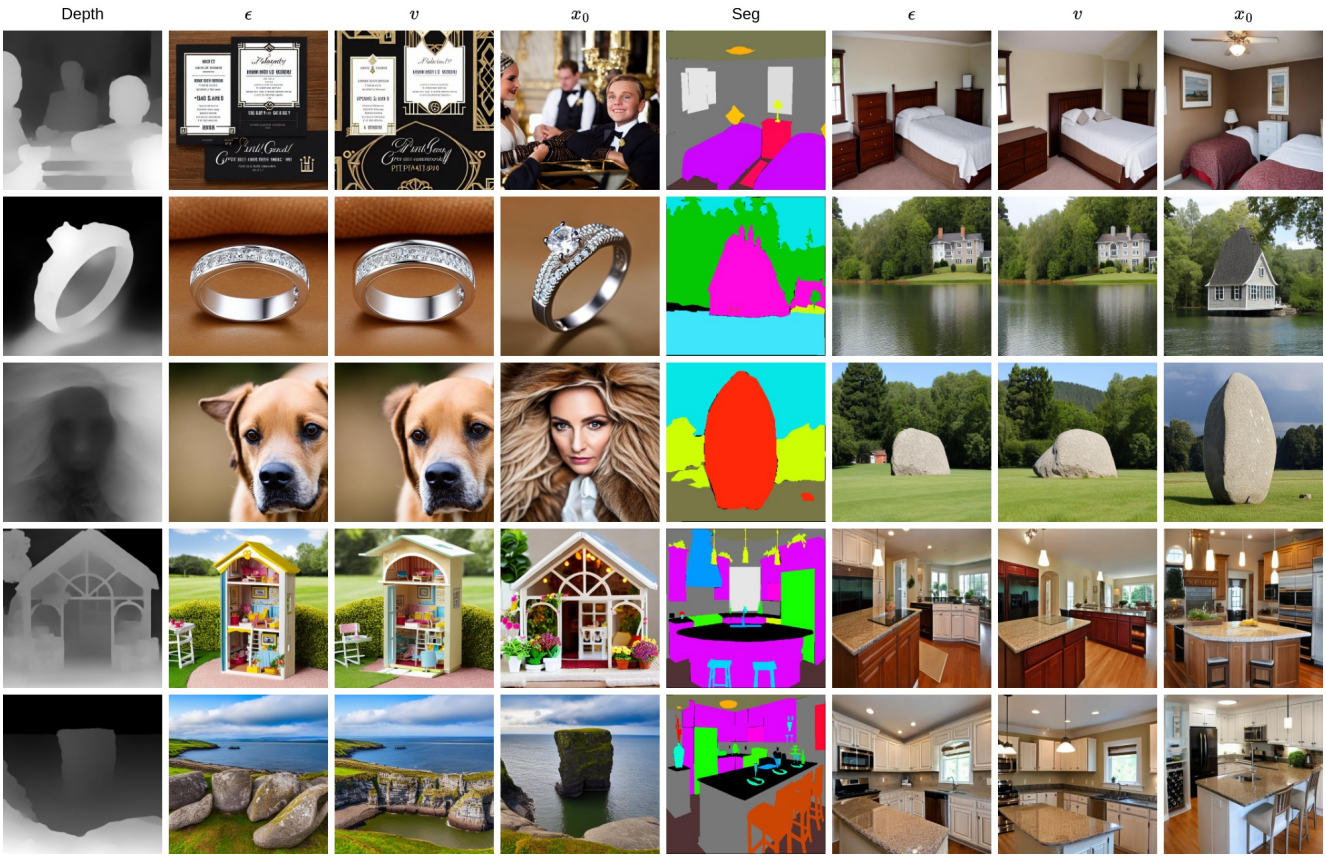


Figure 4. Qualitative results on depth and segmentation ControlNet with the three supervision signals after 10k training steps.

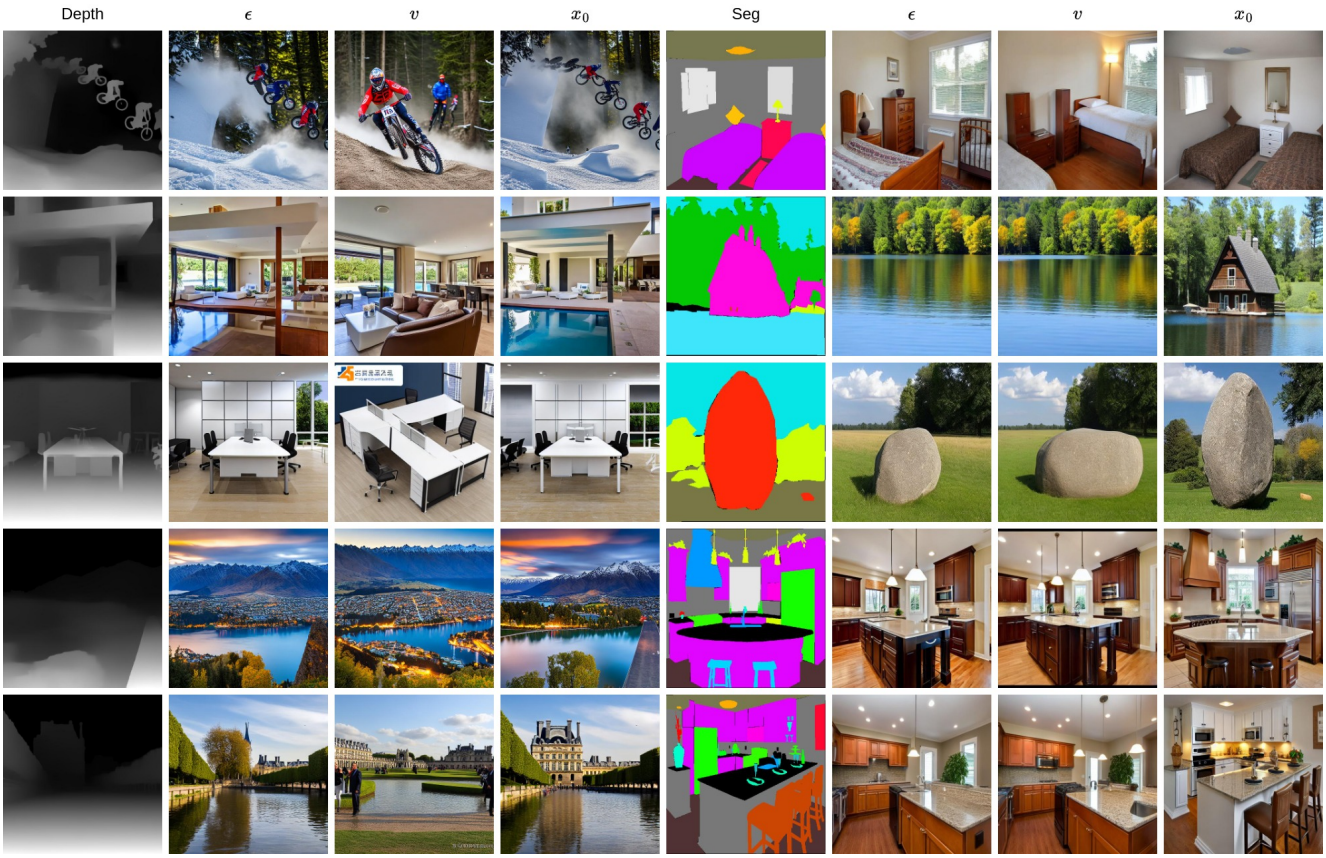


Figure 5. Qualitative results on depth and segmentation ControlNet with the three supervision signals after 25k training steps.

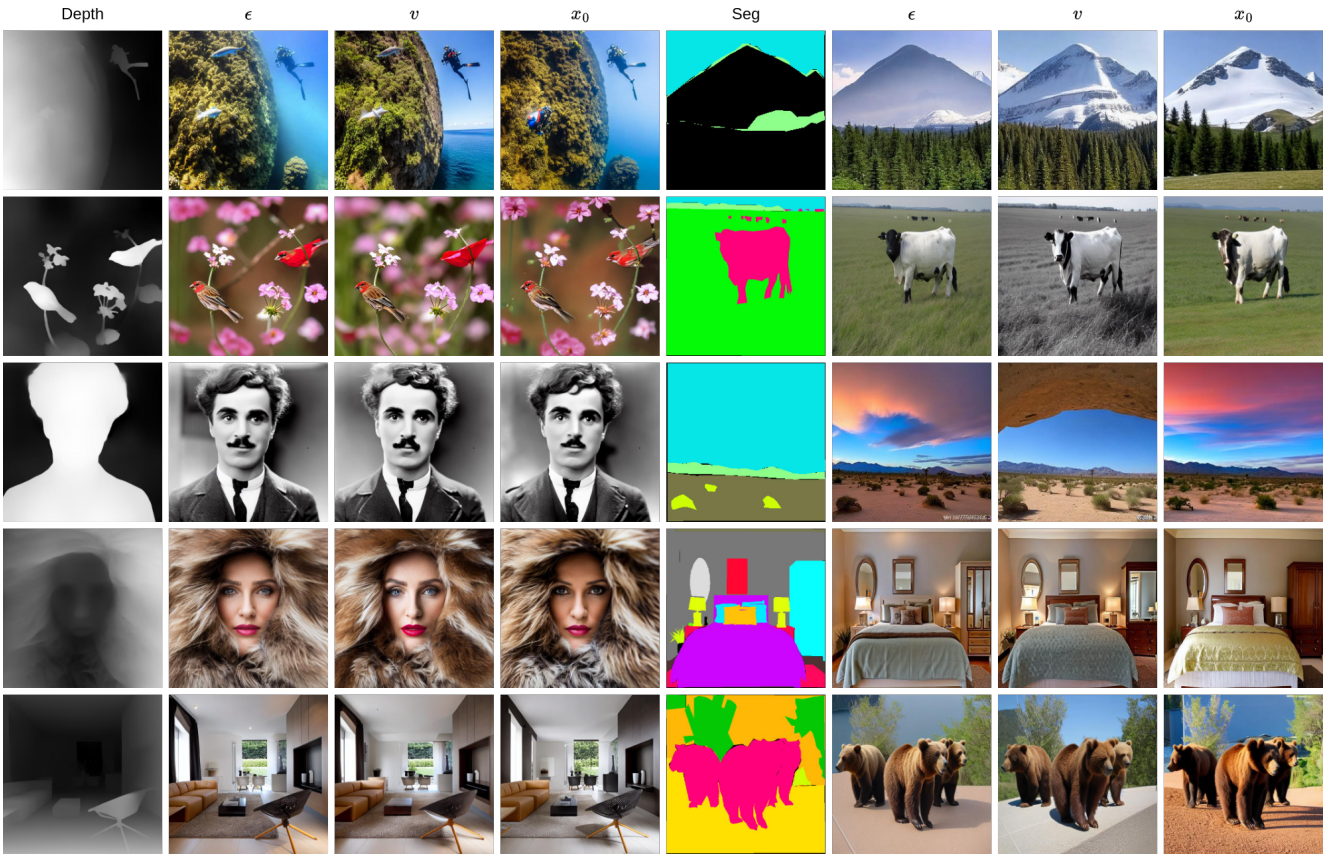


Figure 6. Qualitative results on depth and segmentation ControlNet with the three supervision signals after 200k training steps.

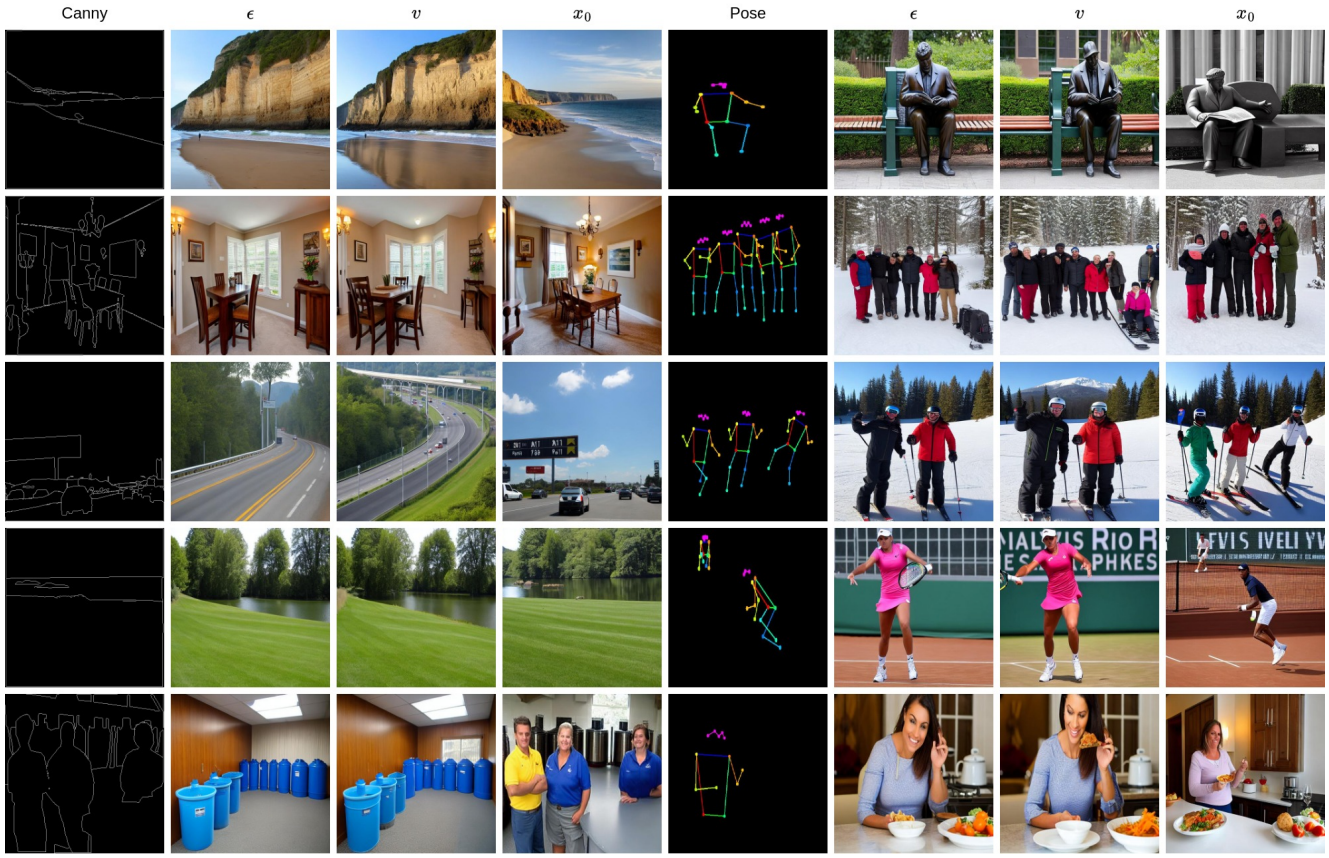


Figure 7. Qualitative results on Canny and pose ControlNet with the three supervision signals after 10k training steps.

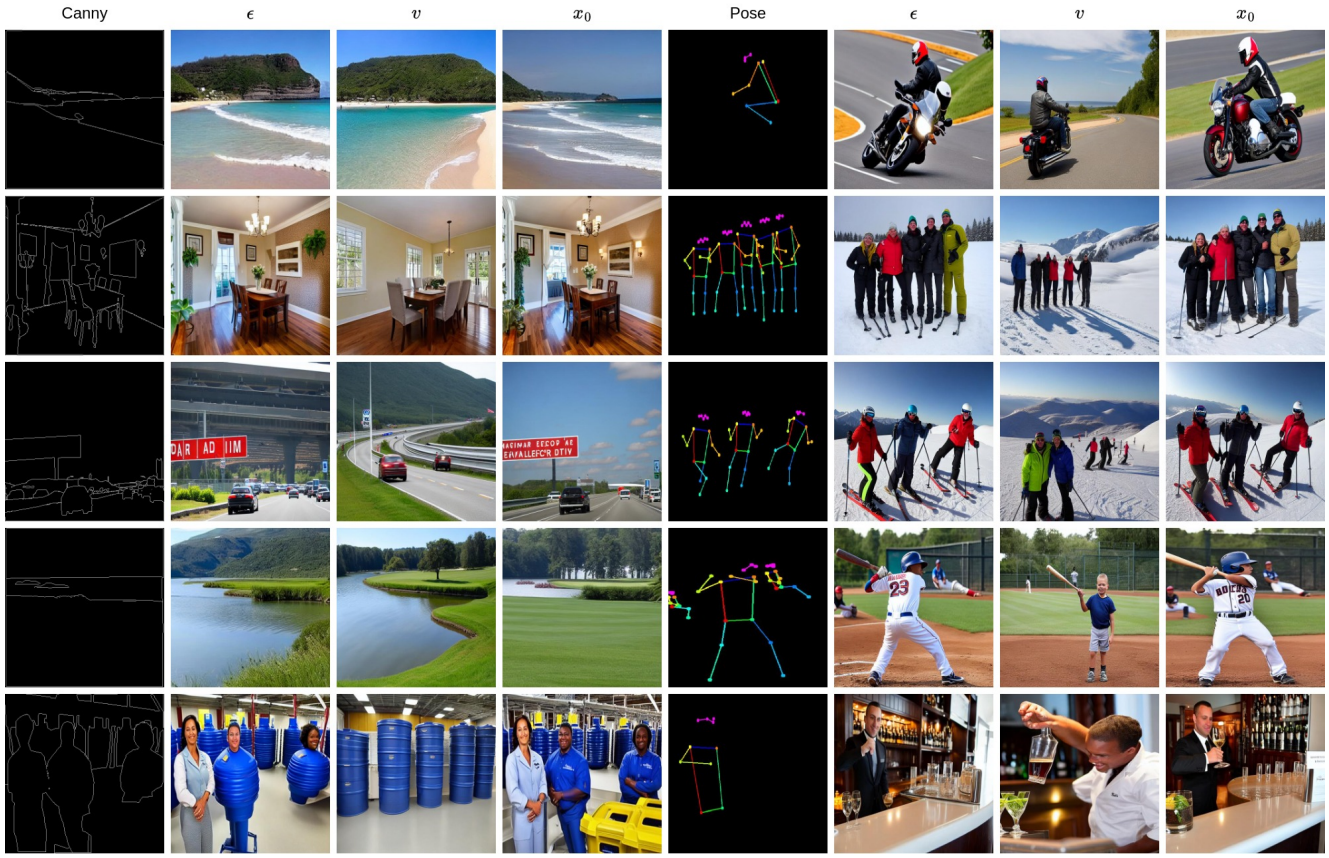


Figure 8. Qualitative results on Canny and pose ControlNet with the three supervision signals after 25k training steps.

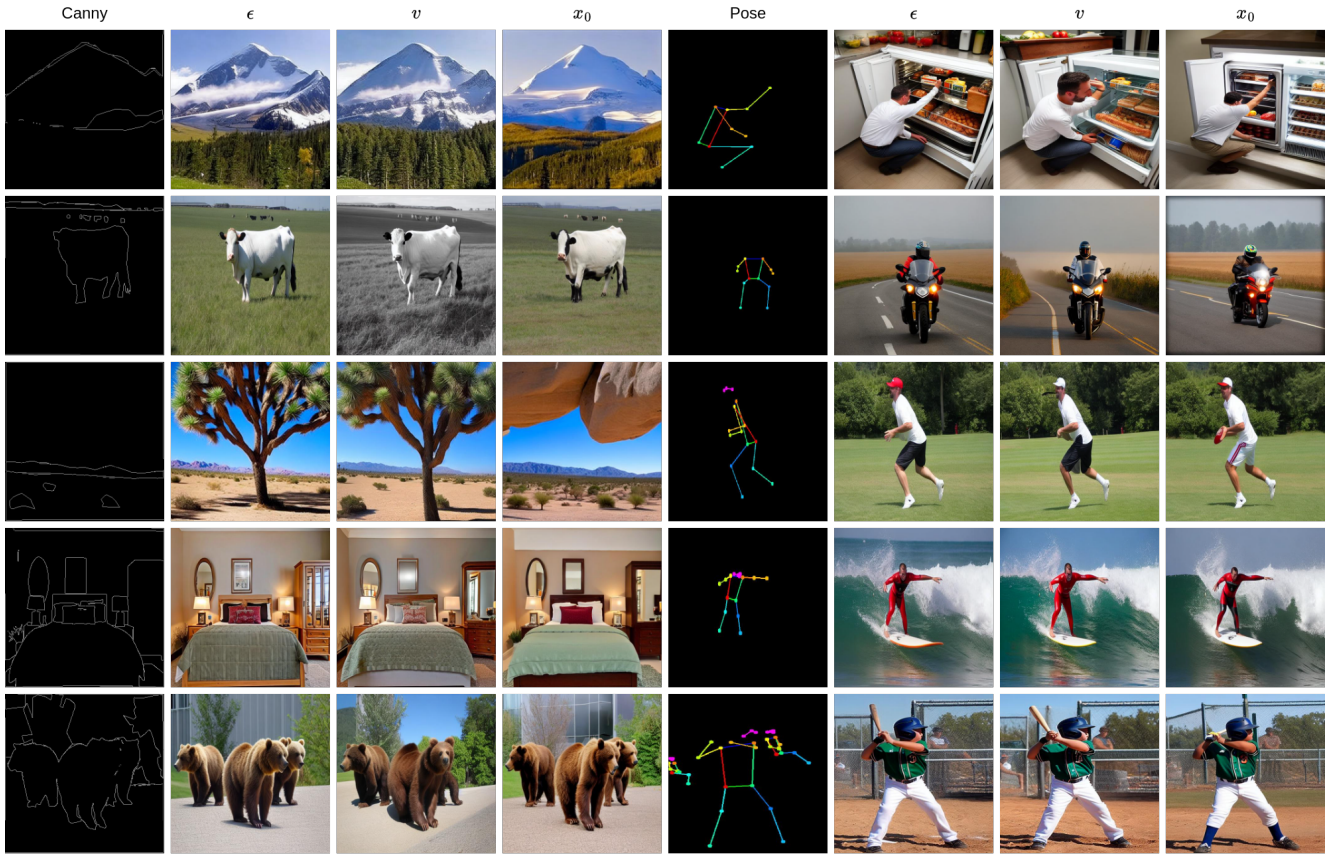


Figure 9. Qualitative results on Canny edge and pose ControlNet with the three supervision signals after 200k training steps.

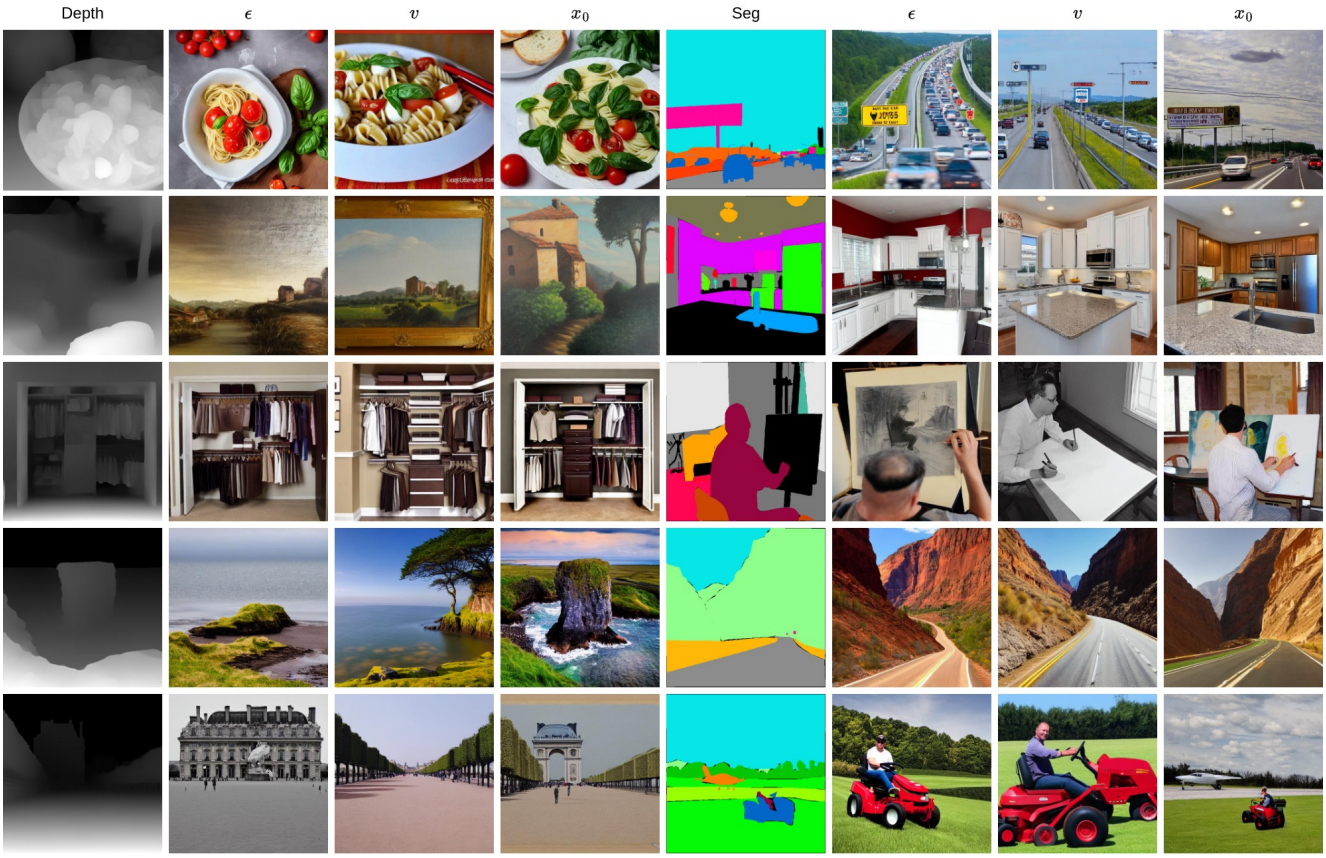


Figure 10. Qualitative results on depth and segmentation T2I-Adapter with the three supervision signals after 10k training steps.

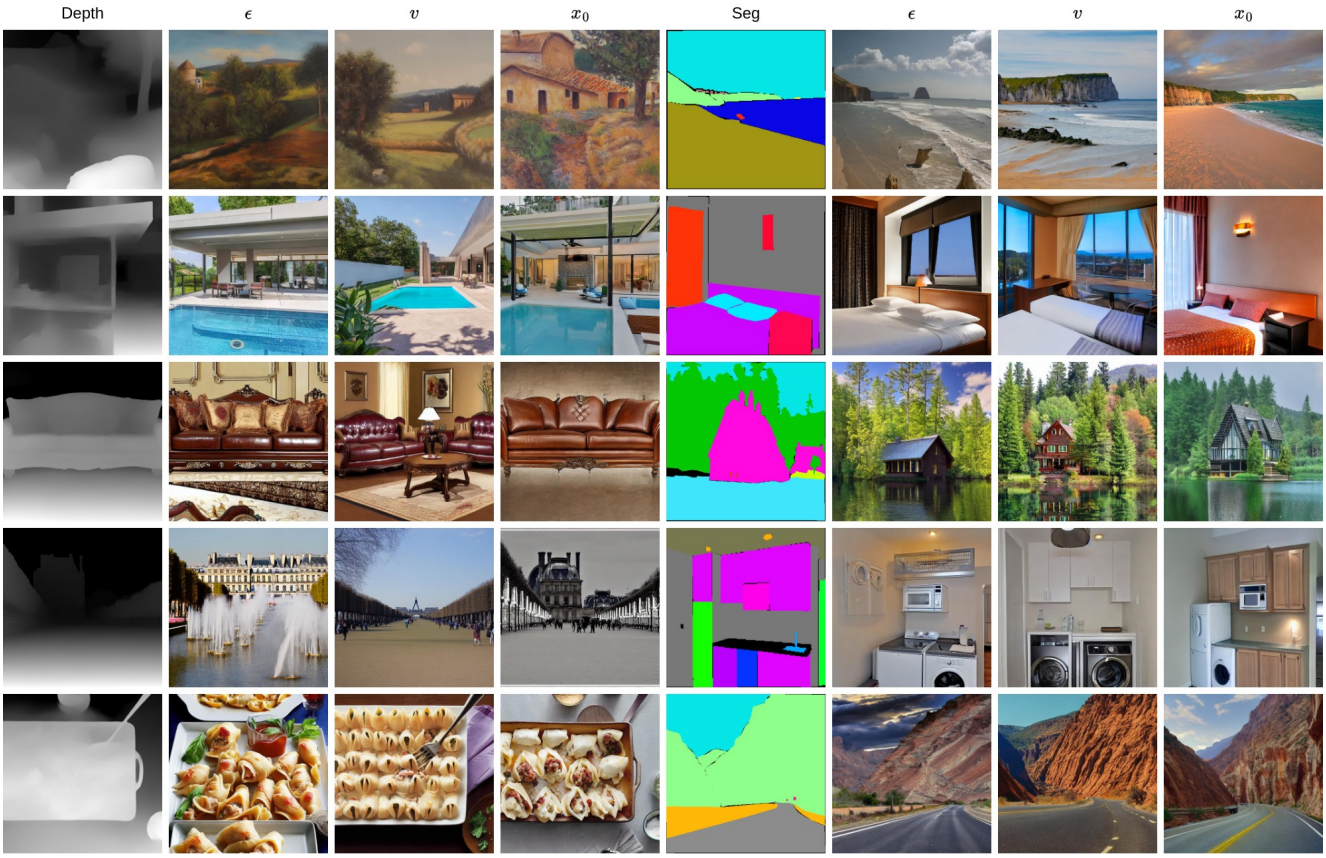


Figure 11. Qualitative results on depth and segmentation T2I-Adapter with the three supervision signals after 200k training steps.

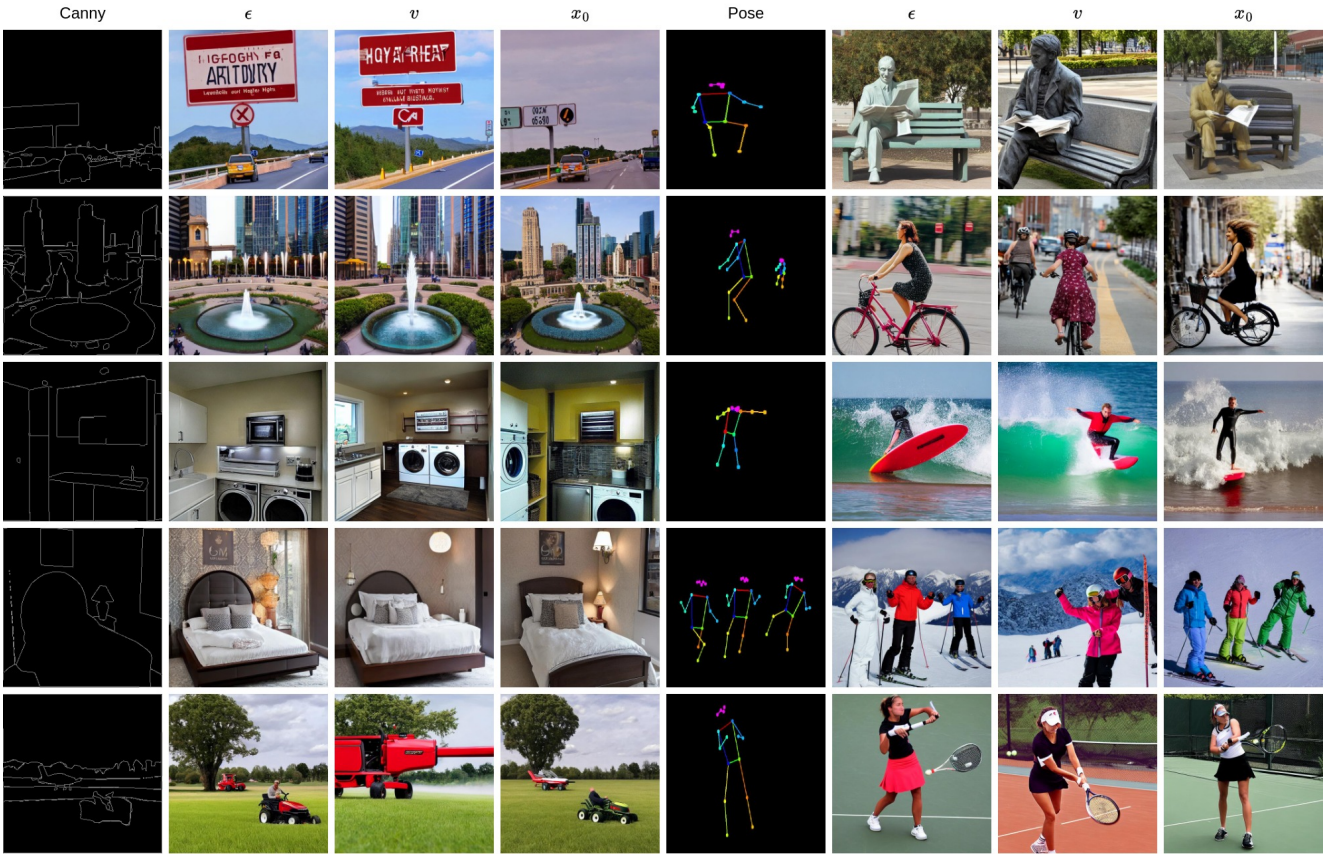


Figure 12. Qualitative results on Canny edge and pose T2I-Adapter with the three supervision signals after 10k training steps.

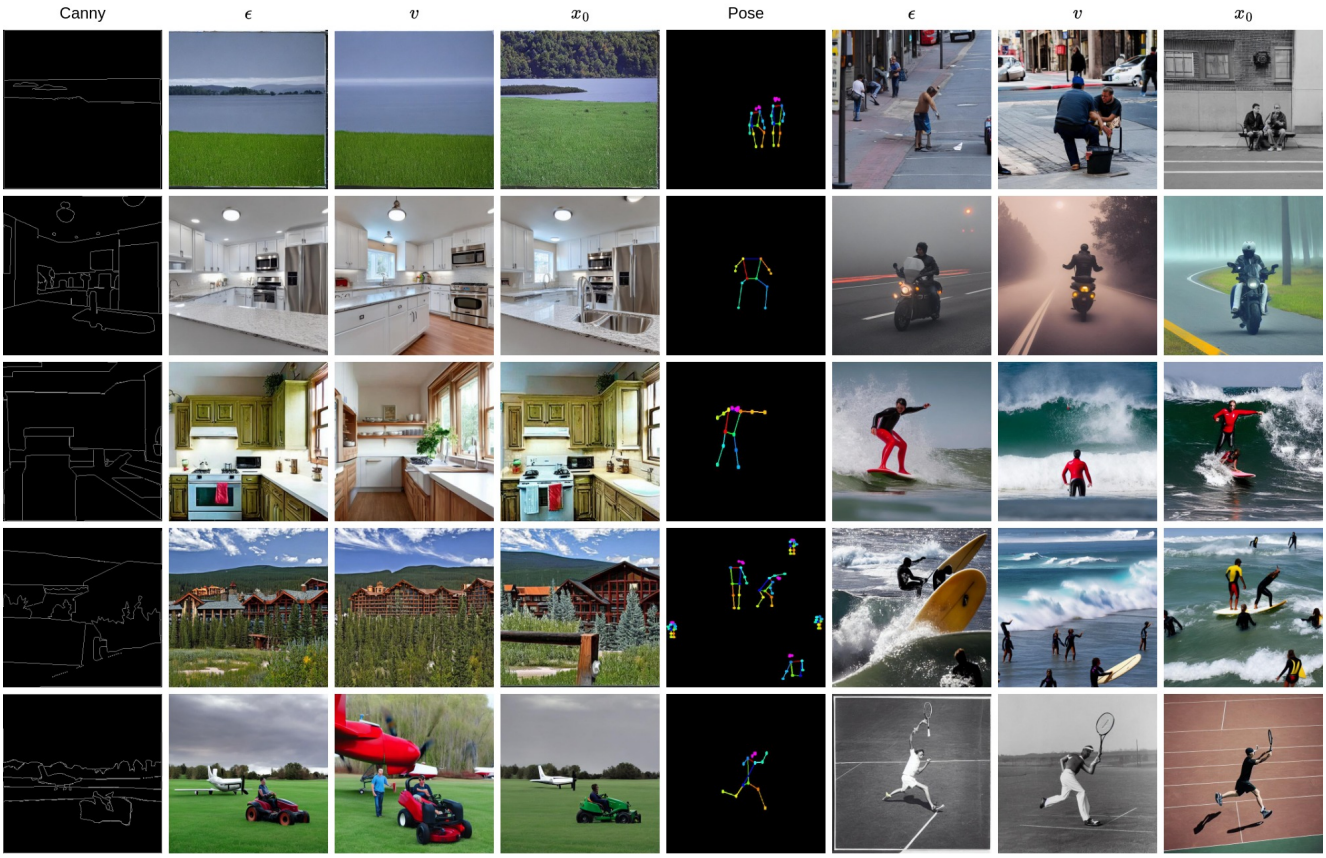


Figure 13. Qualitative results on Canny edge and pose T2I-Adapter with the three supervision signals after 200k training steps.

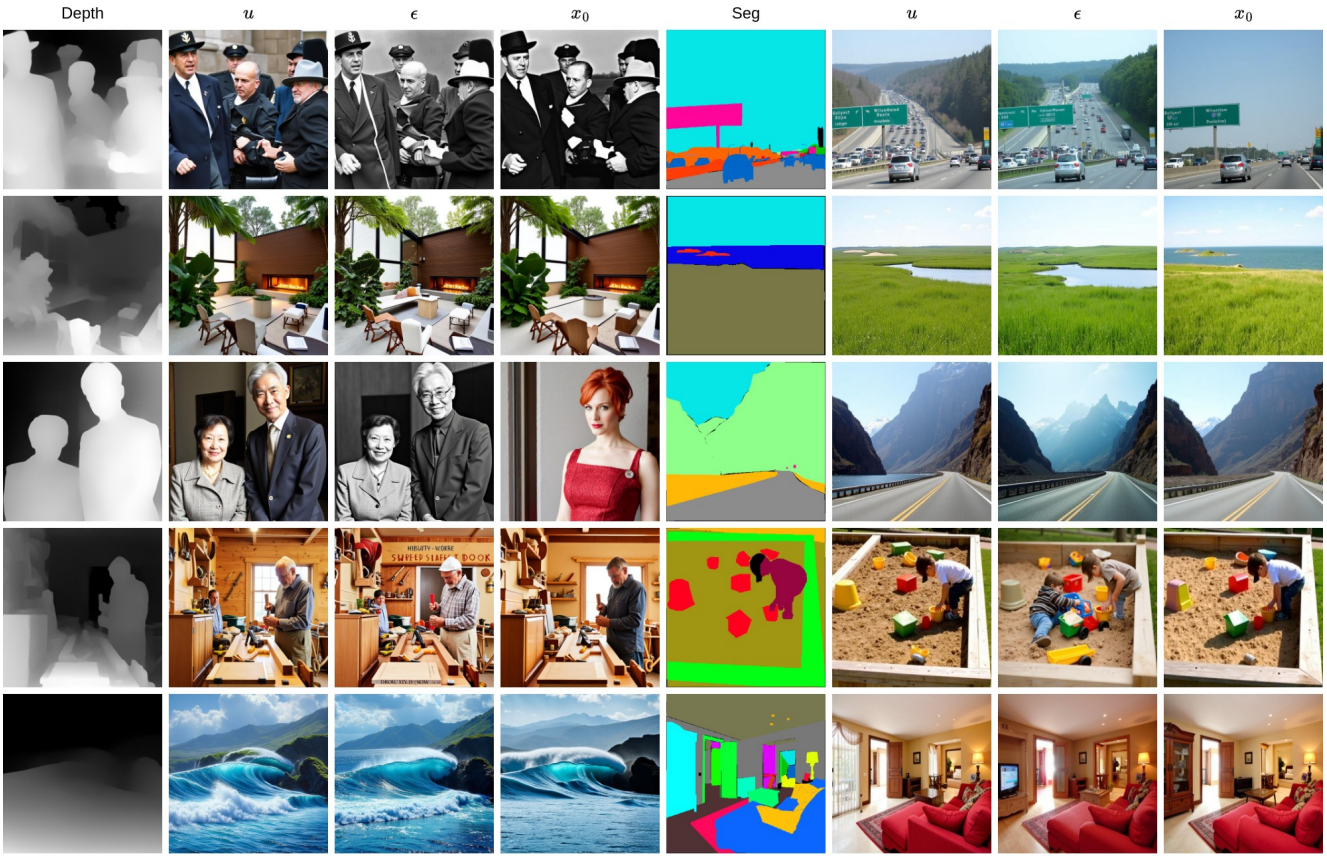


Figure 14. Qualitative results on depth and segmentation OminiControl with the three supervision signals after 5k training steps.

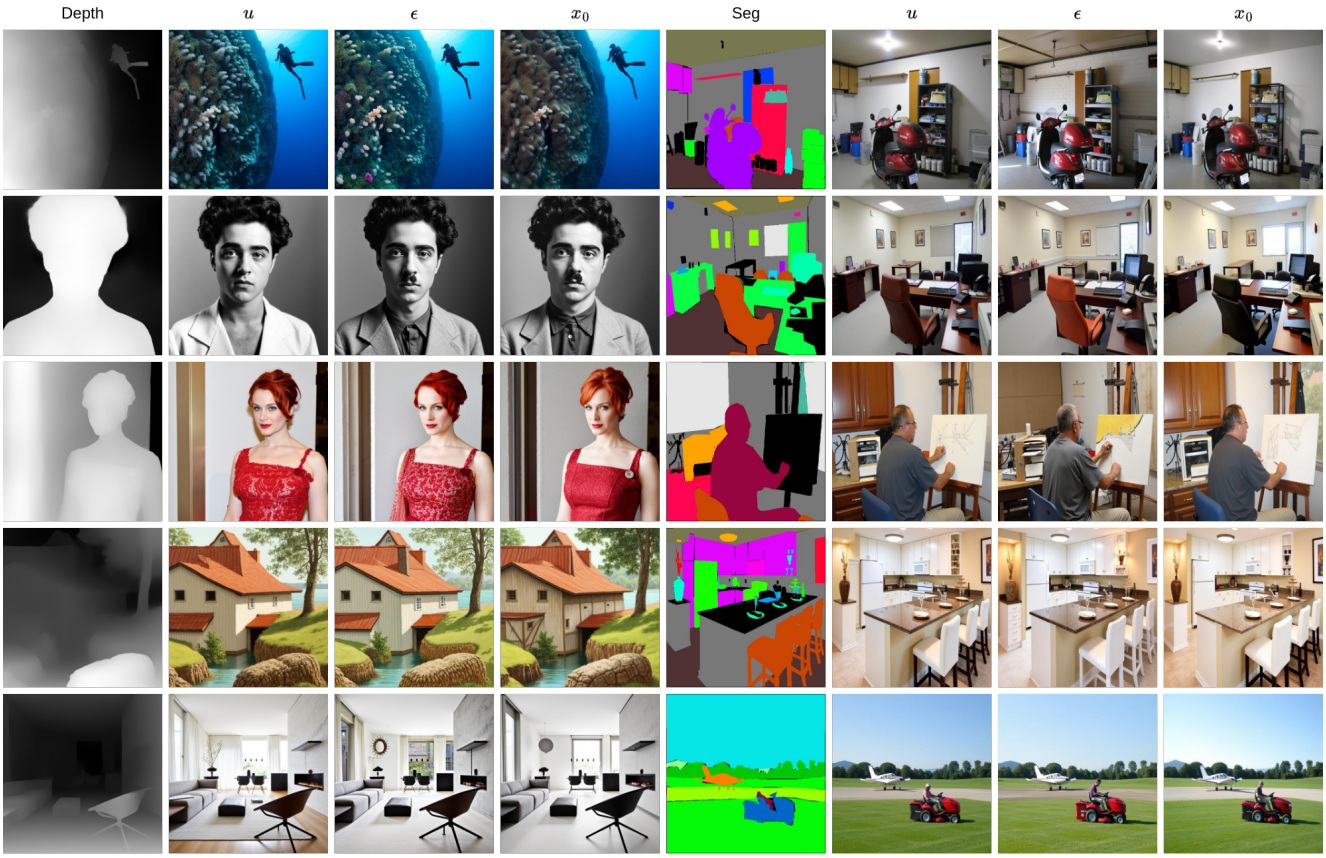


Figure 15. Qualitative results on depth and segmentation OminiControl with the three supervision signals after 40k training steps.

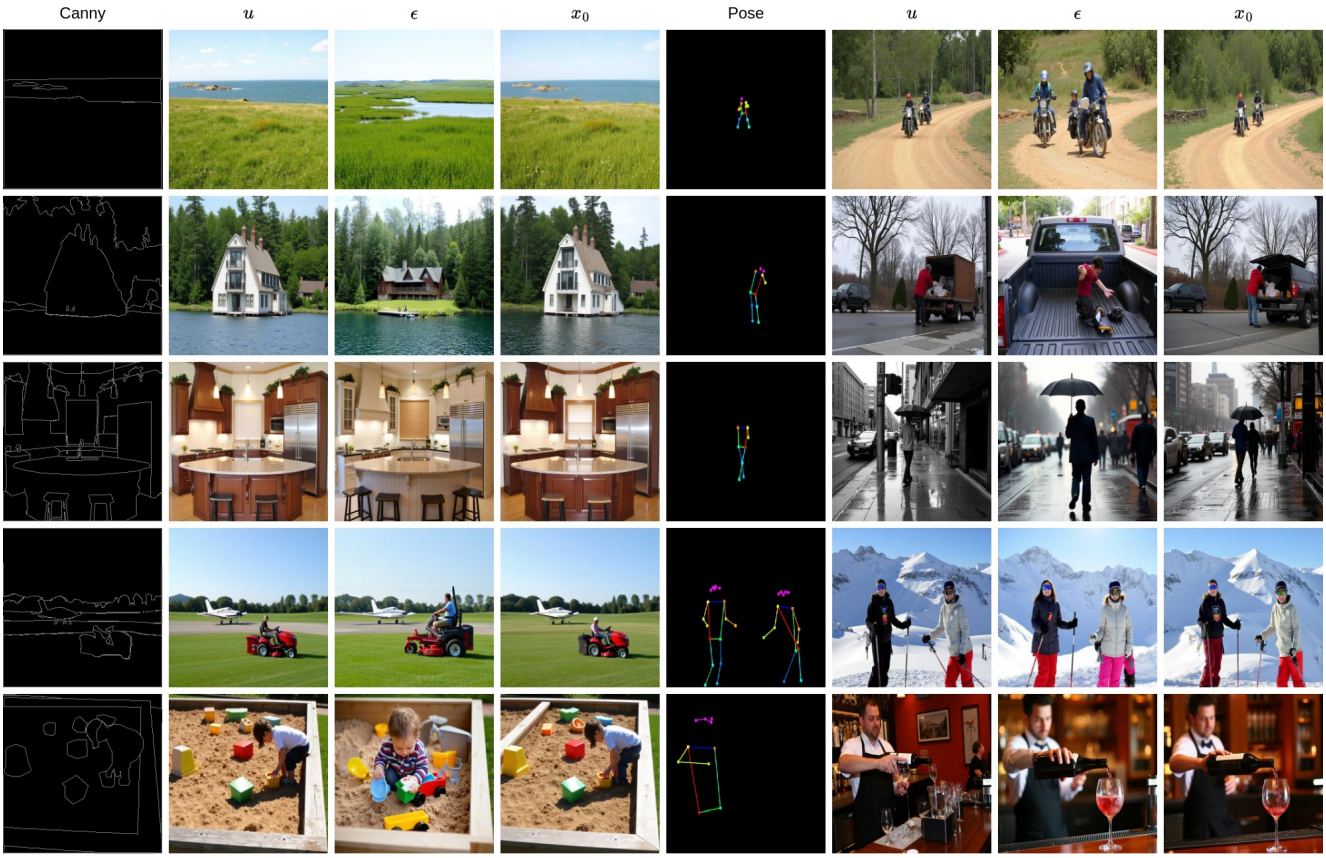


Figure 16. Qualitative results on Canny edge and pose OminiControl with the three supervision signals after 5k training steps.

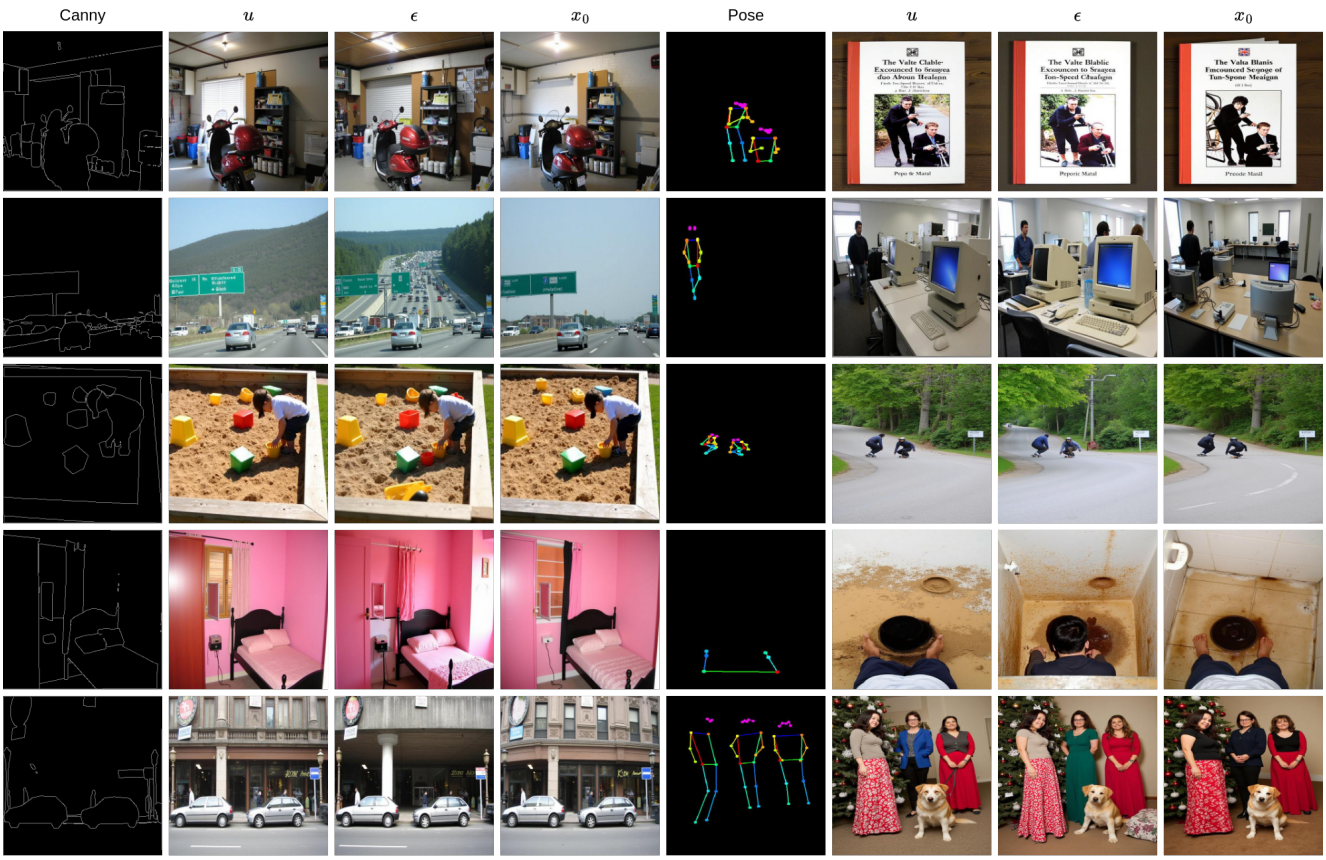


Figure 17. Qualitative results on Canny edge and pose OminiControl with the three supervision signals after 40k training steps.