One Algorithm to Align Them All

Boyi Pang[1,*] Savva Ignatyev[2,*] Vladimir Ippolitov[2,*] Ramil Khafizov[2]
Yurii Melnik[2] Oleg Voynov[2,3] Maksim Nakhodnov[4,5,6] Aibek Alanov[4,7]
Xiaopeng Fan[1,8,9,†] Peter Wonka[10,†] Evgeny Burnaev[1,2,3]

[1]Harbin Institute of Technology [2]Applied AI Institute [3]AXXX
[4]FusionBrain Lab, AXXX [5]MSU [6]Constructor University [7]HSE University
[8]Peng Cheng Laboratory [9]HIT Suzhou Research Institute [10]KAUST

[*]Equal contribution [†]Indicates the corresponding author

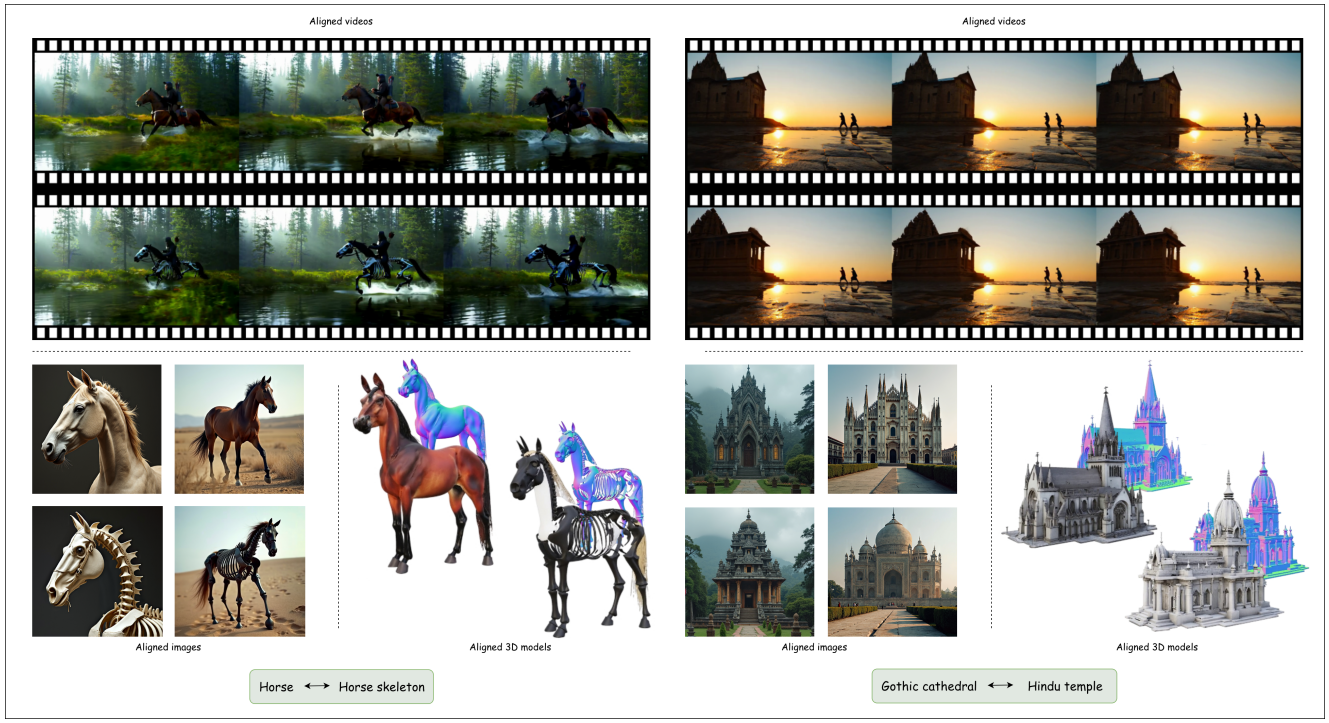


Figure 1. Visualization of generated images, videos, and 3D shapes using our method. The left pair is (horse animal, horse skeleton), the right pair is (gothic temple, Hindu temple).

Abstract

We suggest a new multi-modal algorithm for joint inference of paired structurally aligned samples with Rectified Flow models. While some existing methods propose a codependent generation process, they do not view the problem of joint generation from a structural alignment perspective. Recent work uses Score Distillation Sampling to generate aligned 3D models, but SDS is known to be time-consuming, prone to mode collapse, and often provides cartoonish results. By contrast, our suggested approach relies on the joint transport of a segment in the sample space, yielding faster computation at inference time. Our approach can be built on top of an arbitrary Rectified Flow model operating on the structured latent space. We show the applicability of our method to the domains of image, video, and 3D shape generation using state-of-the-art baselines and evaluate it against both editing-based and joint inference-based competing approaches. We demonstrate a high degree of structural alignment for the sample pairs obtained

with our method and a high visual quality of the samples. Our method improves the state-of-the-art for image and video generation pipelines. For 3D generation, it is able to show comparable quality while working orders of magnitude faster. voyleg.github.io/atata/

1. Introduction

Images, videos, and 3D models are three important domains for generative AI. While users often employ AI to generate one sample at a time, there are many use cases where a user wants to generate a set of examples that are somehow related. Examples are personalization [1], where a user wants to have the same object or person appearing in multiple images or videos, or style consistent generation [13], where multiple samples should be generated in the same style.

In this paper, we address the topic of structurally aligned generation. The goal of structurally aligned generation is to generate multiple samples (images, videos, 3D models) that showcase different main objects, scenes, or environments, but with their semantically (structurally) corresponding parts aligned across spatial/temporal dimensions. This aligned generation comes in handy in generating virtual worlds for training and entertainment, where different objects and scene parts can be easily replaced across domains and "sewn up" into a new scene. It can also help in CAD and CAM applications for creating objects with interchangeable parts. For image and video editing, these tools can be useful for generating artistic effects, for example, transitions called matchcuts [38]. Finally, aligned generation methods can be useful for paired synthetic training data generation for image, video, and 3D, which can be later used to train editing models. While existing methods try to use editing for aligned generation [5, 21, 41], this biases the generation to construct samples that fit one of the descriptions better than the others.

Current methods for aligned generation fall into three categories: 1) Editing-based methods, which use the input sample defined by the first description and force the second sample to adopt its structure. 2) Zero-shot generation using large foundation models. Many large generative models like Nanobanana [12], QWEN [54], or FLUX [4] can generate a grid of semi-consistent images or 3D renders. 3) Native aligned generation. Some works solve the problem of the joint aligned generation, introducing some kind of interaction between the samples during generation. This includes joint generation up to some step in MatchDiffusion [38], attention sharing [13], or embedding into a common latent space in A3D [19].

We propose a novel, highly generalizable, multi-modal method for structurally aligned generation. Inspired by A3D [19], we rethink the necessary properties of transitions between samples with regard to rectified flow field numerical integration and introduce additional constraints to the process, which are necessary for the algorithm's convergence. The method is theoretically applicable to any rectified flow model that operates on the structured latent space, which is demonstrated for images, videos, and 3D objects with minor implementation differences between them. The method requires only changing the inference loop of the flow model, without changing any other components of the pipeline, and does not depend on the specific domain. The main idea of the method is to perform joint inference for pairs of samples by moving them and the linear interpolations between them along the velocity field of the rectified flow model while preserving the interpolation structure. We additionally introduce joint co-guidance during the transport process aimed at ensuring the smoothness of the linear transitions between the samples. Unlike A3D [19], our method is inference-based and does not use SDS, which makes it orders of magnitude faster, allows for reaching state-of-the-art visual quality results without artifacts, and provides a notable variety of samples avoiding mode collapse. Unlike MatchDiffusion [38], our method optimizes for the plausibility of the linear transitions between samples, notably improving structural alignment. It also provides a flexible way to co-guide samples during the inference process.

In summary, we make the following contributions.

- We propose a new algorithm for joint paired inference with rectified flow models, which is based on velocity-guided transport of a segment in latent space, and provide a theoretical justification for it.
- We derive an analytical way to convert the rectified model velocity field into a velocity transport field for joint segment transport.
- We demonstrate that the proposed algorithm, combined with models trained on structured latents, produces highly structurally aligned samples for three modalities: images, videos, and 3D models.
- We obtain results on par with state-of-the-art, specifically trained methods on images and 3D models, and show superior state-of-the-art results on videos.

2. Related work

2.1. Image Generation and Editing

Image generation and editing are the two core problems in generative modeling and form the foundation for controllable video and 3D synthesis. The early approaches relied on Generative Adversarial Networks (GANs) [11, 17, 23] and Variational Autoencoders (VAEs) [25, 52], which were later surpassed by higher-quality diffusion-based methods such as DDPM [14] and Stable Diffusion [44]. The introduction of these models enabled scalable text-to-image generation [42, 46, 48].

More recent progress in text-to-image modeling has been driven by the emergence of Rectified Flow [35] and Diffusion Transformers (DiT) [39], which underpin several state-of-the-art systems, including FLUX.1 [4], Stable Diffusion 3 [9], and Qwen-Image [54]. Their stable training and expressive conditioning have also made them effective when applied to image editing [26, 41, 45].

2.2. Video Generation and Editing

Video-based models are still far behind image-based models due to computational requirements and data scarcity [15, 53]. Existing video editing methods fall into two groups: training-free modifications of video generation models and methods that train specific editing networks.

Training-free approaches are easy to adapt to existing models. A representative example is MatchDiffusion [37], which relies on joint and disjoint diffusion stages, balancing between visual coherence and semantic divergence.

Methods requiring training allow more versatile and fine-grained edits. Among these methods, VACE [20] supports diverse conditions, including inpainting, depth, and motion preservation. Another work, LucyEdit [8], focuses on purely textual edits and special "trigger" words, allowing control of the granularity of edits.

2.3. 3D Generation and Editing

3D object and scene synthesis has also advanced significantly. Unlike 2D domains, where large-scale datasets enable highly generalizable models [4, 54], limited 3D data availability has led researchers to exploit 2D priors from pretrained models—either through multi-view generation with subsequent reconstruction [10, 18, 31, 47, 49] or through score distillation from 2D diffusion models ([34, 36, 40]). More recently, diffusion and flow matching models that operate directly in a 3D latent space have become competitive [27, 57, 58, 60]. These methods fall into two main groups: those using structured voxel-based latents [55, 57, 58, 60] and unstructured latents [27, 28, 63]. While unstructured latents are more computationally efficient, structured ones offer better interpretability—an advantage we build upon in our work. 3D editing has followed a similar trajectory. Early progress relied on the use of 2D priors: either via score distillation [7, 33, 64, 65] or via multi-view diffusion [5, 6, 29]. Recently, methods operating directly in 3D latent space have enabled editing directly in 3D space [32, 57, 61]. However, they still have limited generalization ability due to data scarcity, so 2D-based methods remain a strong baseline for this task.

2.4. Joint Generation of Consistent Objects

Recent work has shown that generative models can jointly produce multiple objects with shared properties. These properties may vary in nature: [30, 62] exploit synchronization mechanisms to produce multiple views of a scene that can subsequently be merged into a panorama, [1] proposed a procedure to generate a set of images with consistent identity. The joint generation of geometrically aligned 3D assets remains more challenging, and the existing solution relies on an iterative, time-consuming SDS-based A3D algorithm [19]. Its core idea is to enforce smooth transitions between generations. MatchDiffusion [38] proposes a training-free joint-generation approach that merges two trajectories before a threshold timestep, but we found this insufficient for reliable structural alignment, motivating our method.

3. Preliminaries

3.1. Flow Matching

Recently, there emerged a tendency in the community to switch from the denoising diffusion models to simpler and more effective Rectified Flow [35] models. Given a distribution X_0 and a sample from it $x_0 \sim X_0$, the initial distribution is transformed into the "noised" version via linear interpolation with $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$, where $t \in [0, 1]$

$$x_t = (1 - t)x_0 + t\epsilon. \tag{1}$$

Then a transformer model is trained to directly regress the velocity field $v_\theta(x_t, t)$ by minimi

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\, x_0 \sim X_0,\, x_1 \sim X_1} \left[\left\| v - v_\theta(x_t, t) \right\|^2 \right]. \tag{2}$$

4. Method

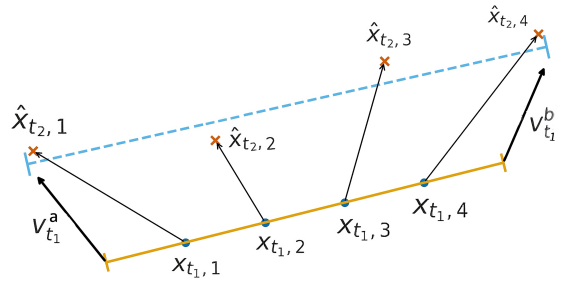
Multiple works [3, 19, 24] imply the importance of two requirements for learning the transitions between samples, which lead to the generation of the semantically aligned objects: 1) Transitions between the aligned objects should provide plausible and realistic samples. 2) Transitions should be smooth (or have a bounded Lipschitz constant). In this section, we analyze these requirements and suggest a principled algorithm for joint inference with rectified flow models, which can be combined with an arbitrary pre-trained model operating on the set of structured latents.

4.1. Joint Inference with Rectified Flow Models

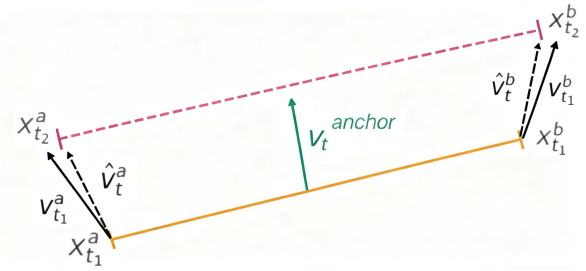
Flow-matching models use time discretization to approximate trajectories along the velocity vector field $v_\Theta(x_t, t, c)$, which is parameterized by a neural network. Given a text embedding c and starting with a sample $x \sim \mathcal{N}(0, \boldsymbol{I})$ taken from a Gaussian noise distribution, the sample x_{t_1} at time step t_1 can be used to calculate x_{t_2} (where $t_1 > t_2$) with the following update rule:

$$x_{t_2} = x_{t_1} + (t_2 - t_1)v_\Theta(x_{t_1}, t_1, c). \tag{3}$$

Let c^a, c^b be two text embeddings and x^a, x^b be the sample variables corresponding to a pair of objects a and b. We initialize x^a and x^b with the same value. Instead of transporting them independently, we want to do it jointly to improve



(a) Joint Step



(b) Smoothness Correction

Figure 2. Method

the plausibility of the transitions between them. To achieve this, we consider transporting a distribution of samples on the line segment $[x^a, x^b] = \{(1-\alpha)\, x^a + \alpha\, x^b \mid \alpha \in [0,1]\}$ defined by some density function $p(\alpha)$. Thus, we shift from transporting individual samples to transporting probability distributions, while preserving the linear structure of these distributions.

To define the update rule for joint transport, we represent the samples distributed on the line segment with the density $p(\alpha)$ by a set of weighted samples evenly distributed across the segment $\{\, x_{t,i} = (1-\alpha_i)\, x^a_t + \alpha_i\, x^b_t \mid i = 1, \dots, k \,\}$ with the respective weighting factors $p(\alpha_i)$. Given the segment $[x^a_{t_1}, x^b_{t_1}]$ we update it to the segment $[x^a_{t_2}, x^b_{t_2}]$ in two steps. First, we update each point $x_{t_1,i}$ to $\hat{x}_{t_2,i}$ individually using the rule in Equation 3 and the interpolated text embedding $c_i = (1-\alpha_i)c^a + \alpha_i c^b$. While the points $\{\, x_{t_1,i} \mid i = 1, \dots, k \,\}$ by definition always lie on a single line in high-dimensional space there is, generally speaking, no such guarantee for the updated points $\{\, \hat{x}_{t_2,i} \mid i = 1, \dots, k \,\}$. That is why we restore the linear structure of the distribution by solving a linear regression problem in Equation 4, minimizing $\mathcal{L}$ w.r.t. $x^a_{t_2}$ and $x^b_{t_2}$.

$$\begin{aligned} \mathcal{L}(x^a_{t_2}, x^b_{t_2}) &= \sum_{i=1}^{k} p(\alpha_i) \|x_{t_2,i} - \hat{x}_{t_2,i}\|_2, \\ \hat{x}_{t_2,i} &= x_{t_1,i} + (t_2 - t_1)\, v_\Theta\big(x_{t_1,i}, t_1, c_i\big), \\ x_{t_2,i} &= (1-\alpha_i)\, x^a_{t_2} + \alpha_i\, x^b_{t_2}. \end{aligned} \tag{4}$$

This regression problem has an explicit solution. We define:

Then the optimal endpoints are:

$$x^a_{t_2} = \frac{c_{11} d_0 - c_{01} d_1}{\Delta}, \quad x^b_{t_2} = \frac{c_{00} d_1 - c_{01} d_0}{\Delta}. \tag{5}$$

The described approach enables a rapid update of the distribution parameters, thereby avoiding the slow gradient-based optimization. Finally, we can define the velocities of the parameters of the probability distribution (boundary points of the segment x^a_t and x^b_t): The scheme of the joint update is shown in Figure 2a.

$$v^a_{t_1} = \frac{x^a_{t_2} - x^a_{t_1}}{t_2 - t_1}, \quad v^b_{t_1} = \frac{x^b_{t_2} - x^b_{t_1}}{t_2 - t_1}. \tag{6}$$

In the presented way, we transform the transport velocity field for samples into the joint transport velocity field for probability distributions supported by the segments in the sample space. In practice, we found it important to use the density $p(\alpha)$ centered around the midpoint during the early iterations and gradually shifted towards a uniform distribution.

4.2. Smoothness regularization

The second aspect is the need for smooth transitions between the two objects. For linear transitions, the "speed" of transition does not depend on the point and is always the same: $\frac{dx_t}{d\alpha} = \frac{d}{d\alpha}((1-\alpha)\, x^a_t + \alpha x^b_t) = x^b_t - x^a_t$. Thus, regularizing the "speed" of transition is the same as regularizing the norm $||x^b_t - x^a_t||_2$ of the segment $[x^a_t, x^b_t]$. Since we initialize x^a and x^b with the same value, the segment norm is zero in the beginning, and over time it diverges, resulting in different, but structurally aligned samples.

The derivative of the L^2 norm of the segment can be written as

$$\frac{d||x^b(t) - x^a(t)||_2}{dt} = \frac{\langle v^b(t) - v^a(t), x^b(t) - x^a(t) \rangle}{||x^b(t) - x^a(t)||_2}, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ is the dot product. We observe that in the cases where we see the severe misalignment between the samples, it usually correlates with the rapid growth of the segment norm during the early iterations of the inference. We can see that this derivative depends on the difference between the velocities of the endpoints of the segment $v^b - v^a$. Thus, minimizing the derivative can be achieved by minimizing this expression or making the velocities of the distribution parameters closer to each other. One particular way to achieve this is to correct the velocities by pulling them closer to some specific *anchor* velocity v^{anchor} (Equation 8)

$$\hat{v}^a_t = w_t v^{anchor}_t + (1-w_t) v^a_t, \quad \hat{v}^b_t = w_t v^{anchor}_t + (1-w_t) v^b_t. \tag{8}$$

The choice of v^{anchor} is important - the anchor velocity vector should point at the "denoising" direction for all the

points on the segment in order not to break the noise schedule of the samples. One possible solution for this task would be to choose $v_t^{anchor} = \frac{v_t^a + v_t^b}{2}$. Though our experiments show that this solution is suboptimal, probably because these two endpoint velocities often have conflicting directions. Our suggested solution is to choose v^{anchor} as a predicted velocity (Equation 9) for the midpoint of the segment

$$v_t^{anchor} = v_\Theta(\frac{x_t^a + x_t^b}{2}, t, \frac{c^a + c^b}{2}). \tag{9}$$

Note that such choice of v^{anchor} is essentially inseparable from the approach presented in Section 4.1 serving as a "correction" to the formulated velocity field, because for the proper inference $\frac{x_t^a + x_t^b}{2}$ should be a plausible sample from the noised distribution at the timestep t which requires taking into account the velocities for the intermediate points of the segment. The scheme for the smoothness correction mechanism is shown in Figure 2b. We observe that choosing the moderate schedule for the values w_t with the emphasis on early iterations does not lead to the degradation of the final samples compared to the "base" rectified flow method.

5. Experiments

We show the universality of our method for joint aligned generation by applying it to three major domains for generative modeling: images, 3D, and video. We select three state-of-the-art rectified flow pre-trained models, which operate on structured latents: voxels, pixels, and video frames. We modify the inference loops of these three models in a self-contained way, which requires only minor differences between the models. We do not change model weights or any other parts of the pipeline.

We use two types of methods as competitors i) joint generation methods, which produce paired output at single inference (A3D [19], MatchDiffusion [38]) ii) editing-based methods which take an independently generated sample from the first prompt *source* and complement the pair by editing it with another prompt from the pair (RF-Inversion [45], VACE [21], and MVEdit [5]).

Metrics: We use multiple metrics to evaluate the degree of structural alignment between the samples and consistency between the samples and the corresponding textual descriptions.

Modality agnostic metrics are based on the evaluation of the sample "projections" to the 2D image space (multi-view images, video frames). **DIFT** alignment **score** was introduced in A3D [19] as a way to measure structural similarity between two salient objects on a pair of images. It works by building a dense grid of 2D points on the source image and finding the corresponding point for each point in the grid using the DIFT [50] method. The pairwise distance between the points is averaged. Finally, the metric is calculated in the reverse direction and averaged one more time. When calculating the DIFT Score, we use the Grounded SAM [43] segmentation model to focus on the foreground object to avoid matching the background. In the recent work SPIE [2] L_1 distance (or equivalently, MAE mean absolute error) between the **depth** maps extracted from the image pair was shown to be an effective proxy metric for structural alignment. We extract depth maps for evaluation using the Depth Anything V2 model [59]. We also employ the widely-used **CLIP Score** to estimate how well each prompt fits the corresponding 2D image by calculating the similarity between the prompt embedding and image embedding. On par with CLIP, we use specific subcriteria from the MLLM[16] method to evaluate image-prompt alignment.

Table 1. 2D Metrics.

	CLIP ↑	DIFT distance ↓ % of object size	Depth L1 ↓ distance	MLLM-based ↑ score	Inference time (s) ↓
Qwen	**24,10**	11,33	37,64	89,60	77
RF_Inversion	23,36	8,62	28,50	83,07	37
Ours	23,23	**6,44**	**26,83**	89,60	**35**

Video metrics are specifically designed to evaluate the qualities of videos. The temporal consistency of the videos is evaluated by calculating the similarity between the **DINO** features of neighboring frames for the edited video. To assess the overall quality of the videos, we use a **VLM** score [22] to evaluate the overall quality, prompt alignment, and preservation of the details of the source video.

3D metrics. We use **GPTEval** [56], a VLM-based method to evaluate the quality of 3D objects, including geometry, texture, and prompt consistency.

5.1. Images

We build our joint image generation pipeline on top of the widely used FLUX.1-dev [4], which is known for the excellent quality of text-to-image generation. For the consistency with 3D experiments, we use the set of object pairs from A3D [19] with short prompts. We compare with two editing-based methods, RF-Inversion [45], which is a flow-inversion training-free method, and instruction-based Qwen-Image-Edit [41, 54]. To obtain *source* editing-free samples, we use FLUX.1-dev model. We rewrite prompts into instructions for Qwen-Image-Edit, asking it to change the content of the source image so that the geometry and background are preserved. For image pair evaluation, we use DIFT Score, Depth Structural Score, CLIP Score, and MLLM Score. The quantitative results are presented in the Table 1. Our method notably improves over RF-Inversion in terms of structural alignment metrics. While Qwen demonstrates a superior understanding of user instructions, it lags far behind in structural alignment score, which makes it unsuitable for this particular problem.

(a) dog, robot (b) gopher, kangaroo (c) spider, octopus (d) amphitheatre, stadium

Figure 3. Visualization of geometry preservation between two generated images. For each example, two images are blended into one with a blending coefficient $\alpha(column)$ that depends on the column index, increasing from 0 (left side) to 1 (right side). With such blending, we show that not only geometry is preserved, but also that smooth transitions between two generations are enabled.

Table 2. GPTEval 3D Metrics, % of comparisons where our method based on Trellis.2 is preferred over competitors and over our method based on Trellis.1.

	Text-asset alignment	3D plausibility	Text-geometry alignment	Texture details	Geometry details	Overall quality
vs. MVEdit	68,17	66,02	77,35	74,77	79,28	77,35
vs. LucidDreamer	64,53	76,69	76,21	69,39	77,77	76,81
vs. A3D	50,97	63,81	59,75	49,86	58,50	61,00
vs. Trellis	67,80	62,68	69,67	72,91	68,59	67,46

Our method can be used to seamlessly combine parts of the samples from parallel domains. We demonstrate it by changing the α blending coefficient across the horizontal axis for the pairs of images in Figure 3.

5.2. 3D Shapes

For joint 3D object generation, we build on two pipelines: Trellis [57], a text-to-3D method, and its improved image-to-3D version, Trellis.2 [58]. The Trellis pipeline consists of two parts: *Structure Generation* operating on dense voxels and *Structured Latents Generation* working with a sparse latent representation. To generate geometrically aligned 3D models, we modify the inference loop for the rectified flow model in the first part of the pipeline, leaving everything else intact.

Because Trellis was trained primarily on detailed text descriptions, the short prompts used in the A3D [19] evaluation introduce a distribution shift. Therefore, we rewrite the short prompts into more detailed descriptions using GPT-5. With Trellis.2, the difference in our approach is that the interpolated input conditions are image tokens rather than text embeddings. To generate aligned 3D objects from aligned images, we use image pairs produced by our image joint generation method, built on top of the Flux model. Trellis.2 and Flux joint generation methods together result in text-to-3D pipeline. We compare our approach with editing-based methods (MVEdit [5], LucidDreamer [34]) and with the joint generation method A3D. The source 3D models for the editing-based methods are obtained using the generative pipelines associated with each method. For evaluation, we use the DIFT score, GPTEval score, and CLIP score. The results are shown in Tables 2 and 3. Our method shows the best CLIP score for text-to-3D semantic alignment. Regarding DIFT structural alignment, our alignment correction combined with the Trellis.1 pipeline yields strong results, albeit slightly lagging behind the exceptionally strong A3D and MVEdit scores because of the limited generalization ability of the backbone model. When incorporated into the Trellis.2 pipeline, our method shows strongest alignment results with decent generation quality. On GPTEval, it confidently improves over the competitors' results. Another important quality of our method is its speed. It provides an order-of-magnitude speedup compared with A3D and is significantly faster than MVEdit. Qualitative results are shown in the Figure 4, demonstrating the high degree of geometric alignment.

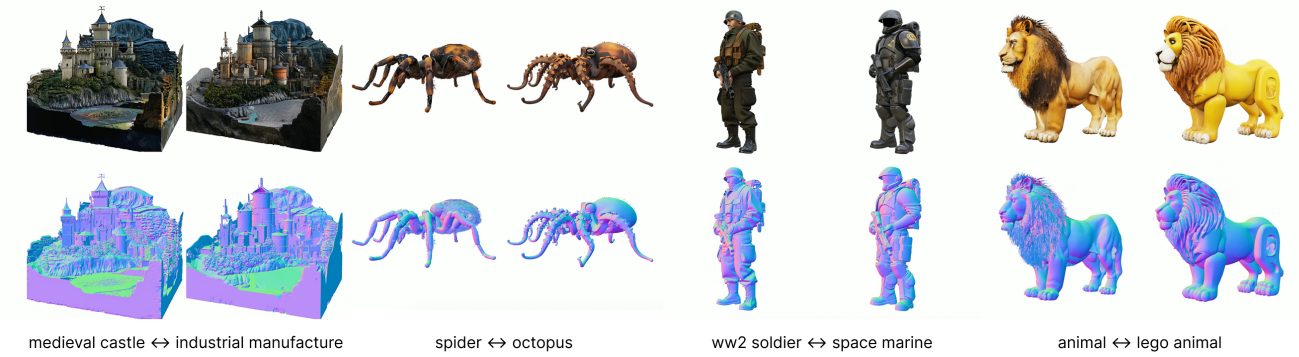
Table 3. Other 3D Metrics.

	CLIP ↑	DIFT distance ↓ % of object size	Time ↓
MVEdit	27,10	5,45	40 min
LucidDreamer	26,40	11,29	2 h
A3D	27,69	5,82	2 h
Ours (Trellis1)	**28,29**	6,98	**1 min**
Ours (Trellis2)	27,51	**4,44**	2 min

5.3. Video

For video experiments, we modify the WAN 2.1 [51] rectified flow model with our method. Due to the dynamic nature of video samples, unlike the static 3D objects and images, we compose a novel set of scenes for evaluation. We aim to cover diverse and complex scenarios, including animals in motion, cities, and human activities. We use two editing-based competitors, LucyEdit [8] and VACE [21]

Figure 4. 3D Alignment Samples



(using depth-conditioned ControlNet). For editing-based models, we use WAN-generated videos as a source video. We also compare our setup with the joint generation method MatchDiffusion [38]. Depth-conditioned VACE tends to preserve the overall layout but substantially alters other aspects, including poses, motion, objects, and background. On the other hand, LucyEdit can produce precise manipulations but is observed to work well only for humans and some animals, making nonsensical edits for the majority of the examples. Quantitative results are shown in Table 4. Our method achieves the highest DINO score, indicating the best self-consistency. While LucyEdit formally shows lower depth MAE than our method, the reason for this is quite trivial - LucyEdit fails to provide necessary edits, leaving inputs intact. This is reflected in its very low VLM score, indicating poor alignment with the text. Despite the depth-conditioned setup, VACE has the worst depth MAE, likely due to distribution shift induced by changing the prompt relative to the first frame. On the other hand, MatchDiffusion shows good results both for VLM and DINO metrics but lags behind in terms of depth alignment, highlighting the need for a more principled approach. Figure 5 visualizes depth absolute error for MatchDiffusion and our method. Both methods show small depth differences in the background, but MatchDiffusion produces much stronger errors around foreground object boundaries, indicating pose misalignment. User-study results are reported in Table 5. The user study favors our method over VACE and LucyEdit across the main criteria. It also supports our hypothesis of superior structural alignment, where our method shows a clear lead over MatchDiffusion.

Table 4. Video Metrics.

	VLM ↑	DINO ↑	Depth MAE ↓
MatchDiffusion	<u>7,64</u>	0,96	1,12
Lucy-edit	4,77	0,97	**0,84**
VACE	6,26	<u>0,98</u>	2,60
Ours	**7,66**	**0,98**	<u>0,89</u>

Table 5. User study.

User study, % of comparisons where our method is preferred

	Structural alignment	Video quality	Text-to-video consistency
vs. MatchDiffusion	67,50	58,60	47,00
vs. Lucy-edit	69,30	77,70	76,40
vs. VACE	88,00	60,90	58,60

6. Ablation

To validate the design choices of our algorithm, we progressively remove components from our algorithm. After a series of such simplifications, our method effectively reduces to MatchDiffusion [38], yielding a sequence of controlled comparisons that isolates the contribution of each component. We start from the setup (A), which is our full pipeline described in Section 4. Setup (B) is obtained by switching $v_t^{anchor} = \frac{v_t^a + v_t^b}{2}$ from $v_\Theta(\frac{x_t^a + x_t^b}{2}, t)$. Setup (C) further removes sampling of intermediate points $x_{t,i}$. Without intermediate points, the method no longer enforces plausible transitions between samples, which is central to our approach. When we move to setup (D), the only restriction synchronizing the movement of x_t^a and x_t^b left in place is the anchor velocity $v_t^{anchor} = \frac{v_t^a + v_t^b}{2}$. Instead of using the smooth schedule for weight coefficients, we simplify it to a hard cutoff which makes the algorithm effectively equivalent to the MatchDiffusion [38] baseline. This step deprives the algorithm of a flexible co-guidance mechanism for the early stages of inference and removes any form of synchronization for the late stages. We evaluate setups (B) and (C) with the video pipeline reporting depth MAE structural

(a) Our method's source video frame

(b) Our method's target video frame

(c) Depth difference between our method's source and target video frames

(d) MatchDiffusion's source video frame

(e) MatchDiffusion's target video frame

(f) Depth difference between MatchDiffusion's source and target video frames

Figure 5. Visualization of geometry preservation between source and target video frames. Larger white areas mean higher difference between corresponding pixels.

score in Table 6a. Since the setup (D) is roughly equivalent to MatchDiffusion, for which we have already reported results in Section 5, we decide to evaluate this setup with the image modality; the results are shown in Table 6b. Visual examples for all the setups with the image pipeline are demonstrated in Figure 6. Quantitative and qualitative comparisons demonstrate a gradual degradation of the results during the component removal.

	Depth MAE ↓
Setup A (ours)	**0,89**
Setup B	1,33
Setup C	1,50

(a) Depth MAE ↓

	DIFT distance ↓ % of object size
Setup A (ours)	**6.44**
Setup D	8,42

(b) DIFT distance ↓ (% of object size)

Table 6. Ablation study results. Lower is better (↓).

(a) Setup A (ours) (b) Setup B (c) Setup C (d) Setup D

Figure 6. Visualization of the impact of different components of our method. We can see that each component contributes to the results and that the results degrade as we gradually remove components from our algorithm.

7. Conclusions

We present a new universal method for joint inference with rectified flow models, which enables the rapid production of structurally aligned samples across different modalities. Our method requires only a compact and local modification of the inference loop and can be applied on top of any pre-trained rectified flow model working with structured latent representations. We demonstrate the performance of our method across three modalities—video, 3D, and images — comparing it both with editing-based methods and joint training methods. Across all modalities, our method either achieves performance on par with the state of the art or surpasses it. When applied to video modality, our method shows superior performance, enabling the accurate alignment of complex and vibrant environments. Our method has multiple potential applications, including the creation of aligned virtual environments and assets, and the generation of synthetic data. In future work, the method can be further applied to other modalities, such as 4D video and keypoint movements.

Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement №139-10-2025-033.

References

[1] Omri Avrahami, Amir Hertz, Yael Vinker, Moab Arar, Shlomi Fruchter, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. The chosen one: Consistent characters in text-to-image diffusion models. *arXiv preprint arXiv:2311.10093*, 2023. 2, 3

[2] Elior Benarous, Yilun Du, and Heng Yang. Spie: Semantic and structural post-training of image editing diffusion models with ai feedback. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6395–6407, 2025. 5

[3] David Berthelot*, Colin Raffel*, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019. 3

[4] Black Forest Labs. FLUX.1 [dev]. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024. Open-weight rectified-flow text-to-image model. 2, 3, 5

[5] Hansheng Chen, Ruoxi Shi, Yulin Liu, Bokui Shen, Jiayuan Gu, Gordon Wetzstein, Hao Su, and Leonidas J. Guibas. Generic 3d diffusion adapter using controlled multi-view editing. *arXiv preprint arXiv:2403.12032*, 2024. 2, 3, 5, 6

[6] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. In *European Conference on Computer Vision*, pages 74–92. Springer, 2024. 3

[7] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 3

[8] DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025. Technical report. 3, 6

[9] 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. 3

[10] Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21524–21536, 2025. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[12] Google. Image editing in gemini just got a major upgrade, 2025. Describes *Nano Banana* (Gemini 2.5 Flash Image). Accessed: 2025-11-13. 2

[13] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4775–4785, 2024. 2

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

[15] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022. 3

[16] Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–17, 2025. 5

[17] Nick Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The gan is dead; long live the gan! a modern gan baseline. *Advances in Neural Information Processing Systems*, 37:44177–44215, 2024. 2

[18] Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16377–16387, 2025. 3

[19] Savva Victorovich Ignatyev, Nina Konovalova, Daniil Selikhanovych, Oleg Voynov, Nikolay Patakin, Ilya Olkov, Dmitry Senushkin, Alexey Artemov, Anton Konushin, Alexander Filippov, Peter Wonka, and Evgeny Burnaev. A3d: Does diffusion dream about 3D alignment? In *International Conference on Learning Representations (ICLR)*, 2025. Poster. 2, 3, 5, 6

[20] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing, 2025. 3

[21] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17191–17202, 2025. 2, 5, 6

[22] Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, Daniil Pakhomov, Zhe Lin, Soo Ye Kim, and Qiang Xu. Editverse: Unifying image and video editing and generation with in-context learning, 2025. 5

[23] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10124–10134, 2023. 2

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. 3

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[26] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 3

[27] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 3

[28] Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Fuyun Wang, Huiwen Shi, Xianghui Yang, Qingxiang Lin, Jingwei Huang, Yuhong Liu, et al. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*, 2025. 3

[29] Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Sangheon Shin, Sangmin Kim, and Sangpil Kim. Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11135–11145, 2025. 3

[30] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36:50648–50660, 2023. 3

[31] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 3

[32] Lin Li, Zehuan Huang, Haoran Feng, Gengxiong Zhuang, Rui Chen, Chunchao Guo, and Lu Sheng. Voxhammer: Training-free precise and coherent 3d editing in native 3d space. *arXiv preprint arXiv:2508.19247*, 2025. 3

[33] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3279–3287, 2024. 3

[34] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. *arXiv preprint arXiv:2311.11284*, 2023. 3, 6

[35] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[36] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin M Solomon. Score distillation via reparametrized ddim. *Advances in Neural Information Processing Systems*, 37:26011–26044, 2024. 3

[37] Alejandro Pardo, Fabio Pizzati, Tong Zhang, Alexander Pondaven, Philip Torr, Juan Camilo Perez, and Bernard Ghanem. Matchdiffusion: Training-free generation of match-cuts, 2024. 3

[38] Alejandro Pardo, Fabio Pizzati, Tong Zhang, Alexander Pondaven, Philip Torr, Juan Camilo Perez, and Bernard Ghanem. Matchdiffusion: Training-free generation of match-cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 3, 5, 7

[39] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 3

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

[41] Qwen Team. Qwen-image-edit. https://huggingface.co/Qwen/Qwen-Image-Edit, 2025. Image editing foundation model based on Qwen-Image. 2, 3, 5

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[43] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 5

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

[45] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *Proceedings of the Thirteenth International Conference on Learning Representations*, 2025. 3, 5

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

[47] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[49] Stanislaw Szymanowicz, Jason Y Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T Barron, and Philipp Henzler. Bolt3d: Generating 3d scenes in seconds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24846–24857, 2025. 3

[50] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. 5

[51] 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. 6

[52] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[53] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 3

[54] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Wei Hu, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. 2, 3, 5

[55] Guanjun Wu, Jiemin Fang, Chen Yang, Sikuang Li, Taoran Yi, Jia Lu, Zanwei Zhou, Jiazhong Cen, Lingxi Xie, Xiaopeng Zhang, et al. Unilat3d: Geometry-appearance unified latents for single-stage 3d generation. *arXiv preprint arXiv:2509.25079*, 2025. 3

[56] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas J. Guibas, Dahua Lin, and Gordon Wetzstein. GPT-4V(ision) is a Human-Aligned Evaluator for Text-to-3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5

[57] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 3, 6

[58] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, et al. Native and compact structured latents for 3d generation. *arXiv preprint arXiv:2512.14692*, 2025. 3, 6

[59] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 5

[60] Yuanbo Yang, Jiahao Shao, Xinyang Li, Yujun Shen, Andreas Geiger, and Yiyi Liao. Prometheus: 3d-aware latent diffusion models for feed-forward text-to-3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2857–2869, 2025. 3

[61] Junliang Ye, Shenghao Xie, Ruowen Zhao, Zhengyi Wang, Hongyu Yan, Wenqiang Zu, Lei Ma, and Jun Zhu. Nano3d: A training-free approach for efficient 3d editing without masks. *arXiv preprint arXiv:2510.15019*, 2025. 3

[62] Kyeongmin Yeo, Jaihoon Kim, and Minhyuk Sung. Stochsync: Stochastic diffusion synchronization for image generation in arbitrary spaces. *arXiv preprint arXiv:2501.15445*, 2025. 3

[63] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 3

[64] Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 3

[65] Jingyu Zhuang, Di Kang, Yan-Pei Cao, Guanbin Li, Liang Lin, and Ying Shan. Tip-editor: An accurate 3d editor following both text-prompts and image-prompts. *ACM Transactions on Graphics (TOG)*, 43(4):1–12, 2024. 3