

GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models

Osama Salem
Technical University of Munich
Munich, Germany

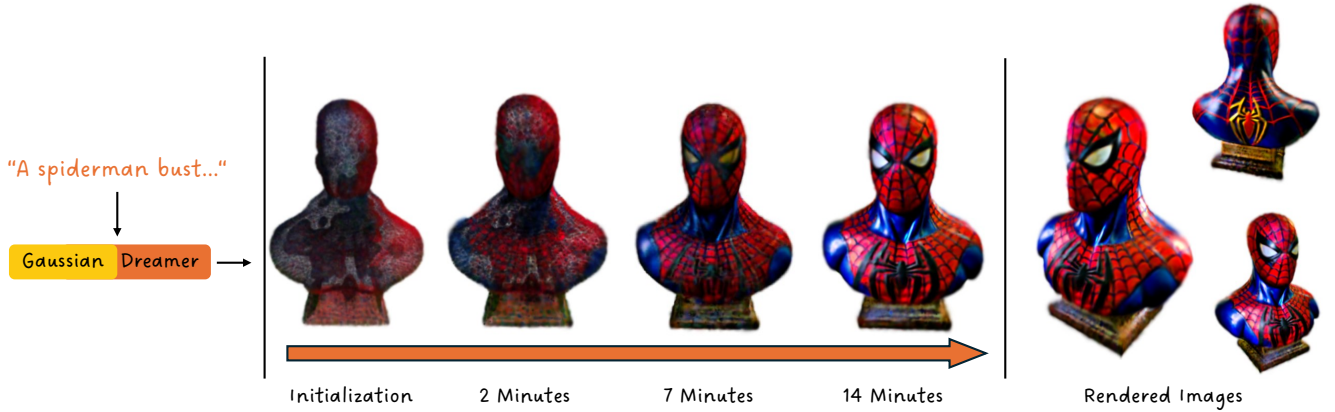


Figure 1. **GaussianDreamer** is a simple yet efficient framework that bridges 3D and 2D diffusion models using Gaussian splatting, enabling both 3D consistency and detailed generation. The system trains in under 15 minutes on a single GPU and supports real-time rendering.

Abstract

*Efficient, high-quality 3D content creation remains a central challenge in computer vision. While optimization-based techniques like Score Distillation Sampling (SDS) offer quality results, they suffer from slow per-sample optimization. Recent work on 3D Gaussian Splatting introduces a fast and expressive explicit representation, enabling new generative approaches. In this paper, we present **GaussianDreamer**, a method that combines 3D priors from diffusion models with the flexibility of 2D diffusion guidance. Unlike traditional NeRF-based pipelines requiring intricate pruning, Gaussian Splatting naturally supports progressive densification via point growing and color perturbation, accelerating convergence. Our method achieves rapid generation of high-fidelity, real-time renderable 3D assets in minutes.*

1. Introduction

The creation of 3D digital assets has broad applications but remains resource-intensive. Recent progress in diffusion models [18] has revolutionized 2D generation, prompting interest in extending these models to 3D, especially for

tasks like text-to-3D synthesis.

Two main paradigms have emerged. The first trains diffusion models directly on 3D data [4, 13], offering geometric consistency but limited generative quality due to data scarcity. The second, “2D lifting,” [8, 16, 22] leverages powerful 2D diffusion models to guide 3D representations such as NeRFs [12] via Score Distillation Sampling (SDS). While effective in appearance generation, these methods suffer from expensive volumetric rendering and slow optimization. Even acceleration techniques like pruning struggle under the weak supervision provided by SDS.

We propose **GaussianDreamer**, a text-to-3D framework built on the recent 3D Gaussian Splatting representation. This explicit and efficient format provides a strong geometric prior and supports fast rendering and progressive densification. Our method combines 3D priors from diffusion models with the expressiveness of 2D diffusion guidance, all through splatting. We first initialize geometry with a 3D diffusion model, convert it to 3D Gaussians, and apply noisy **point growing** and **color perturbation** to enable richer optimization under SDS.

Our main contributions are:

- Introduction of a hybrid framework that combines 3D and 2D diffusion models via 3D Gaussian Splatting for effi-

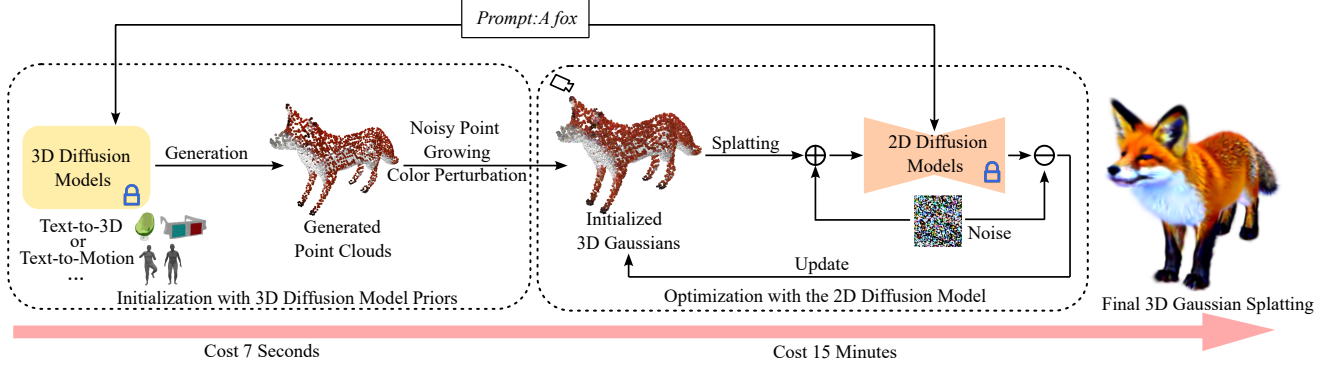


Figure 2. We start by generating an initial point cloud using a 3D diffusion model (e.g., text-to-3D or text-to-motion). After applying noise and color perturbations, we initialize 3D Gaussians, which are then optimized with SDS using a 2D diffusion model. The final images are rendered via 3D Gaussian Splatting.

cient text-to-3D generation.

- Proposing a noisy point growing and color perturbation to enhance the initialization for SDS optimization.
- Achieving high-quality 3D asset generation in under 15 minutes on a single GPU, enabling direct real-time rendering.

2. Related Work

Recent advances in 3D content generation have largely followed two paths: (1) training diffusion models directly on 3D data, and (2) lifting pretrained 2D diffusion models to operate in 3D.

3D Diffusion Models Models like Point-E [13] and Shap-E [4] generate 3D assets from text using paired text-3D data. These approaches offer fast inference but are limited by dataset size and lack fine-grained detail or textures. Some models also extend to motion synthesis, often using SMPL [9] representations. **GaussianDreamer** leverages such models (e.g., Shap-E) to initialize coarse 3D geometry, which is then refined.

Lifting 2D Diffusion Models Inspired by theoretical analysis of diffusion model score functions, DreamFusion [16] introduced Score Distillation Sampling (SDS), which uses gradients from a 2D diffusion model to optimize a 3D NeRF representation. This enables high-quality detail generation by leveraging strong 2D priors. Follow-up works improved view consistency using CLIP [17] or semantic guidance, but NeRF-based methods remain slow and computationally intensive due to volumetric rendering.

3D Representations While NeRFs offer expressive appearance modeling, their rendering cost is high. Mesh- and point-based alternatives improve efficiency but often trade off quality. Recently, 3D Gaussian Splatting [5] has emerged as a fast, explicit representation supporting

real-time rendering. Works like DreamGaussian [19] apply splatting for image- or text-conditioned 3D generation. **GaussianDreamer** adopts this representation to accelerate optimization and enable high-fidelity, text-aligned 3D asset generation via 2D SDS and 3D priors.

3. Method

Our approach, as illustrated in Fig. 2, combines 3D Gaussian Splatting with diffusion priors to efficiently generate high-quality 3D assets. We first revisit Score Distillation Sampling (SDS) and 3D Gaussian Splatting (3DGS), then detail our pipeline including initialization and optimization.

3.1. Preliminaries

Score Distillation Sampling (SDS) SDS [16] guides the optimization of a 3D model’s parameters θ by leveraging a pretrained 2D diffusion model ϕ . Given a differentiable renderer g producing image $\mathbf{x} = g(\theta)$, SDS defines the gradient:

$$\nabla_{\theta} \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t, \epsilon} \left[w(t) (\hat{\epsilon}_{\phi}(\mathbf{z}_t; \mathbf{y}, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta} \right], \quad (1)$$

where $\mathbf{z}_t$ is a noisy version of $\mathbf{x}$ at timestep t , conditioned on text embedding $\mathbf{y}$. This gradient aligns rendered views with the 2D diffusion prior.

3D Gaussian Splatting (3DGS) 3DGS represents a scene as a set of anisotropic 3D Gaussians, each with position μ_i , covariance Σ_i , color $\mathbf{c}_i$, and opacity α_i . Rendering projects these Gaussians onto the image plane, alpha-blending their contributions along each pixel ray $\mathbf{r}$:

$$\mathbf{C}(\mathbf{r}) = \sum_{i \in \mathcal{N}} \mathbf{c}_i \sigma'_i \prod_{j=1}^{i-1} (1 - \sigma'_j), \quad (2)$$

where σ'_i is the screen-space opacity of Gaussian i .

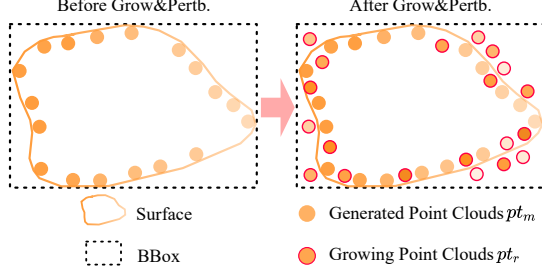


Figure 3. Illustration of the point cloud augmentation process. New points $\mathbf{p}_r$ are sampled within the bounding box and added near the original surface points $\mathbf{p}_m$, with color noise applied to enhance variation.

3.2. Framework Overview

Given a text prompt y , we first generate a coarse 3D mesh m using a pretrained 3D diffusion model. From m , a sparse point cloud $\mathbf{p}_m$ is sampled and augmented via noisy point growing and color perturbation to produce a dense point set $\mathbf{p}_f$. This set initializes 3D Gaussians θ_b , which are then optimized through SDS with a 2D diffusion model to yield the final representation θ_f . The output supports real-time splatting-based rendering.

3.3. Initialization with 3D Diffusion Priors

We leverage two 3D diffusion model types:

- **Text-to-3D:** Models like Shap-E output implicit SDFs converted to meshes via Marching Cubes [10]; vertex colors are obtained from the model.
- **Text-to-Motion:** Motion diffusion models produce posed meshes (e.g., SMPL), with colors assigned uniformly if absent.

Vertices and colors form the initial point cloud $\mathbf{p}_m$, centered if pose-based.

3.4. Point Cloud Augmentation

As illustrated in Fig. 3, we enrich the initial point cloud $\mathbf{p}_m$ through two key steps:

1. **Noisy Point Growing:** Points $\mathbf{p}_r$ are uniformly sampled within the bounding box of $\mathbf{p}_m$ and retained if they lie within a small distance threshold of existing points.
2. **Color Perturbation:** Each retained point inherits the color of its nearest neighbor in $\mathbf{p}_m$, with added random noise to introduce variation.

The resulting augmented point cloud $\mathbf{p}_f = (\mathbf{p}_f, \mathbf{c}_f)$ is used to initialize the 3D Gaussians: the means μ and colors $\mathbf{c}$ are directly set, opacities α are initialized to a small constant (e.g., 0.1), and covariances Σ are estimated from local point density. This densification step enables the Gaussians to capture finer geometric and appearance details, as shown in Fig. 4, which compares results before and after augmentation.



Figure 4. Qualitative comparison showing the effect of point growing and color perturbation. GaussianDreamer (right) produces more detailed and coherent 3D assets compared to the version without augmentation (left).

3.5. Optimization with 2D Diffusion

Initialized Gaussians θ_b are refined by minimizing SDS loss using a pretrained 2D diffusion model F_{2D} . Differentiable splatting renders images from various viewpoints; gradients update Gaussian parameters $(\mu, \Sigma, \mathbf{c}, \alpha)$, enabling the model to capture rich textures and details while maintaining 3D consistency.

4. Experiments

We evaluate **GaussianDreamer** through quantitative benchmarking and implementation analysis. We first detail our experimental setup, then report results on the T3Bench benchmark, which standardizes text-to-3D evaluation.

4.1. Implementation Details

Our pipeline is built on PyTorch [15]. We use Shap-E (fine-tuned on Objaverse [2]) and MDM [20] as 3D priors, and *Stable Diffusion 2.1-base* for 2D supervision (guidance scale 100). Gaussian parameters are optimized over 1200 iterations with key learning rates: 10^{-2} (opacity), 5×10^{-5} (position), 1.25×10^{-2} (color), and separate rates for covariance scale (10^{-3}) and rotation (10^{-2}). Training takes 15 minutes at 1024^2 resolution (downsampled to 512^2 for SDS) on a single GPU (batch size 4).

Rendered views are sampled from a camera radius of 1.5–4.0 units, azimuth range of $[-180^\circ, 180^\circ]$, and elevation range of $[-10^\circ, 60^\circ]$.

4.2. Quantitative Evaluation on T³ Bench

We evaluate on **T³ Bench** [3], a standardized benchmark for text-to-3D generation, which addresses the challenge of open-ended prompt evaluation through structured categories and automated metrics.

T³ Bench includes prompts at three complexity levels:

1. **Single Object:** Simple object-centric prompts.
2. **Object + Surroundings:** Object within a minimal scene.
3. **Multi-Object:** Prompts describing multiple interacting entities.

Evaluation combines text-image similarity (via multi-view rendering) with regional consistency to assess **visual**

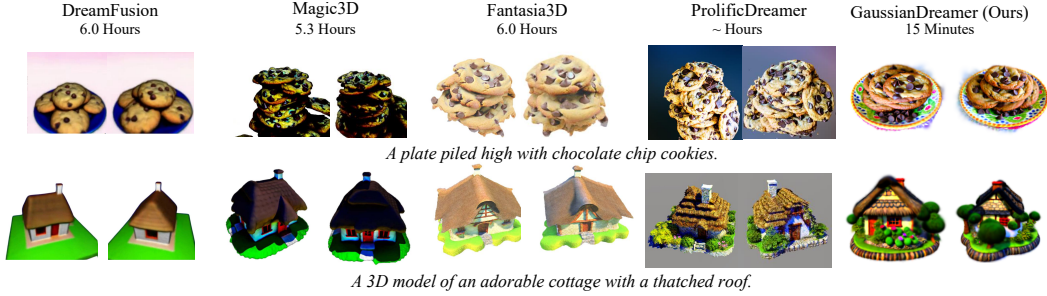


Figure 5. **Qualitative comparison of text-to-3D generation results across methods.** GaussianDreamer achieves comparable or superior visual quality to prior methods such as DreamFusion, Magic3D, and Fantasia3D, while dramatically reducing generation time to just 15 minutes. Each row shows results for the same text prompt; GaussianDreamer consistently produces coherent and detailed 3D assets with minimal computation.

Table 1. Quantitative comparisons on T³Bench.

Method	Time	Single Obj.	Single w/ Surr.	Multi Obj.	Average
SJC [21]	—	24.7	19.8	11.7	18.7
DreamFusion [16]	6 hours	24.4	24.6	16.1	21.7
Fantasia3D [1]	6 hours	26.4	27.0	18.5	24.0
LatentNeRF [11]	15 minutes	33.1	30.6	20.6	28.1
Magic3D [8]	5.3 hours	37.0	35.4	25.7	32.7
ProlificDreamer [22]	~10 hours	49.4	44.8	35.8	43.3
Ours	15 minutes	54.0	48.6	34.5	45.7

Scores above are average of two metrics (quality and alignment).

quality, and uses multi-view captioning [6] with large language model scoring (e.g., GPT-4 [14]) to assess **text alignment**. These automated metrics correlate well with human judgment.

As shown in Table 1, **GaussianDreamer** achieves competitive or superior performance across all complexity levels of the T³ benchmark, while maintaining significantly faster generation times. This highlights both its effectiveness and efficiency. In addition to strong quantitative results, Figure 5 demonstrates that **GaussianDreamer** also produces high-quality 3D assets that are visually comparable to other methods—yet at a fraction of the computational cost.

5. Conclusion

We presented **GaussianDreamer**, a framework for efficient text-to-3D generation that unifies the geometric priors of 3D diffusion models with the visual fidelity of 2D diffusion supervision, using 3D Gaussian Splatting as an expressive and real-time-friendly representation.

5.1. Key Takeaways

GaussianDreamer demonstrates notable advantages in speed, quality, and consistency. It generates high-quality 3D assets within minutes, significantly accelerating the optimization pipeline compared to prior methods. The use of explicit Gaussian primitives enables real-time rendering and strong geometric grounding, reducing common artifacts

such as the Janus problem. Our initialization strategy, involving point cloud augmentation with noisy point growing and color perturbation, ensures that the optimization begins from a well-conditioned starting point.

5.2. Limitations

Despite its speed and rendering quality, **GaussianDreamer** has practical limitations due to the nature of the 3D Gaussian Splatting representation. Gaussians lack surface continuity, making them unsuitable for direct editing, relighting, or physical simulation. They also don’t provide explicit geometry, which complicates integration into traditional 3D pipelines. Mesh extraction remains challenging and often yields suboptimal results. Additionally, the method still relies on per-prompt optimization, limiting real-time or large-scale generation.

5.3. Future Directions

Two key research directions emerge. First, tighter integration between Gaussians and explicit mesh representations—such as in GaussianDreamerPro [23]—could enable more structured asset generation suitable for downstream tasks like animation or physical simulation. Second, moving toward feed-forward generative pipelines, as explored in DiffSplat [7], offers the potential for direct 3D Gaussian generation from text. Achieving this would require large-scale training datasets and architectures tailored to Gaussian representations, but promises near-instant asset synthesis with strong text alignment.

Advancing along these paths will help bridge the gap between generative quality and practical utility, bringing us closer to real-time, controllable, and deployment-ready 3D content creation.

References

- [1] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for

- high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 4
- [2] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, pages 13142–13153, 2023. 3
 - [3] Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T³bench: Benchmarking current progress in text-to-3d generation. *arXiv preprint arXiv:2310.02977*, 2023. 3
 - [4] Heewoo Jun and Alex Nichol. Shape-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 1, 2
 - [5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 2
 - [6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. *CoRR*, abs/2201.12086, 2022. 4
 - [7] Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong Mu. Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation, 2025. 4
 - [8] 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 *CVPR*, pages 300–309, 2023. 1, 4
 - [9] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866, 2023. 2
 - [10] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIG-GRAPH Comput. Graph.*, 21(4):163–169, 1987. 3
 - [11] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, pages 12663–12673, 2023. 4
 - [12] 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 *ECCV*, pages 405–421, 2020. 1
 - [13] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 1, 2
 - [14] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Rei-ichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rim-bach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sas-try, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Val-

- lone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. [4](#)
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. [3](#)
- [16] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. [1](#), [2](#), [4](#)
- [17] 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 *ICML*, pages 8748–8763. PMLR, 2021. [2](#)
- [18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. [1](#)
- [19] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. [2](#)
- [20] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. [3](#)
- [21] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pages 12619–12629, 2023. [4](#)
- [22] 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. *arXiv preprint arXiv:2305.16213*, 2023. [1](#), [4](#)
- [23] Taoran Yi, Jiemin Fang, Zanwei Zhou, Junjie Wang, Guan-jun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Xinggang Wang, and Qi Tian. Gaussiandreamerpro: Text to manipulable 3d gaussians with highly enhanced quality, 2024. [4](#)