LISA: LIKELIHOOD SCORE ALIGNMENT FOR VISUAL-CONDITION CONTROLLABLE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The prevalent *dual-branch paradigm*, *i.e.*, training a side network to encode visual conditions and fusing its intermediate-layer features to a frozen pretrained main network, has shown remarkable success in visual-condition controllable generation. Despite its widespread adoption, the role of the side branch and its training efficiency remain underexplored. In this paper, we first revisit this mainstream paradigm through the lens of score-based generative modeling: 1) The main network preserves visual perceptual quality by providing a prior unconditional score. 2) The side network steers conditional control by implicitly contributing a likelihood score. Guided by this perspective, we propose *LIkelihood Score Alignment (**LISA**)*, an effective regularization method that explicitly aligns the intermediate features of the side network with an approximated likelihood score. Specifically, we first hook features from a designated layer of the side network and project them into the score latent space by a lightweight decoder. Then, we construct an approximated likelihood score target and calculate the distance between the decoder's output and this target as an additional regularization loss. Finally, we jointly optimize the side network and decoder with both standard diffusion loss and our regularization loss. Experiments across various image/video tasks, architectures, and diffusion/flow models demonstrate that LISA can not only consistently accelerate the training convergence and improve final synthetic results, but also encourage the side network's features to be more disentangled for conditional modeling with negligible additional training cost and zero extra inference cost.

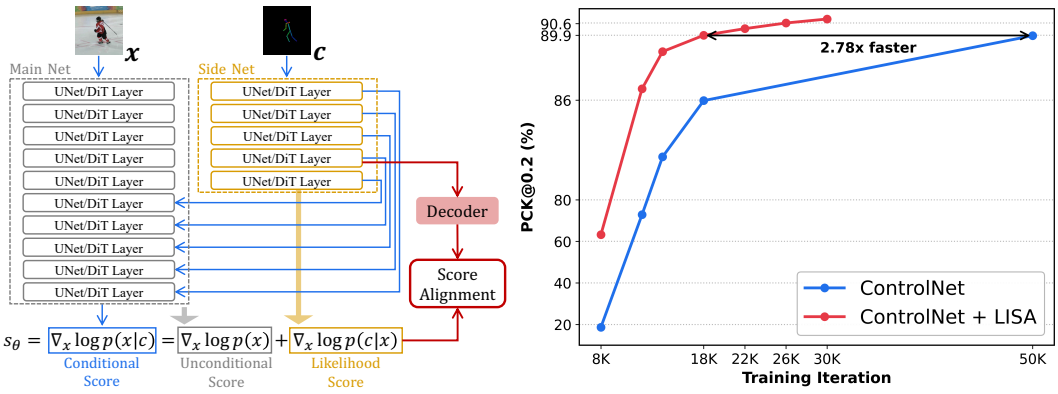


Figure 1: **Likelihood score alignment (LISA) can improve training convergence and synthetic quality**. Our framework, LISA, explicitly decomposes roles within the dual-branch paradigm: the main network and side network are responsible for the *unconditional* and *likelihood score*, respectively. By aligning a certain feature of the side network with an approximated likelihood score via a lightweight decoder, LISA can achieve $> 2.78\times$ faster convergence (*e.g.*, as in ControlNet).

1 INTRODUCTION

Recent advances in diffusion (Ho et al., 2020; Song et al., 2020) and flow matching models (Liu et al., 2022; Lipman et al., 2022) show remarkable visual generation capability. In particular, unconditional and text-conditioned generation tasks have been addressed quite well by billion-parameter

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

models (Esser et al., 2024; Labs, 2025; Wan et al., 2025; HaCohen et al., 2026) trained on large-scale and easy-to-collect training data. However, the increasing application requirements introduce a more challenging scenario (Batzolis et al., 2021): *Visual-condition Controllable Generation*, *i.e.*, integrating visual-modality conditions, especially spatial conditions (*e.g.*, pose, segmentation, and depth maps) for more fine-grained, structurally controllable image and video generation.

To achieve this goal, prior studies (Zhang et al., 2023; Mou et al., 2024; Rowles, 2023) resort to a **dual-branch paradigm** (*c.f.*, Figure 1): 1) Freezing a pretrained diffusion (or flow matching) model as the *main network* backbone. 2) Training a condition encoder as the *side network* to adopt the condition input and output a set of intermediate features. 3) Integrating these features into the main network's original forward process to achieve conditional control. Under this paradigm, the follow-up study (Xie et al., 2026) extends representation alignment technology (Yu et al., 2024) to controllable generation, *i.e.*, aligning model features with a pretrained semantic encoder as an additional regularization to improve training efficiency. However, the dependence on external encoders bounds their performance to the chosen encoder.

In this paper, we revisit this paradigm through the lens of score-based generative modeling: each branch network implicitly plays decomposed roles, and the feature-level integration mechanism essentially tries to induce an augmented result: 1) The frozen main network does not adopt the condition input, *i.e.*, it is mainly responsible for providing an *unconditional score* $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ to guarantee general perceptual quality. 2) The trainable side network encodes the condition $\boldsymbol{c}$ and learns to bridge the gap between the *conditional score* $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c})$ and the *unconditional score* $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, *i.e.*, achieving conditional control. 3) According to Bayes' rule, this residual gap is the *likelihood score*, *i.e.*, $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$.

Our approach. Based on the above analyses, we argue that the main challenge in the dual-branch paradigm lies in training the side network, which learns to provide the control signal and implicitly corresponds to a likelihood score. Motivated by this, we propose *LIkelihood Score Alignment* (LISA), an effective regularization technique that explicitly aligns the side network with an approximated likelihood score. Since the frozen main network is naturally an unconditional score predictor and the paired training data can provide the closed-form conditional score, we can estimate a likelihood score by calculating the difference between them. By aligning the intermediate features of the side network with this approximated score alongside standard generative training, LISA introduces an efficient prior supervision. This explicit constraint acts as a regularization loss, significantly accelerating convergence and improving overall synthesis performance (*c.f.*, Figure 1). Meanwhile, such regularization can encourage the side network's features to be more disentangled for conditional modeling, thereby naturally demonstrating better compositional control. It is worth noting that, compared with existing representation alignment, our LISA does not require external semantic encoders and achieves comparable improvements in training convergence and synthesis quality.

Specifically, during the standard generative training, we first hook features from a designated layer of the side network and feed them into a lightweight trainable decoder (usually around 0.1% of the size of the side network). This decoder only contains a few layers (*e.g.*, convolution, activation, and upsampling layers) to transform the intermediate features of the side network into a latent score space. Then, we calculate the distance between the decoder output and the likelihood score as a regularization loss, which is added to the diffusion loss as the final optimization objective. Notably, the additional training cost introduced by such regularization is almost negligible. During inference, we directly drop the decoder and use the trained side network for final conditional generation.

We evaluate the effectiveness of our LISA across various image and video conditions, including pose maps, depth maps, low-resolution images, segmentation maps, and pose videos. All experimental results have consistently indicated that LISA can significantly accelerate training convergence and bootstrap to a better synthetic quality (both perceptual quality and condition fidelity). Further evaluations, such as architecture-agnostic generalization and more challenging compositional controls, have verified that LISA is an effective and extensible solution for visual-condition controllable generation. In summary, our main contributions are as follows:

- We analyzed the mainstream dual-branch paradigm for visual-condition generation from a novel perspective, revealing that the side network lacks explicit regularization for its intended role.
- Based on the role decomposition, we proposed an effective likelihood alignment method, LISA, which regularizes the side network's intermediate output with an approximated likelihood score.

- Extensive experiments across diffusion models, network architectures, and tasks have demonstrated significant and consistent gains in both training convergence and synthesis quality.

2 PRELIMINARIES

We present a brief overview of diffusion and flow matching models via the unified perspective of the stochastic process (Jolicoeur-Martineau et al., 2021) and score matching (Song & Ermon, 2019).

The diffusion/flow models aim to capture the target data distribution p_0 by learning a transport process from a prior distribution p_T (*e.g.*, a Gaussian distribution) to p_0. To achieve that, a forward diffusion process from p_0 to p_T can be described with such a stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $t \in [0, T]$ is the time-index, $\boldsymbol{x}_0 \sim p_0$, $\boldsymbol{w}$ is a Brownian motion, $\boldsymbol{f}$ and g are the drift and diffusion coefficient. Meanwhile, the marginal distribution of $\boldsymbol{x}_t$ determined by this SDE is $p_t(\boldsymbol{x}_t)$. Then, we have a reversed SDE to describe the reversed process ($p_T \to p_0$) of the forward process:

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\boldsymbol{w}}, \tag{2}$$

where $\overline{\boldsymbol{w}}$ is the reversed Brownian motion. According to the Fokker–Planck equation (Maoutsa et al., 2020) (F-P Equation), we can convert this reversed SDE into an ordinary differential equation (ODE) that has the same marginal distribution $p_t(\boldsymbol{x}_t)$:

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right] \mathrm{d}t. \tag{3}$$

We can sample synthetic samples by solving the SDE in Eq. (2) or the ODE in Eq. (3). However, $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ (also known as the unconditional score) is not known. Thus, a parameterized network $s_\theta(\cdot)$ with parameters θ can be trained to predict it. The optimization target is:

$$\min_\theta \mathbb{E}_{\boldsymbol{x}_0, t, \boldsymbol{x}_t}\left[||s_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)||_2^2\right], \tag{4}$$

where $\boldsymbol{x}_0 \sim p_0$, $t \in [0, T]$ and $\boldsymbol{x}_t \sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$. $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is tractable since the conditional distribution is defined by the forward SDE in Eq. (1), and it has closed-form solutions. After training, s_θ can be used as a score predictor to replace $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ for solving Eq. (2) and Eq. (3).

3 LISA: LIKELIHOOD SCORE ALIGNMENT

Problem Formulation. Assume there exists an underlying joint distribution $(\boldsymbol{x}_0, \boldsymbol{c}) \sim p$, which describes the joint probability measurement between visual-modality conditions $\boldsymbol{c}$ and corresponding clean samples $\boldsymbol{x}_0$. For the conditional visual generation task, our objective is to use given $(\boldsymbol{x}_0, \boldsymbol{c})$ pairs to train a conditional score predictor $s(\boldsymbol{x}_t, \boldsymbol{c}, t)$. In the inference stage, we use $s(\boldsymbol{x}_t, \boldsymbol{c}, t)$ to transport a random noise $\boldsymbol{x}_T$ to a clean sample $\boldsymbol{x}_0$ conditioned on the given condition $\boldsymbol{c}$[1].

3.1 SCORE DECOMPOSITION OF THE DUAL-BRANCH PARADIGM

Standard Conditional Diffusion. Given the joint data distribution $\boldsymbol{x}_0, \boldsymbol{c}$, the noisy conditional distribution $p_t(\boldsymbol{x}_t|\boldsymbol{c})$ induced by the forward SDE of Eq. (1) is:

$$p_t(\boldsymbol{x}_t|\boldsymbol{c}) = \int p(\boldsymbol{x}_0|\boldsymbol{c}) p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_0. \tag{5}$$

Taking the gradient with respect to $\boldsymbol{x}_t$, the conditional score can be written as:

$$\begin{aligned}\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) = \frac{\nabla_{\boldsymbol{x}_t} p_t(\boldsymbol{x}_t|\boldsymbol{c})}{p_t(\boldsymbol{x}_t|\boldsymbol{c})} &= \int p_t(\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{c}) \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_0 \\ &= \mathbb{E}_{\boldsymbol{x}_0|\boldsymbol{x}_t, c}\left[\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right].\end{aligned} \tag{6}$$

[1]For clarity, we omit text prompts in notations, and when needed they can be regarded as extra conditions.

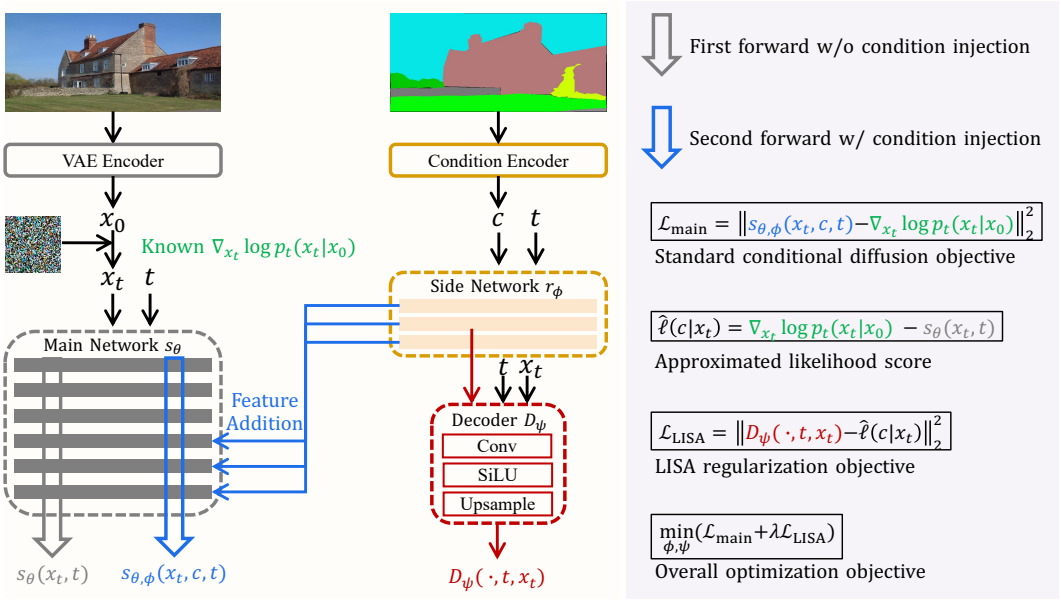


Figure 2: **The framework of LISA**. The first forward w/o condition injection provides the unconditional score $s_\theta(\boldsymbol{x}_t, t)$. By subtracting it from the known $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|x_0)$, we can construct an approximated likelihood score $\hat{\ell}_t(\boldsymbol{x}_t, c)$. In the second forward w/ condition injection, we align the features of the side network with the $\hat{\ell}_t(\boldsymbol{x}_t, c)$ via a decoder as an extra regularization objective.

Eq. (6) shows that the conditional score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c})$ is the expectation of the clean-sample score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$. Therefore, although $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is conditioned on the clean sample $\boldsymbol{x}_0$ rather than directly on $\boldsymbol{c}$, it provides an unbiased supervision for learning the conditional score.

Thus, for the dual-branch paradigm, a conditional score predictor for visual-condition generation is trained with the standard denoising score matching objective:

$$\mathcal{L}_{\text{main}} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{c},t,\boldsymbol{x}_t}\left[\|s_{\theta,\phi}(\boldsymbol{x}_t, \boldsymbol{c}, t) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\|_2^2\right], \tag{7}$$

where θ and ϕ denote the parameters of the frozen pretrained main network and the trainable side network. The optimal solution of Eq. (7) is $s^*_{\theta,\phi}(\boldsymbol{x}_t, \boldsymbol{c}, t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c})$.

Dual-branch Decomposition. We now decompose this conditional score with Bayes' rule:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t), \tag{8}$$

where the term $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c})$ disappears because $\log p_t(\boldsymbol{c})$ is independent of $\boldsymbol{x}_t$. Eq. (8) indicates that conditional generation can be interpreted as the combination of two scores: the unconditional score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ and the likelihood score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t)$.

This decomposition naturally matches the dual-branch paradigm (*c.f.*, Figure 2). We denote the side network by r^i_ϕ and its intermediate features by $\{r^i_\phi(\boldsymbol{x}_t, \boldsymbol{c}, t)\}_{i=1}^L$, where L is the number of selected side features. The full conditional score predictor can be written as:

$$s_{\theta,\phi}(\boldsymbol{x}_t, \boldsymbol{c}, t) = \mathcal{S}_\theta\left(\boldsymbol{x}_t, t; \{r^i_\phi(\boldsymbol{x}_t, \boldsymbol{c}, t)\}_{i=1}^L\right), \tag{9}$$

where disabling the side features gives the frozen main-network prediction $s_\theta(\boldsymbol{x}_t, t) = \mathcal{S}_\theta(\boldsymbol{x}_t, t; \emptyset)$. Since s_θ has been pretrained to approximate $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$, the trainable side network is implicitly responsible for supplying the residual correction from the unconditional score to the conditional score, which is called the likelihood score:

$$s_{\theta,\phi}(\boldsymbol{x}_t, \boldsymbol{c}, t) - s_\theta(\boldsymbol{x}_t, t) \approx \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t). \tag{10}$$

However, the standard objective in Eq. (7) supervises only the final prediction, leaving this likelihood-score role of the side network implicit. This motivates us to explicitly align side-network features with an approximated likelihood score.

3.2 Alignment with Approximated Likelihood Score

Approximated Likelihood Score Construction. To explicitly supervise the side network during training, we first need to obtain a likelihood-score target. According to Bayes' rule, the likelihood score can be written as the difference between the conditional and unconditional scores:

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t). \tag{11}$$

Although the conditional score $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c})$ is intractable, we can use the denoising target $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ to provide a single-sample supervision signal whose expectation equals the conditional score (*c.f.*, Eq. (6)).

Meanwhile, the pretrained main network is naturally an unconditional score predictor. To this end, we additionally forward the main network without any condition injection to obtain $s_\theta(\boldsymbol{x}_t, t) \approx \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ (*c.f.*, Figure 2). Substituting two estimates into Eq. (11):

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c}|\boldsymbol{x}_t) \approx \mathbb{E}_{\boldsymbol{x}_0|\boldsymbol{x}_t,c} \left[\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) - s_\theta(\boldsymbol{x}_t, t)\right]. \tag{12}$$

This motivates the following sample-wise approximated likelihood score:

$$\widehat{\ell}_t(\boldsymbol{c}|\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) - s_\theta(\boldsymbol{x}_t, t). \tag{13}$$

Therefore, Eq. (13) provides a practical and efficient target for likelihood-score alignment.

Likelihood Score Alignment. To align the side network with this target, we select a certain feature $r^k_\phi(\boldsymbol{x}_t, \boldsymbol{c}, t)$ (the k-th layer's output) from the side network before integrating. We feed it into a lightweight decoder $\mathcal{D}_\psi$ composed of convolution, activation, and upsampling layers:

$$\widetilde{\ell}^k_\psi(\boldsymbol{c}|\boldsymbol{x}_t) = \mathcal{D}_\psi\left(r^k_\phi(\boldsymbol{x}_t, \boldsymbol{c}, t), t, \boldsymbol{x}_t\right). \tag{14}$$

The decoder maps the selected side feature into the same latent score space as the diffusion target. We then impose the LISA regularization loss:

$$\mathcal{L}_{\text{LISA}} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{c},t,\boldsymbol{x}_t}\left[\left\|\widetilde{\ell}^k_\psi(\boldsymbol{c}|\boldsymbol{x}_t) - \text{sg}\left[\widehat{\ell}_t(\boldsymbol{c}|\boldsymbol{x}_t)\right]\right\|_2^2\right], \tag{15}$$

where $\text{sg}[\cdot]$ denotes stop-gradient operation. Combining the standard diffusion loss, the final objective is

$$\phi^*, \psi^* = \arg\min_{\phi,\psi}(\mathcal{L}_{\text{main}} + \lambda\mathcal{L}_{\text{LISA}}), \tag{16}$$

where λ controls the strength of likelihood-score alignment. During training, the parameters of the main network are frozen. We optimize the side network and the lightweight decoder jointly. The standard loss $\mathcal{L}_{\text{main}}$ supervises the final conditional prediction, while $\mathcal{L}_{\text{LISA}}$ provides a direct auxiliary gradient to the side network. Therefore, LISA makes the side network learn its intended likelihood-score role more explicitly and efficiently. During inference, the auxiliary decoder is discarded, and the trained side network is used exactly in the same way as the original architecture.

4 Experiments

4.1 Main Results

To verify the effectiveness, we compared LISA with the vanilla representative dual-branch baselines: T2I-Adapter (Mou et al., 2024), ControlNet (Zhang et al., 2023), and ControlNet+REPA (Xie et al., 2026) across four types of conditional image generation tasks: Pose (Cao et al., 2017), ADE20K Segmentation (Zhou et al., 2017), Depth (Ranftl et al., 2020) maps and low-resolution image conditional generation. We took SDXL-1.0 (Podell et al., 2024) and Stable Diffusion 2.1 (Rombach et al., 2022) as the pretrained diffusion model for T2I-Adapter and ControlNet, respectively. For a fair comparison, we used AdamW (Loshchilov & Hutter, 2017) optimizer with a $1e-5$ learning rate and maintained all the hyperparameters the same as the baseline. For the REPA implementation, we followed DIvControl (Xie et al., 2026): align the feature (same layer as our LISA) with DINOv2-B (Oquab et al., 2023) using a regularization weight of 0.05.

Metrics. For four image-conditioned generation tasks, we adopted the Frechet Inception Distance (*FID*) (Heusel et al., 2017), which quantifies the distributional similarity between synthetic and

Table 1: Comparisons with dual-branch baselines across four conditional image generation tasks.

Method	Pose Map				Segmentation Map			
	Iter.	FID$_\downarrow$	CLIP$_\uparrow$	PCK(%)$_\uparrow$	Iter.	FID$_\downarrow$	CLIP$_\uparrow$	mIoU(%)$_\uparrow$
T2I-Adapter	10K	59.10	32.27	84.85	10K	32.12	31.67	29.07
+ LISA (ours)	10K	**58.12**	**32.30**	**85.94**	10K	**32.07**	**31.70**	**29.53**
ControlNet	10K	56.37	**31.47**	19.38	10K	34.98	**31.96**	10.10
+ LISA (ours)	10K	**56.28**	31.39	**83.02**	10K	**30.68**	31.86	**28.30**
ControlNet	30K	58.54	30.87	89.82	30K	32.63	31.39	34.52
+ REPA	18K	62.23	30.81	**92.75**	25K	37.72	30.88	34.34
+ LISA (ours)	18K	**56.50**	**31.07**	89.90	25K	**32.51**	**31.46**	**34.88**
Method	Low-resolution Image				Depth Map			
	Iter.	FID$_\downarrow$	PSNR$_\uparrow$	LPIPS$_\downarrow$	Iter.	FID$_\downarrow$	CLIP$_\uparrow$	RMSE$_\downarrow$
T2I-Adapter	5K	31.76	16.17	0.457	5K	66.31	29.59	0.125
+ LISA (ours)	5K	**31.64**	**16.26**	**0.456**	5K	**62.70**	**29.77**	**0.121**
ControlNet	4K	44.77	10.58	0.740	2K	67.68	28.10	0.162
+ LISA (ours)	4K	**34.47**	**15.48**	**0.506**	2K	**65.23**	**28.13**	**0.153**
ControlNet	20K	23.96	16.89	0.426	10K	77.84	25.61	0.120
+ REPA	15K	28.84	15.98	0.464	4K	72.65	**27.29**	0.122
+ LISA (ours)	15K	**23.87**	**17.51**	**0.425**	4K	**66.75**	26.80	**0.114**

ground-truth images. Besides, for the pose-conditioned task, we used averaged CLIP similarity (*CLIP*) (Radford et al., 2021) between given text prompts and generated images to quantify the text condition following performance, as well as Percentage of Correct Keypoints (Yang & Ramanan, 2011) at the threshold of 0.2 (*PCK*) to quantify the pose condition following performance. For the segmentation-conditioned task, we used *CLIP* and mean Intersection over Union (*mIoU*) (Everingham et al., 2010) to quantify the segmentation condition following performance. For the low-resolution-conditioned task, we used Peak Signal-to-Noise Ratio (*PSNR*) and Learned Perceptual Image Patch Similarity (*LPIPS*) (Zhang et al., 2018), which focus on image-level similarity between ground truth and generated images to quantify the effectiveness. For the depth-conditioned task, we used *CLIP* and Root Mean Square Error (*RMSE*) (Eigen et al., 2014) to quantify the depth-conditioned following performance.

Quantitative Results. As shown in Table 1: (1) LISA consistently improves the condition-following ability across all four image tasks. In the early training stage, LISA achieves substantial gains in structure-related metrics, *e.g.*, improving *PCK* from 19.38 to 83.02 for pose-conditioned image generation relative to ControlNet. (2) LISA achieves better performance with fewer training iterations, demonstrating improved training efficiency. For example, in depth-conditioned image generation, LISA trained for only 4K iterations obtains better *FID*, *CLIP*, and *RMSE* than ControlNet trained for 10K iterations. (3) Compared with REPA, LISA achieved comparable performance without depending on any extra pretrained models. Overall, these results indicate that LISA not only enhances the fidelity of synthesis but also accelerates convergence.

Qualitative Results. We also gave qualitative comparisons in Figure 3. We can see that LISA shows better condition-following capability across various settings, as well as a decent visual quality (more natural with fewer artifacts). For example, in the second pose-conditioned example, ControlNet generated an image with the person inverted front-to-back, while ours produced the correct pose.

4.2 Ablation Study

We ablated two main hyperparameters: the feature depth used for alignment and the weight λ (*c.f.*, Eq. (16)). We used the pose-conditioned image generation task and ControlNet+LISA (with 18K training iterations) as the default setting. Besides, we provided a computational overhead analysis.

Alignment Depth. We studied the effect of the alignment depth while fixing $\lambda = 0.2$. As shown in Table 2, introducing the alignment module consistently improves the pose consistency measured by *PCK* compared with the baseline. Specifically, using alignment depths of 2 and 8 improves *PCK* from 85.97% to 88.03% and 88.06%, respectively, while maintaining comparable *FID* and *CLIP*.

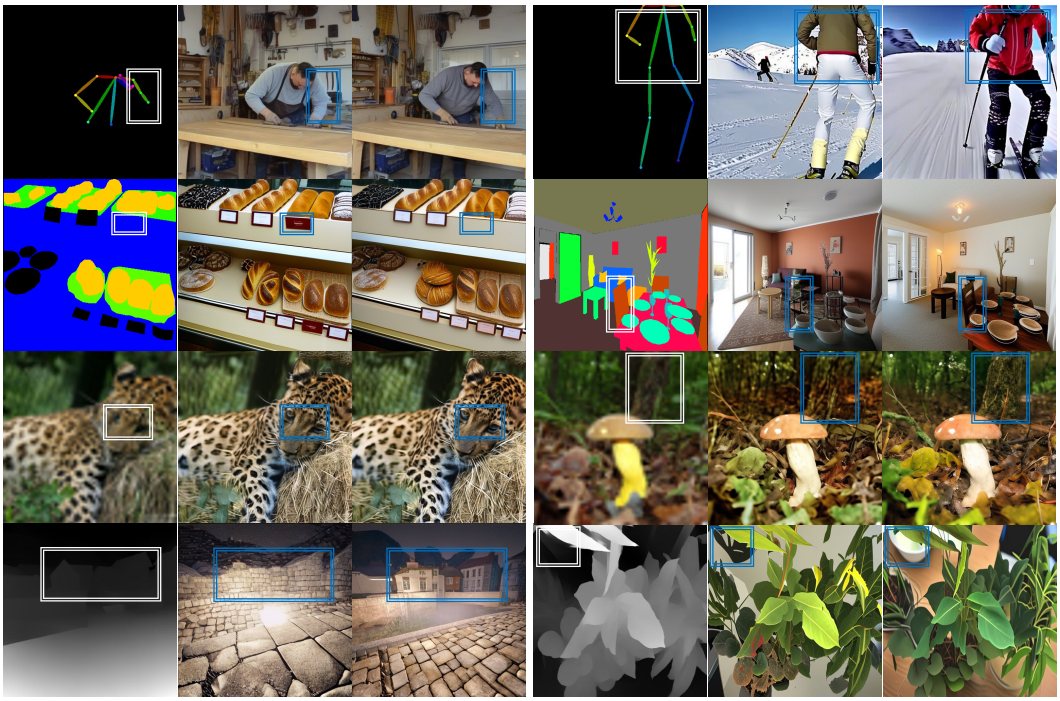


Figure 3: **Qualitative examples across four image-condition generation tasks.** LISA shows better condition-following performance (see highlighted parts in blue boxes).

Among different depths, setting the depth to 5 achieves the best *PCK* of 89.90%. A shallower alignment may be insufficient to fully capture structural correspondence, whereas an overly deep alignment does not bring further improvement and may introduce redundant constraints. Therefore, we adopt an alignment depth of 5 in our final implementations for all four conditional tasks.

Effect of λ. We further analyzed the influence of the loss weight λ (*c.f.*, Eq. 16) with the alignment depth fixed to 5. When $\lambda = 0.1$, the model obtains a lower *PCK* of 86.19%, suggesting that a weak alignment constraint is insufficient to guide pose-consistent generation. Increasing λ to 0.5 slightly improves FID to 56.34, but the *PCK* drops to 87.83%, indicating that an overly strong alignment constraint may hurt structural matching. In contrast, $\lambda = 0.2$ achieves the best overall balance, yielding the highest *PCK* of 89.90%. Thus, we set $\lambda = 0.2$ as the default configuration.

Table 2: **Ablation of alignment depth and λ.** The first row is the baseline.

Depth	λ	FID	CLIP	PCK(%)
-	-	56.71	31.24	85.97
2	0.2	56.40	30.96	88.03
8	0.2	56.75	31.14	88.06
5	0.2	56.50	31.07	89.90
5	0.5	56.34	31.08	87.83
5	0.1	57.31	31.08	86.19

Computational Overhead Analysis. We compared the computational cost on 8 H20 GPUs between ControlNet and our LISA in Table 3. LISA introduces only a negligible number of additional parameters, increasing the model size from 364.2M to 364.6M, *i.e.*, about 0.1% extra parameters. Both methods require the same GPU memory consumption of 21G, showing that the proposed alignment module does not increase the memory footprint. In terms of training time per iteration, LISA takes 2.3s compared with 2.1s for ControlNet, introducing only 0.2s additional latency. As a highlight, during the inference stage, LISA directly drops the decoder, and thus the computational cost is completely the same as naive ControlNet. These results indicate that LISA improves performance with minimal computational overhead, making it efficient and practical for deployment.

Table 3: Computational overhead comparisons.

	#Params	GPU	Time
ControlNet	364.2M	21G	2.1s
+LISA	364.6M	21G	2.3s

4.3 Generalization Study

Extend to Flow and DiT. Since our main results are based on the U-Net (Ronneberger et al., 2015) along with the Variance-Preserving (VP) SDE, *i.e.*, Stable Diffusion v2.1 (Rombach et al., 2022),

Table 4: Compatible with Stable Diffusion 3.

Method	Iter.	FID↓	CLIP↑	mIoU(%)↑
ControlNet	1K	32.08	**32.07**	20.81
+ LISA	1K	**31.87**	32.03	**22.64**
ControlNet	5K	31.69	32.02	26.01
+ LISA	5K	**31.47**	**32.03**	**27.58**

Table 5: Compatible with Video Generation.

Method	Iter.	FVD↓	SSIM↑	LPIPS↓	PCK(%)↑
SVD Control	5K	10.57	0.852	19.07	30.22
+ LISA	5K	**7.85**	**0.882**	**10.62**	**57.00**
SVD Control	30K	7.79	0.902	9.28	90.53
+ LISA	30K	**6.89**	**0.905**	**8.74**	**91.86**



Figure 4: **Qualitative comparisons on pose-conditioned video generation.** LISA shows better condition-following performance in the latter frame (see the highlighted parts in the blue boxes).

we further test our effectiveness on Diffusion Transformer (Peebles & Xie, 2023) (DiT) along with Optimal Transport Flow Matching (Liu et al., 2022; Lipman et al., 2022) (OT-FM), *e.g.*, Stable Diffusion v3-medium (Esser et al., 2024). To this end, we conducted the same segmentation-conditioned experiments as in Section 4.1 with the 8-th layer output.

As shown in Table 4, we can see that at the early training stage, *i.e.*, 1K iterations, LISA reduces FID from 32.08 to 31.87 and improves mIoU from 20.81% to 22.64%, while maintaining a comparable CLIP score. When trained for 5K iterations, LISA further improves all metrics, achieving lower FID, higher CLIP score, and higher mIoU compared with the ControlNet baseline. These results demonstrate that LISA is not limited to the U-Net architecture or VP-SDE formulation, but can also generalize well to diffusion transformers trained with flow matching objectives.

Extend to Controllable Video Generation. To further verify our generalization for controllable video generation, we also compared our LISA with SVD Temporal ControlNet (SVD Control) (Rowles, 2023) based on a pretrained image-to-video model, *i.e.*, Stable Video Diffusion (Blattmann et al., 2023) for the pose-guided video generation task on the UBC Fashion dataset (Zablotskaia et al., 2019). We used the same hyperparameters as in Section 4.1. There are four reported metrics: Frechet Video Distance (Unterthiner et al., 2018) (*FVD*), which can indicate distributional differences between the ground-truth videos and the synthetic videos, frame-level *SSIM* (Wang et al., 2004), *LPIPS*, and *PCK*.

As shown in Table 5, our LISA also generalizes well to conditional video generation. At 5K iterations, LISA significantly improves all metrics, *e.g.*, reducing *FVD* from 10.57 to 7.85 while increasing *PCK* from 30.22% to 57.00%. At 30K iterations, LISA further maintains consistent gains over the ControlNet baseline across all metrics. These improvements demonstrate that LISA can effectively enhance both generation quality and condition controllability for video diffusion models with good generalization. We also provide the visualization case in Figure 4.

4.4 Bonus: Compositional-condition generation

Since the alignment between the feature and the likelihood score can encourage the side network to model the condition more independently, the features under such regularization potentially should show more disentangled control performance. To verify this, we further investigated whether LISA benefits the composition of multiple visual conditions. To fairly evaluate the compositional ability, we took independently trained single-condition side networks for pose and segmentation, where ControlNet and LISA show comparable performance under the corresponding single-condition settings. During inference, we directly composed the two conditions by summing their injected features at each corresponding layer, resulting in a pose-plus-segmentation conditioned generation setting.

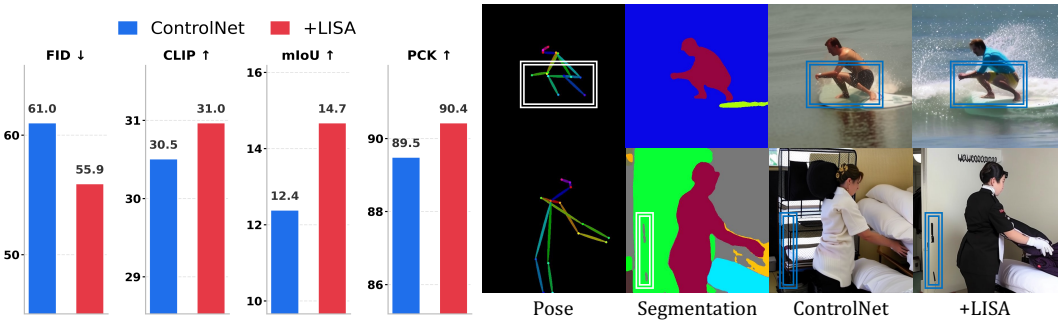


Figure 5: **Quantitative (left) and qualitative (right) results of compositional-condition generation.** Benefiting from the explicit role decomposition, LISA shows better composition properties.

As shown in Figure 5, LISA demonstrates stronger compositional generation ability than the naive ControlNet baseline with both quantitative and qualitative evidence. These results suggest that the explicit role decomposition and likelihood-score alignment introduced by LISA help disentangle the main network and side networks, making the learned condition representations more composable and thus more suitable for multi-condition controllable generation as an extra bonus.

5 RELATED WORK

Conditional Visual Generation. Besides the class and text (Dhariwal & Nichol, 2021; Ho & Salimans, 2022), the visual-modality conditions can provide spatial-structure guidance for generation. Composer (Huang et al., 2023) trains a network that can adopt multi-modality conditions from scratch, thus can naturally achieve conditional control. However, the training cost limits its extension efficiency when facing new condition types. To this end, works (Li et al., 2023; Zhang et al., 2023; Mou et al., 2024; Rowles, 2023; Choi et al., 2025) propose to freeze the pretrained diffusion model and fine-tune a side network (*e.g.*, a copied encoder or condition adapter) and inject the condition feature into the original pretrained model. To further improve the controllability and efficiency, ControlNeXt (Peng et al., 2024) aligns denoising distributions with the control features and ControlNet++ (Li et al., 2024a) incorporates reinforcement learning for post-training. Moreover, Uni-ControlNet (Zhao et al., 2023) built a ControlNet that can model various visual conditions by training unified control adapters with extensive training data. Our method motivates the incorporation of an additional alignment to further improve efficiency.

Training Diffusion Models with Regularizations. The vanilla diffusion and flow matching models regress the target (*e.g.*, noise, score, and velocity) as the main training loss. On top of that, some works propose adding an extra regularization during training to accelerate convergence. Studies (Yu et al., 2024; Pernias et al., 2024; Li et al., 2024b) leverage pretrained semantic visual encoders to help diffusion models' efficiency and final performance. ΔFM (Stoica et al., 2025) constructs a contrastive objective to regularize the flow trajectories and accelerate the training of the flow model. In video generation, works (Wu et al., 2025; Huang et al., 2025; Zhang et al., 2025) incorporate pretrained 3-D models' features or additional proxy 3-D tasks for training video diffusion models, enhancing consistency in synthetic videos. Our method shares a similar motivation with the above works, but differs in regularizing with the conditional probabilistic and score perspectives.

6 CONCLUSION

In this paper, we focus on the dual-branch paradigm for visual-condition controllable generation. Based on the role decomposition of its main and side networks from the score perspective, we propose LISA, which aligns the middle feature of the side network with a constructed likelihood score. By adding such a simple extra regularization objective, LISA can significantly accelerate training and bootstrap better synthetic results on perceptual quality and condition fidelity. Extensive ablations verify our consistent effectiveness and compatibility with U-Net/DiT architectures, diffusion/flow models, and image/video tasks. Besides, we found that LISA naturally shows better potential for compositional control, benefiting from the decomposition. In the future, we will extend the LISA regularization to practical applications and more general conditional generation scenarios.

AI Use Statement. Generative AI tools were used to polish the manuscript and improve the clarity and readability of the writing. We explicitly emphasize that these technologies were strictly limited to language editing and played no role in research conceptualization, data generation, experimental design, or the formulation of core scientific claims. Furthermore, every AI-assisted modification underwent rigorous manual scrutiny to ensure it accurately reflected our original scientific intent. The authors maintain full accountability and take complete responsibility for the entirety of the final manuscript, including all findings, data, and conclusions.

REFERENCES

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

Yisol Choi, Sangkyung Kwak, Sihyun Yu, Hyungwon Choi, and Jinwoo Shin. Controllable human image generation with personalized multi-garments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28736–28747, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Chun-Hao Paul Huang, Niloy Mitra, Hyeonho Jeong, Jae Shin Yoon, and Duygu Ceylan. Jog3r: Towards 3d-consistent video generators. *arXiv preprint arXiv:2501.01409*, 2025.

Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback: Project page: liming-ai. github. io/controlnet_plus_plus. In *European Conference on Computer Vision*, pp. 129–147. Springer, 2024a.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37: 125441–125468, 2024b.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22511–22521, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, 2020.

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 *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Bohao Peng, Jian Wang, Yuechen Zhang, Wenbo Li, Ming-Chang Yang, and Jiaya Jia. Controlnext: Powerful and efficient control for image and video generation. *arXiv preprint arXiv:2408.06070*, 2024.

Pablo Pernias, Dominic Rampas, Mats L Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 25097–25109, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

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*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Ciara Rowles. Stable Video Diffusion Temporal Controlnet. https://github.com/CiaraStrawberry/svd-temporal-controlnet, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1185–1194, 2025.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. *arXiv preprint arXiv:2507.07982*, 2025.

Yucheng Xie, Fu Feng, Ruixiao Shi, Jing Wang, Yong Rui, and Xin Geng. Divcontrol: Knowledge diversion for controllable image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27108–27116, 2026.

Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pp. 1385–1392. IEEE, 2011.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*, 2019.

Ke Zhang, Yiqun Mei, Jiacong Xu, and Vishal M Patel. Endless world: Real-time 3d-aware long video generation. *arXiv preprint arXiv:2512.12430*, 2025.

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*, pp. 3836–3847, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in neural information processing systems*, 36:11127–11150, 2023.

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*, pp. 633–641, 2017.

Appendix

This appendix is organized as follows:

- Section A.1 gives the implementation details of our method and reproduction details of the compared baselines.
- Section A.2 provides the derivation of the compatibility of velocity-based Optimal Transport Flow Matching (OT-FM) and noise-based Variance-Preserving SDE (VP-SDE).
- Section A.3 provides additional qualitative results.

A.1 Implementation Details

We implement our method using PyTorch, Hugging Face Diffusers, and Accelerate, with Stable Diffusion 2.1-base (SD2.1) and Stable Video Diffusion (SVD) as the pretrained backbones. We freeze the pretrained denoising backbone, VAE, and text or image encoders, and jointly optimize ControlNet and a lightweight auxiliary decoder. The decoder receives intermediate ControlNet features before the zero-initialized output projection and predicts the residual between the denoising target and the frozen backbone prediction without control. It is used only during training and introduces no additional inference overhead.

Unless otherwise specified, we extract features from `down_5` (zero-based index) and set the auxiliary loss weight to 0.2. All experiments are conducted on eight NVIDIA H20 GPUs. We optimize the trainable parameters using AdamW with a constant learning rate of 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 0.01. Training uses FP16 mixed precision, gradient checkpointing, and gradient clipping with a maximum norm of 1.0. Image experiments use a resolution of 512×512, with a per-GPU batch size of 8 and four gradient accumulation steps, yielding an effective batch size of 256.

For UBCFashion, we train SVD on 14-frame clips sampled at a frame interval of 4, with a resolution of 768×512 (height $\times$ width), a motion bucket ID of 127, and an effective batch size of 8.

At inference, SD2.1 uses the DPM-Solver multistep scheduler with 20 sampling steps and a classifier-free guidance scale of 7.5.

A.2 The Derivation of the Compatibility of OT-FM and VP-SDE

We developed our decomposition of the dual-branch paradigm from the score-based perspective in the main paper. This decomposition is also compatible with diffusion and flow matching models that predict noise and velocity. In this section, we provide the equivalent derivations for v-based Optimal Transport Flow Matching (OT-FM) and ϵ-based Variance-Preserving SDE (VP-SDE).

Recall that the conditional score can be decomposed as

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t), \qquad \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t) := \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \tag{A17}$$

Accordingly, the frozen main network provides $s_\theta(\boldsymbol{x}_t, t)$, while the side network induces the residual correction

$$s_{\theta,\phi}(\boldsymbol{x}_t, \boldsymbol{c}, t) - s_\theta(\boldsymbol{x}_t, t) \approx \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \tag{A18}$$

This relation becomes exact when both predictors recover their corresponding population scores. We omit prediction arguments when the context is clear, and use the subscripts θ and (θ, ϕ) for predictions without and with side-network condition injection, respectively.

For a Gaussian forward process $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent of $(\boldsymbol{x}_0, \boldsymbol{c})$, the sample-wise approximated likelihood score used by LISA is

$$\widehat{\ell}_t(\boldsymbol{c} \mid \boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0) - s_\theta = -\frac{\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0}{\sigma_t^2} - s_\theta. \tag{A19}$$

Its conditional expectation given $(\boldsymbol{x}_t, \boldsymbol{c})$ approximates $\ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t)$, with equality for an exact main-network score. All conversions below are considered at interior times where $\alpha_t, \sigma_t > 0$.

A.2.1 OT-FM with v-prediction.

In Flow Matching with this Gaussian path, the velocity field $v_\theta(\boldsymbol{x}_t, t)$ corresponds to the probability-flow ODE (PF-ODE) drift. Thus, we have the relations

$$v_\theta = \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t) s_\theta, \qquad v_{\theta,\phi} = \boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t) s_{\theta,\phi}. \tag{A20}$$

Substituting these relations into Eq. (A18), we rewrite the decomposition in terms of velocity:

$$\begin{aligned} v_{\theta,\phi} - v_\theta &= -\frac{1}{2} g^2(t) \left(s_{\theta,\phi} - s_\theta\right) \\ &\approx -\frac{1}{2} g^2(t) \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \end{aligned} \tag{A21}$$

For OT-FM, the trajectory is defined by $\sigma_t = 1 - \alpha_t$. Under a linear schedule where $\alpha_t = 1 - t$ and $\sigma_t = t$, we have $\dot{\alpha}_t = -1$ and $\dot{\sigma}_t = 1$. This follows the data-to-noise time convention of the main paper; generation integrates from $t = 1$ toward $t = 0$. The corresponding linear SDE with the same Gaussian perturbation kernels has drift and squared diffusion coefficients

$$\boldsymbol{f}(\boldsymbol{x}_t, t) = \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x}_t = -\frac{\boldsymbol{x}_t}{1 - t}, \tag{A22}$$

$$g^2(t) = 2\sigma_t \dot{\sigma}_t - 2\frac{\dot{\alpha}_t}{\alpha_t} \sigma_t^2 = -2\dot{\alpha}_t \frac{\sigma_t}{\alpha_t} = \frac{2t}{1 - t}. \tag{A23}$$

Substituting these coefficients into Eq. (A21) yields the velocity-based decomposition:

$$v_{\theta,\phi} \approx v_\theta - \frac{t}{1 - t} \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \tag{A24}$$

We now express the alignment target in the same velocity space. The sample-wise velocity target along the linear trajectory is

$$\boldsymbol{v}_t = \dot{\alpha}_t \boldsymbol{x}_0 + \dot{\sigma}_t \boldsymbol{\epsilon} = \boldsymbol{\epsilon} - \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \boldsymbol{x}_0}{\sigma_t}. \tag{A25}$$

Multiplying Eq. (A19) by $-g^2(t)/2$ and using Eq. (A20), we obtain

$$\begin{aligned} -\frac{1}{2}g^2(t)\widehat{\ell}_t(\boldsymbol{c} \mid \boldsymbol{x}_t) &= \frac{g^2(t)}{2\sigma_t^2}\left(\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0\right) + \boldsymbol{f}(\boldsymbol{x}_t, t) - v_\theta \\ &= \frac{\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0}{\alpha_t \sigma_t} - \frac{\boldsymbol{x}_t}{\alpha_t} - v_\theta \\ &= \frac{\boldsymbol{x}_t - \boldsymbol{x}_0}{\sigma_t} - v_\theta = \boldsymbol{v}_t - v_\theta, \end{aligned} \tag{A26}$$

where the last line uses $1 - \sigma_t = \alpha_t$. Consequently, the velocity residual $\boldsymbol{v}_t - v_\theta$ is the same approximated likelihood-score target scaled by $-t/(1-t)$.

A.2.2 VP-SDE WITH ϵ-PREDICTION.

For a VP forward process $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ constrained by $\alpha_t^2 + \sigma_t^2 = 1$, the score function is parameterized as

$$s_\theta = -\frac{\epsilon_\theta}{\sigma_t}, \qquad s_{\theta,\phi} = -\frac{\epsilon_{\theta,\phi}}{\sigma_t}. \tag{A27}$$

Substituting these relations into Eq. (A18), we obtain

$$-\frac{\epsilon_{\theta,\phi} - \epsilon_\theta}{\sigma_t} \approx \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \tag{A28}$$

This reveals the equivalent noise-based decomposition:

$$\epsilon_{\theta,\phi} \approx \epsilon_\theta - \sigma_t \ell_t(\boldsymbol{c} \mid \boldsymbol{x}_t). \tag{A29}$$

Meanwhile, observing that $\boldsymbol{x}_t - \alpha_t \boldsymbol{x}_0 = \sigma_t \boldsymbol{\epsilon}$, we can substitute Eq. (A27) into the alignment target in Eq. (A19):

$$\begin{aligned} \widehat{\ell}_t(\boldsymbol{c} \mid \boldsymbol{x}_t) &= -\frac{\boldsymbol{\epsilon}}{\sigma_t} + \frac{\epsilon_\theta}{\sigma_t} \\ &= -\frac{\boldsymbol{\epsilon} - \epsilon_\theta}{\sigma_t}. \end{aligned} \tag{A30}$$

Thus, the corresponding alignment target in noise space is simply $\boldsymbol{\epsilon} - \epsilon_\theta$, which is the approximated likelihood score multiplied by $-\sigma_t$.

Equivalent alignment across parameterizations. These derivations show that the side network induces a correction proportional to the likelihood score in each prediction space. Its alignment target can be constructed as the standard training target minus the frozen main-network prediction. To preserve the score-space LISA objective exactly, the same conversion must also be applied to the decoder output and the loss weight. Specifically, let $a_t = -t/(1-t)$ or $-\sigma_t$ for the OT-FM velocity or VP noise parameterization, respectively. Writing the converted decoder output as $\mathcal{D}_\psi^{(a)} = a_t \mathcal{D}_\psi$, we have

$$\mathcal{L}_{\text{LISA}} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{c}, t, \boldsymbol{x}_t}\left[\frac{1}{a_t^2}\left\|\mathcal{D}_\psi^{(a)} - \text{sg}\left[a_t \widehat{\ell}_t(\boldsymbol{c} \mid \boldsymbol{x}_t)\right]\right\|_2^2\right], \tag{A31}$$

where the decoder arguments are omitted for brevity and sg denotes stop-gradient. Omitting the factor $1/a_t^2$ instead gives a time-reweighted alignment objective; the underlying decomposition and target conversion remain the same.

A.3 Additional Results

We provide the visualization results of different diffusion backbones in Figure A6 and more qualitative comparisons with ControlNet in Figure A7.



Figure A6: Visualization examples across SD2.1 and SD3.

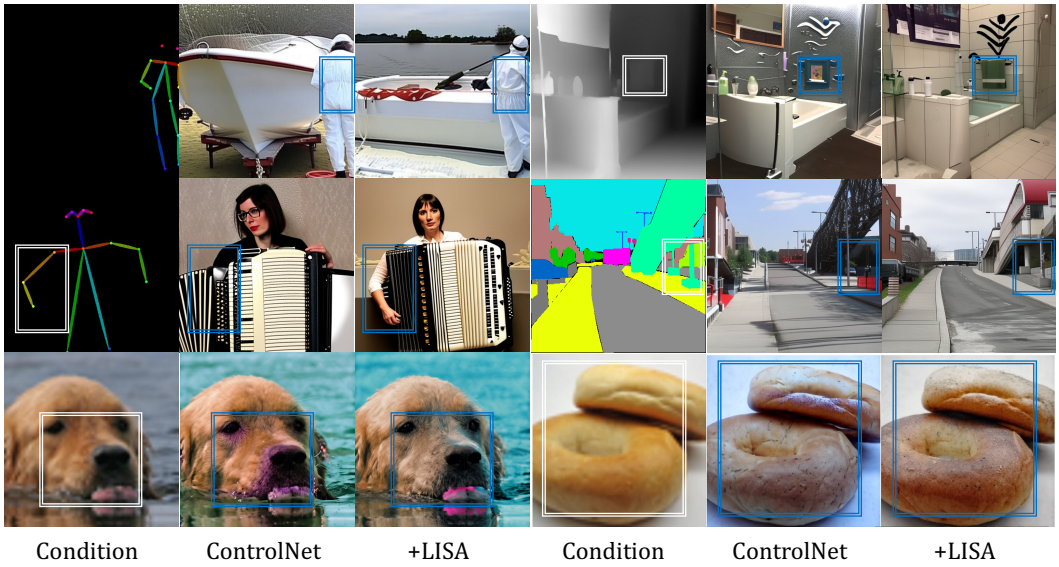


Figure A7: Additional qualitative comparisons with ControlNet across four image-condition generation tasks.