HD-Fusion: Detailed Text-to-3D Generation Leveraging Multiple Noise Estimation

Jinbo Wu Xiaobo Gao Xing Liu Zhengyang Shen Chen Zhao
Haocheng Feng Jingtuo Liu Errui Ding
Department of Computer Vision Technology (VIS), Baidu Inc., China
{wujinbo01,gaoxiaobo,liuxing12,shenzhengyang01,zhaochen03,
fenghaocheng,liujingtuo,dingerrui}@baidu.com

Abstract

In this paper, we study Text-to-3D content generation leveraging 2D diffusion priors to enhance the quality and detail of the generated 3D models. Recent progress [11] in text-to-3D has shown that employing high-resolution (e.g., 512×512) renderings can lead to the production of high-quality 3D models using latent diffusion priors. To enable rendering at even higher resolutions, which has the potential to further augment the quality and detail of the models, we propose a novel approach that combines multiple noise estimation processes with a pretrained 2D diffusion prior. Distinct from the Bar-Tal et al.s' study which binds multiple denoised results [1] to generate images from texts, our approach integrates the computation of scoring distillation losses such as SDS loss and VSD loss which are essential techniques for the 3D content generation with 2D diffusion priors. We experimentally evaluated the proposed approach. The results show that the proposed approach can generate high-quality details compared to the baselines.

1. Introduction

Generating 3D content from text (Text-to-3D) is a crucial technique to support the burgeoning popularity of the Metaverse. This technology allows people to create various items by expressing their ideas through text and interacting with their creations using AR/VR headsets. Such a technique elevates the user experience to a new level of immersion. Ideally, 3D content generated through an optimal algorithm should not only exhibit high quality in terms of geometry and appearance but also maintain a diverse range within constraints. Considering the substantial success that 2D diffusion models have achieved in text-to-image synthesis tasks, it is intuitive to conceive an approach for addressing 3D content generation that involves employing a 3D diffusion model, which would be trained and utilized in a manner similar to 2D diffusion models. However, its potential is limited by the fact that 3D diffusion-based approaches necessitate substantial computational resources and an extensive collection of 3D models paired with text for training to achieve high-quality and diverse generations. To address these difficulties, techniques for generating 3D content using 2D diffusion priors [11, 13, 19] have gained popularity and experienced rapid development in recent months. The advancement of this line of approaches is grounded in DreamFusion [19], which proposed a score distillation loss, termed SDS loss. Given the images rendered from a neural radiance field θ representing a 3D object to be optimized, the SDS loss approximates the gradient directions with respect to θ for each image I. By moving θ along with the directions, the 3D object is optimized in such a way that, when rendered from a random angle, it looks as though it has been generated by the 2D diffusion model. Although DreamFusion has significantly advanced development, it falls short in generating high-quality 3D models. Lin et al. [11] addressed this issue, positing that the quality of 3D models can be enhanced by rendering high-resolution images to compute SDS loss. To increase training efficiency, they introduced the application of DMTet [23] to represent a model's shape, and proposed employing a latent diffusion model for supervision. Building on Lin et al.'s observation that high-resolution renderings contribute to improved quality in the generated 3D models, we propose an approach that allows for memory-efficient training when rendering images at even higher resolutions. In each training iteration utilizing our approach, we abstain from estimating noise for the entire rendered image, as this leads to memory issues when the image is rendered at a high resolution (e.g., 1024×1024). Instead, we crop the latent noisy image into overlapping tiles and conduct independent noise estimation processes for these tiles. Then, we integrate the estimated *tiled noises* into a whole with pixel-wise weights for the computation of the SDS loss. Given that the SDS

Figure 1. The proposed multiple noise estimation approach yields high-quality 3D models with enhanced details. The prompts for the three contents are 1) "A model of a house in Tudor style.", 2) "A delicious croissant." and 3) "a ripe strawberry."

loss can be computed without the need for a computational graph of the weights in the diffusion-UNet model, we can calculate the loss without the necessity for additional GPU memory. Along with our main contribution, we present our entire Text-to-3D pipeline in this paper. We adopt a two-stage framework similar to Magic3D [11]'s, and incorporated ControlNet to mitigate the Janus problem and surface inconsistency. Our experiments demonstrated that the proposed approach improved the performance of 3D content generation. Furthermore, we highlight that the proposed method can be compatible with recent advancements such as VSD [30] in Text-to-3D approaches, indicating that a collaboration between these innovations and our method holds the potential for achieving more significant improvements in visual quality. Our contribution is summarized as follows:

- We propose an approach that combines multiple noise estimation processes, enabling memory-efficient training for 3D generation within a high-resolution rendering space.
- We present an entire Text-to-3D system that leverages the proposed approach and ControlNet for geometrically correct 3D content generation through 2D diffusion priors. We provide a detailed explanation of the system.

2. Related Work

2D diffusion models. Generative image models have rapidly developed [4, 16, 17, 20, 24, 26, 31]. Ho et al. [6] proposed a diffusion-based model for 2D image synthesis, which showed remarkable success in achieving high fidelity and diversity. Rombach et al. [21] extended this technique by proposing its application in the latent space to enhance efficiency. Recently, methods for effectively fine-tuning diffusion models have garnered attention. Zhang et al. [32] proposed a neural network structure named ControlNet, which can be trained end-to-end with limited computational resources. The ControlNet enables a large diffusion model to be controlled by task-specific conditions (e.g., pose, normal, etc.) in addition to text. Hu et al. [7] proposed a low-rank adaptation method (LoRA) which is primarily used for fine-tuning large language models. This method was subsequently adapted for the 2D diffusion models, enabling them to be effectively fine-tuned to generate images with specific content or styles. Bar-Tal et al. [1] proposed a multi-diffusion pipeline for a controllable generation. Their approach binds together multiple diffusion generation processes with a shared set of parameters or constraints.

3D representation. Effective 3D representations have been a subject of study for many years [3, 8, 12, 18, 22, 25, 27, 28]. Recently, neural field-based approaches have demonstrated significant potential. Mildenhall et al. [14] introduced NeRF, which implicitly represents a 3D object or a scene and generates novel views through a volume rendering algorithm given specific camera poses. In subsequent research, Muller et al. [15] proposed an advanced multi-resolution hash-encoding approach to encode the position of a 3D point. As this approach substantially accelerates the convergence during the training of a neural field, it has been widely adopted in later studies for positional encoding, including ours. Wang et al. [29] proposed an approach that

represents the surface of an object using an implicit signed distance function, which is optimized using posed sparse images of the object. On the other hand, Shen et al. [23] proposed a deformable tetrahedral grid model to represent the geometry of a 3D object explicitly. This model can be directly optimized for accurate shape reconstruction and is made possible for mesh generation in a differentiable manner.

Text-to-3D with 2D diffusion priors. Poole et al. [19] proposed a loss function (SDS loss) that enables the training of 3D object generation using 2D diffusion models. Lin et al. [11] proposed a two-stage approach, based on the SDS loss, for generating 3D objects in a coarse-to-fine manner. They demonstrated that their approach could produce higher-quality 3D results with lower computational costs. Metzer et al. [13] introduced how they applied the computation of SDS loss in the latent space. Furthermore, they proposed the utilization of coarse shapes (e.g., a mesh with the coarse structure of a desired object) to guide the 3D generation process. Chen et al. [2] proposed an approach that disentangles geometry and appearance modeling. Through empirical evaluation, we found that this approach can recover finer geometries of a generated 3D object. Wang et al. [30] introduced variational score distillation that serves as an enhanced version of the SDS loss, addressing the issues of over-saturation, over-smoothing, and low diversity that sometimes occur in 3D results generated with SDS loss.

3. Approach

In this section, we introduce the proposed approach. First, we explain how we conducted the multiple noise estimation process for computing SDS loss, which enables the supervision of 3D generation. Then, we explain our two-stage Text-to-3D generation system.

3.1. Multiple Noise Estimation for Memory-Efficient Optimization of 3D Models

We illustrated the proposed multiple noise estimation approach in Fig. 2. Given the latent representation $J \in \mathbb{R}^{H\times W\times C}$ of a rendered image I, a noise $\varepsilon \sim \mathcal{N}(0,1)$ sampled at step t is applied to J to calculate the noisy latent J_t (see equ. 4 in [6]). Then, J_t is partitioned into overlapping tiles using a square sliding window with unchanged window size, producing a set of tiled noisy latents. We employ a pretrained stable diffusion model ϕ (referred to as Controlled SD-UNet in Fig. 2) as our noise estimator. The lock icon means that the pretrained weights remain static during training. An instance of ControlNet is optionally employed to inject task-specific guidance into the noise estimator. The noise estimator produces the *estimated noise* for each tile. We align all the estimated noises to their corresponding locations on J_t and consolidate them into a single noise tensor with pixel-wise weights. Inspired by [1], we set the weights to calculate pixel-wise averages within the overlapping regions.

Formulation We formulate the proposed approach based on Lin et al. [11]'s extension of the SDS loss. We consider $Q_\varphi(\cdot\,; y, t)$ as a noise estimator where φ, y, and t denote its parameters, a condition, and a diffusion time step respectively, and consider $\mathcal{T}(\cdot)$ as an image processing function. In the proposed approach, we employ $\mathcal{T}(\cdot; s, k)$ as a sliding window function that outputs M cropped tiles from its input with stride s and window size k. We define $\mathcal{T}_m(\cdot; s, k)$ to obtain the m^{th} cropped tile, which is centered at (u_m, v_m) with both its height and width equal to k. The notion (u_m, v_m) represents the 2D coordinates. We simplify the notion $\mathcal{T}_m(\cdot; s, k)$ to $\mathcal{T}_m(\cdot)$ for clarity. Consequently, we formulate the single noise estimation process on the m^{th} tile as follows:

$$\mathcal{F}_\phi(J_t, m) \triangleq Q_\phi \circ \mathcal{T}_m(J_t; y, t). \tag{1}$$

We omit the notions t and y in $\mathcal{F}_\phi(\cdot)$ for the sake of clarity. To formulate the proposed *multiple noise estimation*, we further define two variables which are i) $\xi \in \mathbb{R}^{H\times W\times C}$ and ii) $\mathcal{W} \in \mathbb{R}^{H\times W\times C}$, both of which are initialized by being filled with zeros. We employ $\xi_{(u_m,v_m,k)}$ and $\mathcal{W}_{(u_m,v_m,k)}$ to represent the regions centered at (u_m, v_m) and enclosed by a square with its side length equal to k, for the two variables. Additionally, we define two assignment operators, $\mathcal{A}_\xi(\cdot)$ and $\mathcal{A}_\mathcal{W}(\cdot)$, formally expressed as:

$$\mathcal{A}_\xi(m):\ \xi_{(u_m,v_m,k)} = \mathcal{F}_\phi(J_t, m) + \mathcal{T}_m(\xi), \tag{2}$$

$$\mathcal{A}_\mathcal{W}(m):\ \mathcal{W}_{(u_m,v_m,k)} = \sum_{m=1}^{M} \mathbb{1} + \mathcal{T}_m(\mathcal{W}). \tag{3}$$

The two operators replace the values within $\xi_{(u_m,v_m,k)}$ and $\mathcal{W}_{(u_m,v_m,k)}$ with the results computed on the right-hand side of the equalities, as per equations 2 and 3. We define the proposed multiple noise estimation function $\mathcal{M}(\cdot)$ in

Algorithm 1 The definition of the proposed multiple noise estimation function $\mathcal{M}(\cdot)$. Note, ϕ, y, t, s, k are omitted for clarity.

function $\mathcal{M}(J_t, \xi, \mathcal{W})$
  $\xi \leftarrow 0$ ▷ Initialize ξ to 0
  $\mathcal{W} \leftarrow 0$ ▷ Initialize $\mathcal{W}$ to 0
  for $m = 1, \dots M$ **do**
    Run $\mathcal{A}_\xi(m)$ and $\mathcal{A}_\mathcal{W}(m)$ ▷ See equ. 2 and 3
  end for
  $\hat{\varepsilon} \leftarrow \frac{\xi}{\mathcal{W}}$
  return $\hat{\varepsilon}$
end function

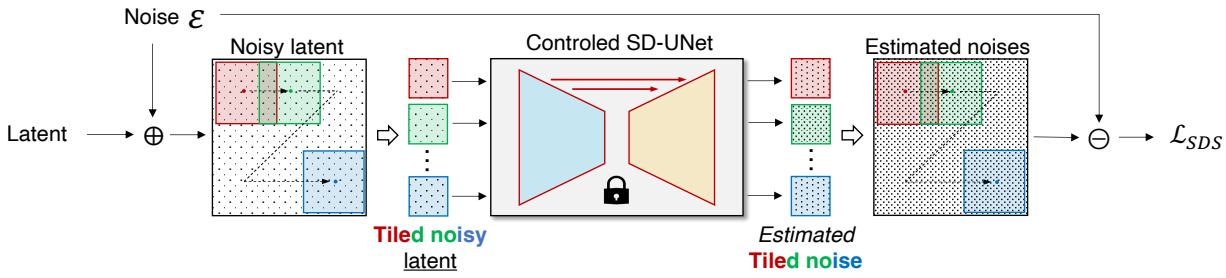

Figure 2. The proposed multiple noise estimation is illustrated here. The "Latent" represents the latent representation of a rendered image. The "Noise ε" means the additive noise sampled at step t from the diffusion process [6]. The "Tiled noisy latent" is obtained by cropping overlapping patches from the "Noisy latent" with a sliding window. The "Controlled SD-UNet" means stable diffusion model optionally powered by an instance of ControlNet. The "Estimated noises" is produced by consolidating all the estimated tiles of noise.

Alg. 1, and re-formulate the SDS loss from Lin et al. [11]'s version as follows:

$$\nabla_\theta \mathcal{L}_{SDS}(\phi, g(\theta)) \equiv \mathbb{E}_{t,\varepsilon}\left[\omega(t)(\mathcal{M}(J_t, \xi, \mathcal{W}) - \varepsilon)\frac{\partial J_t}{\partial I}\frac{\partial I}{\partial \theta}\right] \tag{4}$$

where I denotes an image rendered using $g(\cdot)$, $g(\cdot)$ represents a rendering process. t denotes a diffusion time step. $w(t)$ is a function that outputs a weight for t. J and J_t denote the latent image and the noisy latent image calculated at step t. y denotes the condition(s). ε denotes a noise tensor sampled according to $\mathcal{N}(0, 1)$. Through experimental validation, we demonstrate that the proposed approach effectively harnesses the strengths of high resolution in enhancing local quality and detail.

3.2. Two-stage Text-to-3D System

We outline the proposed two-stage Text-to-3D generation system in this section, detailed explanations for each stage are provided in subsequent sections (Sec. 3.3 and Sec. 3.4).

Outline The proposed Text-to-3D system operates in two stages, aiming to generate 3D models in a coarse-to-fine manner. In the first stage, we employ a neural field to represent the shape and color of a 3D object. To encode the images generated by the neural field to latent space, we adopt the encoder of a pretrained VQ-GAN [5], and compute SDS loss to optimize the neural field in latent space. In the second stage, we adopt a DMTet model and a color network to represent the 3D model. We adopt a differentiable pipeline [9] to render the views of the model. Similar to the first stage, we encode the views using the same encoder and compute SDS loss in the latent space. In both stages, an identical diffusion model is employed for a task. We select the most appropriate diffusion model from its various versions based on performance metrics in the corresponding image generation task. To guarantee the generation of geometrically accurate views through the diffusion model, we optionally inject task-dependent guidance via ControlNet. For instance, in the case of generating 3D figures, we opt for the "deliberate" version of the 2D diffusion model and employ pose guidance.

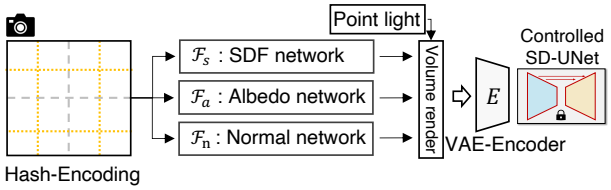

Figure 3. The coarse stage of the proposed approach.

3.3. Stage1: Coarse Level Generation

In this stage, our objective is to generate a coarse representation of an object's shape and color by a neural field. We illustrate our approach in Fig. 3. We employ a signed distance function (SDF) $\mathcal{F}_s$ in conjunction with an albedo prediction network $\mathcal{F}_a$ and a normal prediction network $\mathcal{F}_n$. These modules are implemented using Multi-Layer Perceptrons (MLPs). For computational efficiency, we use one layer for each module. To accelerate convergence, we incorporate a Hash Grid [15] for positional encoding. We set the resolution of the output images from the volume renderer to be 256×256, which we posit is an appropriate configuration for learning the coarse representation.

Initialization The modules $\mathcal{F}_s$, $\mathcal{F}_a$, and $\mathcal{F}_n$ are randomly initialized. The VAE-Encoder is initialized using the pretrained weights provided by [21]. To facilitate content generation concentrated around the center of the 3D coordinate space, we initialize a "density blob" at the space center and utilize it as a spatial bias during training. It is noteworthy that our density blob is modeled on SDF values. This diverges from the modelings used in the previous studies [11, 19], which calculated their density blobs using the

densities predicted in the neural fields. We introduce how we utilize the density blob during training in the following paragraph.

Training We employ a spherical coordinate system for camera placement, which is conducive to center-focused imaging. We place cameras at random locations within a limited dome and generate shaded colors for each camera to render an image. Specifically, assuming a point light source characterized by its 3D coordinate l, color l_ρ, and ambient light color l_α, we sample 3D points along the emitting rays and calculate the color c for each sampled point using diffuse reflectance [10]:

$$c = \rho \circ (l_\rho \circ max(0, n \cdot (l - \mu)/\left\|l - \mu\right\|) + l_\alpha) \quad (5)$$

The albedo, denoted as ρ, and the normal, denoted as n, are predicted by the normal prediction network and the albedo prediction network, respectively. We employ a "textureless" rendering approach (as used in [19]) and incorporate it at a specific ratio to enhance the robustness of the training process. We calculate the aforementioned spatial bias using

$$\tau_{init}(\mu) = \lambda_\tau \cdot \left(||\mu - r||^2\right), \quad (6)$$

where μ is the norm of a point sampled along a ray, r denotes the radius of the "density blob", and λ_τ denotes the scale parameter. $\tau_{init}(\mu)$ is the calculated bias added to the predicted SDF value of the sampled point. We set $\lambda_\tau = 1.0$ and $r = 0$ in our experiment. We omit the introduction to volume rendering techniques as they are already detailed in previous studies such as [29]. We do not use the proposed multiple noise estimation processes during coarse-level training, as we do not expect high-quality details to be learned at this stage.

3.4. Stage2: Fine Level Generation

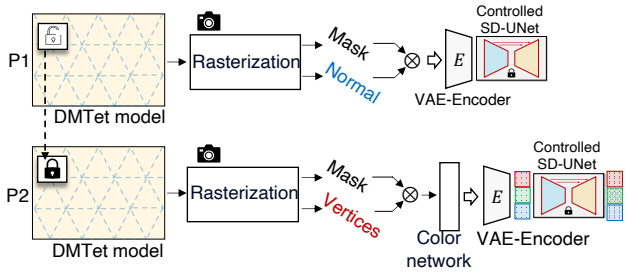

Figure 4. This figure illustrates the fine-level generation stage of the proposed approach. It has two phases, denoted by P1 and P2. We learn geometry and color in P1 and P2 separately. The DMTet model is optimized in P1 and is fixed in P2. The proposed multiple noise estimation is only applied in P2.

We present how we generate high-quality 3D models based on their coarse shapes and colors. The pipeline of this stage is illustrated in Fig. 4. We utilize a Deformable Marching Tetrahedral Grid (DMTet) model [23] to represent an object's geometry due to its efficiency. To represent the colors, we employ a color network that is implemented in the same manner as the albedo network in Stage 1. We employ a rasterization model to output the normal and the indices corresponding to the set of vertices in the DMTet model. The VAE-Encoder takes the rendered image as input and outputs its latent representation. We then compute the SDS loss within the latent space using the diffusion model (depicted as Controlled SD-UNet in Fig. 4). It should be noted that we learn geometry (depicted as P1 in Fig. 4) and colors (depicted as P2 in Fig. 4) separately. The proposed multiple noise estimation is employed only in P2. We explain the details in the following paragraph.

3.4.1 Initialization

DMTet model To initialize the DMTet model, we set the SDF values of the vertices by querying the neural SDF field, which has been trained in Stage 1. We assign zeros to the offsets for all the vertices. This process allows the DMTet model roughly captures the shape of the object.

Color network We initialize the color network by employing the weights obtained from the color network that was trained in Stage 1.

3.4.2 Training

We first train our model to learn geometry (denoted by "P1" in Fig. 4) until it converges, then we freeze the geometry and train it to learn colors (denoted by "P2" in Fig. 4). We employ the spherical coordinate system to position our cameras, enabling them to focus on the center.

Phase1: Learning geometry Inspired by a recent advance [2], We optimize the DMTet model using normal images rendered from various viewpoints. The rasterization module [9] calculates the normal image and the object mask from the DMTet model given a specific camera pose. Subsequently, the mask is employed to filter out the background pixels in the normal image. The filtered normal image is then fed into the VAE-Encoder E for the computation of the SDS loss in the latent space. We set the resolution as 512×512 for rendering the normal image.

Phase2: Learning Color Once the geometry learning is completed in phase 1, we freeze the DMTet model and exclusively learn colors using the pipeline depicted as P2 in Fig. 4. To render an image, we first collect a set of corresponding vertices by utilizing the rasterization module and the object mask. Then, we query the color network using the 3D positions of these vertices. Since the rasterization process has already established the projections from the 3D

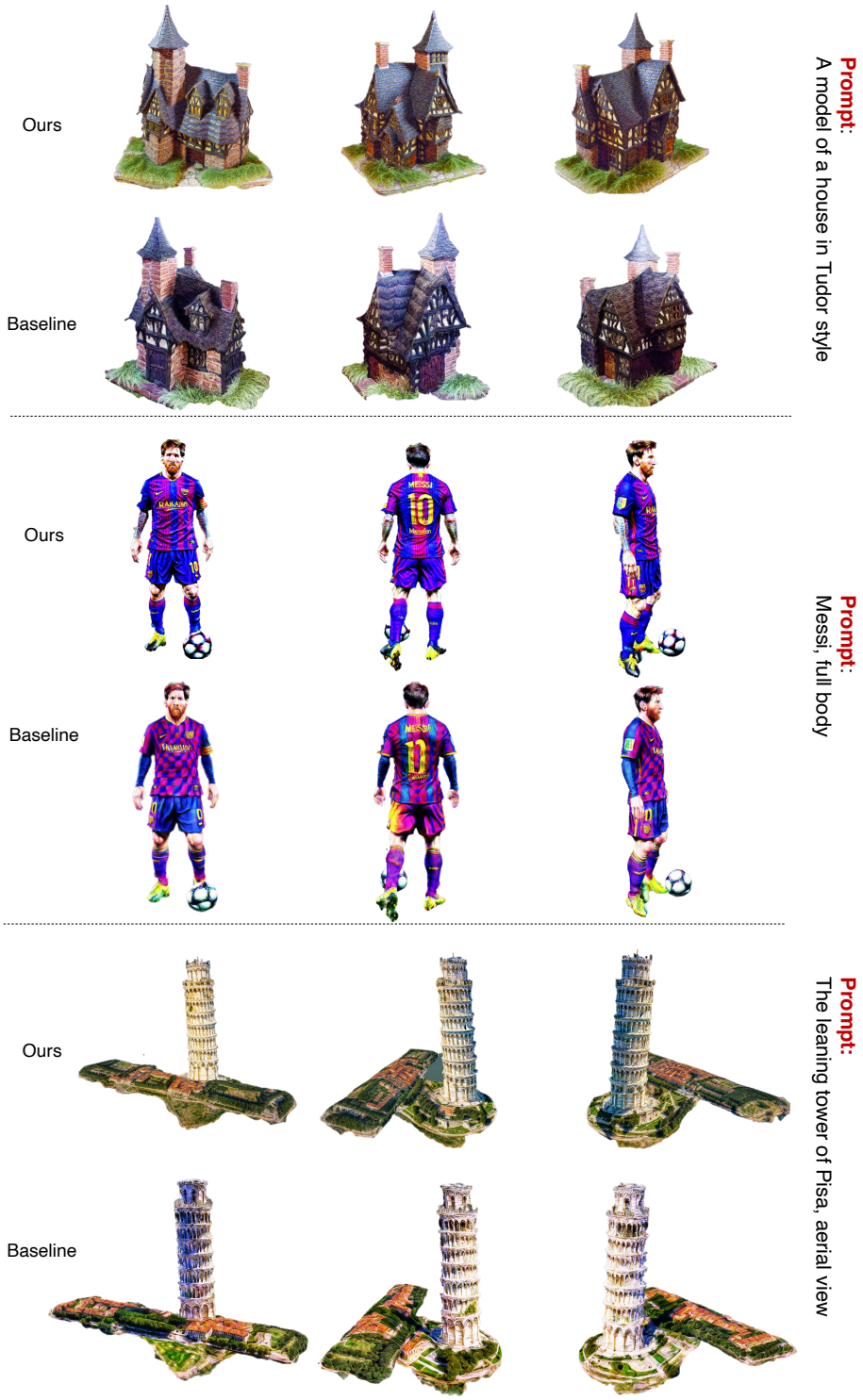

Figure 5. Result comparison. "Ours" means the results generated **with** the proposed multiple noise estimation. "Baseline" means the results generated without applying the multiple noise estimation.

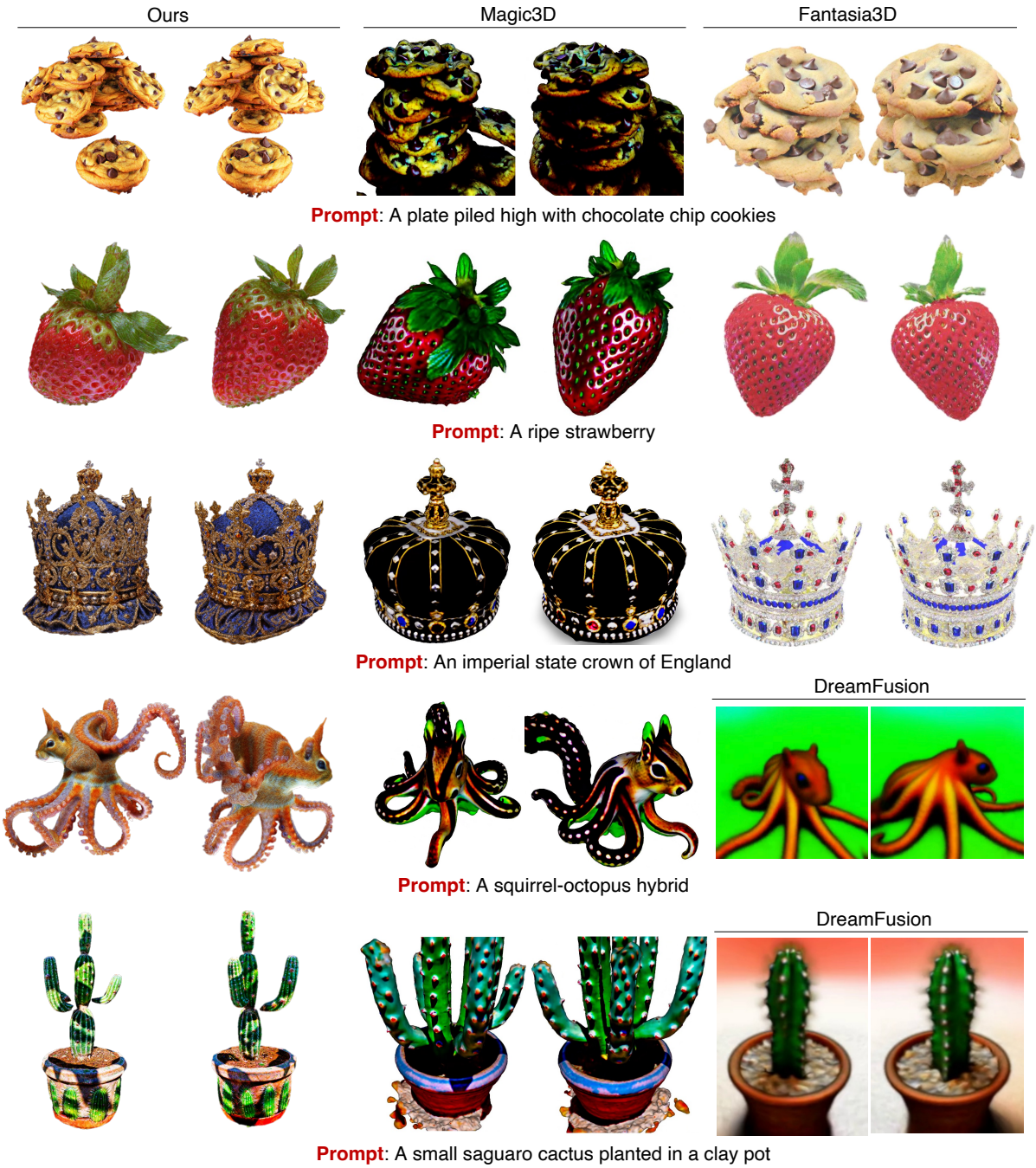

Figure 6. Result comparison. "Ours" means the results generated by the proposed approach. Matic3D [11], Fantasia [2] and DreamFusion [19] are the SOTA methods.

coordinates to their corresponding 2D positions on the output image, the rendered image can be obtained by mapping the colors according to these projections. The VAE-Encoder encodes the rendered images into latent space to compute SDS loss. To facilitate convergence, we first train the model using the standard SDS loss with the diffusion prior, then we activate the proposed multiple noise estimation for continued training until completion

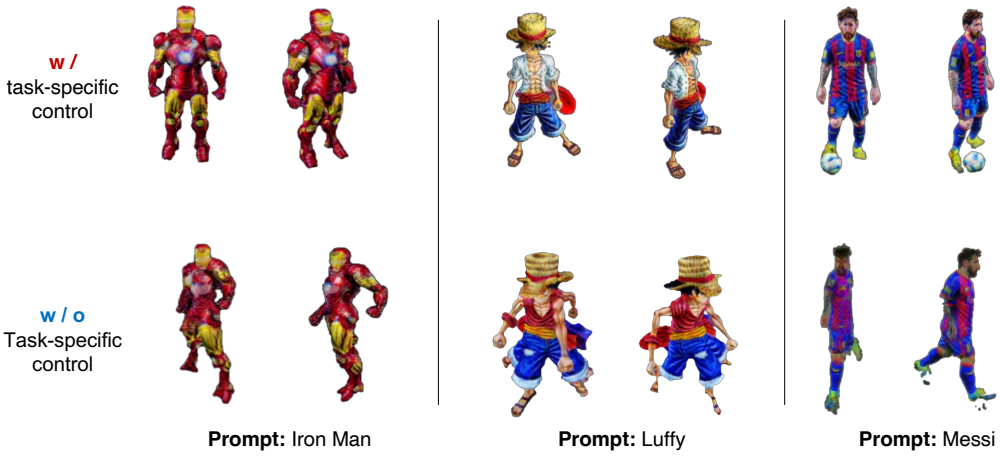

Figure 7. Pose-guidance is employed using ControlNet for 3D human character generation. The results are produced from stage 1 of the proposed system. It is seen that the results (top) generated with the guidance are better than the ones (bottom) generated without the guidance.

4. Experiments

Result comparison In this section, we conduct thorough experiments to evaluate the proposed approach. We first compare the results produced with and without the proposed multiple noise estimation. It is seen in Fig. 5 that the proposed multiple noise estimation approach outperformed the baseline in terms of high quality with enhanced details. Next, we compare the proposed approach with the SOTA methods which are Magic3D [11] and Fantasia3D [2], using identical prompts. For a fair comparison with the SOTA methods, images borrowed from their original papers are used for comparison, with our best efforts made to preserve image quality. It is seen from Fig. 6 that the proposed approach generates better 3D models than the SOTA methods in terms of high quality.

Task-specific guidance We optionally employ task-specific guidance in the proposed approach. We take the 3D human character generation task as an example to study how it affects performance. In the experiment, we employed pose guidance, implemented using ControlNet, to address the Janus problem. The results obtained from stage 1 of our approach are shown in Fig. 7. Notably, we omitted the training in stage 2 because the results produced from stage 1 are sufficiently valid for this study. It is observed that the inclusion of pose guidance effectively mitigates geometric distortions in the generated 3D characters. We believe task-specific guidance plays an important role in generating high-quality 3D content and is a worthwhile subject for in-depth future studies.

Pixel stride As depicted in Fig. 2, we utilize a sliding window operator in our proposed approach to produce overlapping patches from a noisy latent. It is worthwhile to investigate how the pixel stride of the sliding window impacts the generation of high-quality details. We study this by using four distinct pixel strides: 16, 32, 48, and 64. Notably, configuring the pixel stride to 16, 32, or 48 yields overlapping tiles, while a pixel stride of 64 results in non-overlapping tiles. The results are shown in Fig .8. We observe that the results generated with overlapping tiles (that is, when the pixel stride is set to 16, 32, or 48) are comparably good. On the other hand, the result produced without overlapping tiles is of lesser quality.

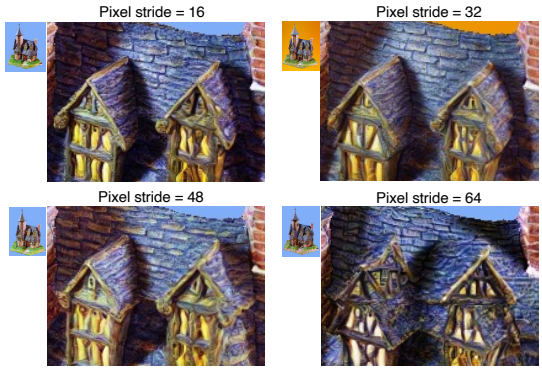

Figure 8. Examples produced with four configurations of pixel stride.

4.1. Conclusion

In this paper, we have proposed a multiple noise estimation approach that enables memory-efficient training for 3D generation within a high-resolution rendering space. We have evaluated the proposed approach through experiments and demonstrated that the proposed approach is effective in generating high-quality 3D models with enhanced details. In addition, we have presented an entire Text-to-3D system that leverages the proposed approach and ControlNet for geometrically correct 3D content generation through 2D diffusion priors.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *In Proc. International Conference on Machine Learning*, 2023. 1, 2, 3

[2] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *arXiv preprint arXiv:2303.13873*, 2023. 3, 5, 7, 8

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *In Proc. European Conference on Computer Vision*, pages 628–644. Springer, 2016. 2

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. 2021. 2

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *In Proc. Conference on Computer Vision and Pattern Recognition*, 2021. 4

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *In Proc. International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020. 2, 3, 4

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[8] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000. 2

[9] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 4, 5

[10] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760. 5

[11] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *In Proc. Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 4, 7, 8

[12] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[13] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *In Proc. Conference on Computer Vision and Pattern Recognition*, 2023. 1, 3

[14] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *In Proc. European Conference on Computer Vision*, 2020. 2

[15] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 4

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

[17] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *In Proc. International Conference on Machine Learning*, 2021. 2

[18] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics*, 36(6):1–11, 2017. 2

[19] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 3, 4, 5, 7

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *In Proc. Conference on Computer Vision and Pattern Recognition*, 2022. 2

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *In Proc. Conference on Computer Vision and Pattern Recognition*, 2022. 2, 4

[22] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35:151–173, 1999. 2

[23] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *In Proc. International Conference on Neural Information Processing Systems*, 2021. 1, 3, 5

[24] Jie Shi, Chenfei Wu, Jian Liang, Xiang Liu, and Nan Duan. Divae: Photorealistic images synthesis with denoising diffusion decoder. *arXiv preprint arXiv:2206.00386*, 2022. 2

[25] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *In Proc. Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[26] 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. 2

[27] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *In Porc. Conference on Computer Vision and Pattern Recognition*, 2019. 2

[28] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. In *In Proc. International Conference on Computer Vision*, 1998. 2

[29] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction.

In *In Proc. International Conference on Neural Information Processing Systems*, 2021. 2, 5

[30] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *arXiv preprint arXiv:2305.16213*, 2023. 2, 3

[31] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *In Proc. International Conference on Learning Representations*, 2021. 2

[32] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *arXiv preprint arXiv:2302.05543*, 2023. 2