Contrastive Text-guided Object Generation

Yunfan Jiang
Department of Electrical Engineering
Stanford University
yjiang05@stanford.edu

Abstract

The ability to synthesize 3D objects is one key of human intelligence. Previous efforts have been made towards 3D object synthesis leveraging 3D datasets. However, these efforts are bottlenecked by the quality, size, and availability of 3D datasets. In this project, we instead explore the possibility to generate 3D objects without requiring any datasets. Leveraging recent advances in foundation models pre-trained using contrastive objectives, we learn neural radiance fields that are able to generate 3D objects described by natural language sentences, fully alleviating the burden on 3D datasets. Quantitatively, experiments show that our method generates 3D objects that are well aligned with corresponding text descriptions. Qualitatively, 3D objects generated using our method are intra-consistent as well, meaning that renderings of a same object from multiple views are consistent with each other. Our code is publicly available at https://github.com/yunfanjiang/clip-nerf.

1. Introduction

The ability to synthesize 3D objects is one key of human intelligence. Combined with prior knowledge, humans are able to imagine 3D shapes of various objects given only a limited number of images from restricted views. Trained experts can even produce 3D objects with high fidelity. Prior work has leveraged 3D datasets and attempted to synthesize 3D objects in various forms [9, 21, 62, 65]. However, they are bottlenecked by the quality, size, and availability of 3D datasets. The expense to build high-quality large-scale 3D datasets even adds more difficulties towards integrating this intelligence into machines. For example, the construction of the Microsoft COCO dataset [35] involves intensive Amazon Mechanical Turk workload. They also developed complicate infrastructures for data collection and labelling. As the demand increases, the paradigm of *learning from data* will exponentially increase the amount of labor.

Recent five years have witnessed the prevalence of *foundation models*, a paradigm in which models are trained at scale and then adapted to numerous diverse downstream tasks [4]. Many researches have completely shifted to Transformer-based [59] foundation models such as BERT [16, 37], Vision Transformer [17], the GPT family [47, 48, 7], T5 [49], DALL-E [50], CLIP [46], the Perceiver family [25, 24, 22], to name a few. Universal knowledge is baked into foundation models through large-scale pre-training such that data hunger is greatly mitigated in downstream tasks. Opportunities of foundation models also bring more possibilities to 3D object synthesis.

Advances in contrastive learning [27] suggests that learning from contrastive objectives is better than learning from "gold" labels. [46, 50] pioneered the efforts in language-image contrastive learning. Other work also shows its efficiency and superiority in decision making [54], video-text understanding [63], and so on. Given that language can encode everything about the world as efficiently as possible, it is natural to ask "can we leverage language-image contrastive learning for 3D object generation, *without* requiring any 3D datasets?".

In this project, we explore the problem of text-guided 3D object generation using contrastive learning. Concretely, we aim to learn Neural Radiance Fields (NeRFs) [39] that can generate 3D renderings of objects described by natural language sentences. We optimize NeRFs from contrastive loss between rendered images and corresponding text descriptions provided by pre-trained CLIP models [46]. Our method is able to generate 3D renderings that are both semantically consistent and intra-consistent. The most relevant work to us is [26]. We share similar motivation and high-level idea. But we deviate from [26] in detailed implementations. For example, their implementation is based on TensorFlow Jax [6]. While our implementation is completely based on PyTorch [43]. Our contributions are twofold.

1. We achieve 3D object generation purely guided by text prompts, completely removing the burden on expensive 3D datasets.

2. We evaluate the generated results to be semantically consistent with corresponding language descriptions and intra-consistent with renderings from other views.

2. Related Work

2.1. Novel View Synthesis

Prior to the introduction of NeRF [39], approaches for novel view synthesis can be classified into two categories. One category uses mesh-based representations of scenes [60, 8, 15, 61, 13, 20, 36]. Another class of methods uses volumetric representations to synthesize views from RGB images [18, 23, 30, 38, 45, 56]. However, the former category of methods suffers from optimization difficulties caused by local minima and poor conditioning. The latter suffers from poor time and space complexity, preventing these methods from generating images with higher resolutions or more details.

A recent trend for novel view synthesis is to use coordinate-based neural representations. NeRF [39] represents 3D scenes as continuous functions parameterized by MLPs. It maps from 3D coordinates (plus 2D viewing directions) to properties including RGB color and density at that location. It has inspired numerous follow-up work for generative synthesis [10, 53], synthesis of dynamic seneces [34, 40], deformable objects [19, 42], relighting [3, 5, 55], to name a few.

2.2. Foundation Models

A foundation model refers to any model that is pre-trained at scale then adapted to downstream tasks through, for example, fine-tuning [4]. Since the introduction of the Transformer model [59], various Transformer-based models have dominated different research areas, e.g., Vision Transformer [17, 57] for computer vision tasks, the GPT family [47, 48, 7] for natural language processing tasks, the DALL-E [50] model, the CLIP [46] model, T5 [49], Decision Transformer [12] and Trajectory Transformer [28] for offline reinforcement learning, the Perceiver family [25, 24, 22] for processing multi-modality. Universal knowledge is backed into these models through large-scale pre-training. Data hunger is greatly mitigated when fine-tuning them for downstream tasks. Recent work show the advantages of fine-tuning on large-scale pre-trained models for offline reinforcement learning [51], multi-modal few-shot learning [58], just to name a few.

2.3. Language-image Contrastive Learning

The goal of contrastive learning is to learn a feature space in which similar data pairs are close and dissimilar pairs are far [27]. A recent advance in contrastive learning, the CLIP model [46], has demonstrated great power of language-image contrastive pre-training. They learn an image encoder and a text encoder such that extracted image features and text features from paired images and sentences are similar. The trained encoders can be zero-shot transferred to downstream tasks. Follow-up works also show that it can be used in decision making [54], video-text understanding [63], and so on.

Previous work leveraging image-text semantic models such as CLIP that is relevant to the topic we study includes CLIP-Forge [52], Text2Shape [11], anamorphic art modelling [14], ClipMatrix [29] StyleCLIP [44], to name a few. Among them, [52] conditions normalizing flow models on CLIP embeddings to generate object geometries. [11] aims to synthesize novel voxel objects by learning a text-conditioned Wasserstein GAN [1]. [29] leverages CLIP models to create deformable humanoid meshes. [44] uses CLIP models to guide the learning of StyleGAN [31]. All these efforts suggest that leveraging guidance from semantic models such as CLIP is promising for object generation and synthesis.

3. Methodology

In this section, we first introduce preliminaries about NeRF. Then we present our method for text-guided object generation. A high-level sketch of our method is provided in Figure 1.

3.1. NeRF Preliminaries

Denoting the parameter of a MLP as θ, NeRF learns this MLP to represent a scene as a continuous volumetric field of particles that block and emit light. Given a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ emitted from the camera center $\boldsymbol{o}$ along the direction $\boldsymbol{d}$, for each distance $t_k \in \boldsymbol{t}$, NeRF compute its corresponding 3D position $\boldsymbol{x} = \boldsymbol{r}(t_k)$. All positions are then transformed using position encoding

$$\gamma(\boldsymbol{x}) = [\sin(\boldsymbol{x}), \cos(\boldsymbol{x}), \ldots, \sin(2^{L-1}\boldsymbol{x}), \cos(2^{L-1}\boldsymbol{x})]^\top, \tag{1}$$

where L is a hyperparameter.

The MLP parameterized by θ then takes input of $\gamma(\boldsymbol{r}(t_k))$ and view direction (ψ, ϕ) and outputs an RGB color $\boldsymbol{c}$ and a density σ. The final predicted color of the pixel $\boldsymbol{C}(\theta; \boldsymbol{r}, \boldsymbol{t})$ is then obtained from

$$\boldsymbol{C}(\theta; \boldsymbol{r}, \boldsymbol{t}) = \sum_k T_k(1 - \exp(-\tau_k(t_{k+1} - t_k)))\boldsymbol{c}_k, \tag{2}$$

where

$$T_k = \exp(\sum_{k'<k} \tau_{k'}(t_{k'+1} - t_{k'})). \tag{3}$$

The classical way to train a NeRF is then to use gradient descent to optimzie the sum of squared differences between predicted pixel values and ground-truth values with known

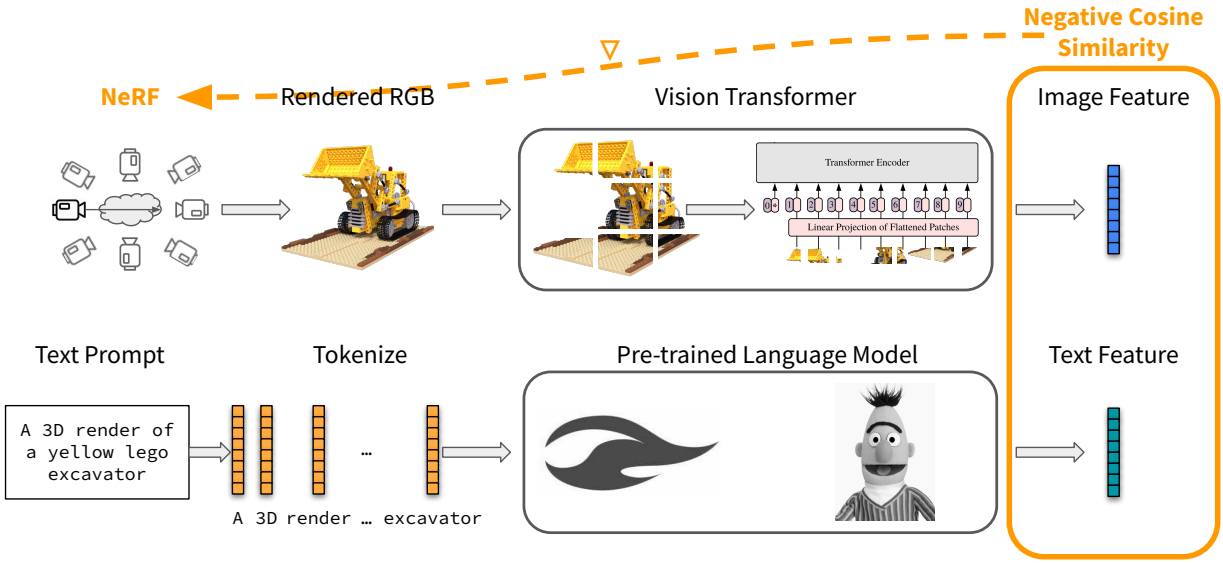

Figure 1: Contrastive text-guided object generation. We learn a NeRF model that can generate 3D renderings of objects described by natural language sentences. We leverage a semantic model, such as a CLIP model, to produce the semantic alignment loss, e.g., negative Cosine similarity, between image feature and text feature.

camera poses. In the following subsection, we introduce our method that guide the optimization using semantic alignment with natural language descriptions.

3.2. Contrastive Text-guided Object Generation

Our method, contrastive text-guided object generation as illustrated in Figure 1, aims to learn a NeRF model parameterized by θ that can generate 3D renderings of objects described by natural language sentences. Given a image-text semantic model such as a CLIP model [46] with an image encoder $g(\cdot) : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{hidden}$ and a text encoder $h(\cdot) : \mathbb{R}^{L_{text}} \rightarrow \mathbb{R}^{hidden}$, denoting the RGB image generated by the NeRF model at a specific pose $\boldsymbol{p} = [\psi, \phi]$ as $\boldsymbol{I}(\theta, \boldsymbol{p})$ and corresponding text description as $\boldsymbol{y}$, the semantic alignment between $\boldsymbol{I}$ and $\boldsymbol{y}$ can be computed

$$\mathcal{J}(\theta; \boldsymbol{p}, \boldsymbol{y}) = g(\boldsymbol{I}(\theta, \boldsymbol{p}))^\top h(\boldsymbol{y}). \tag{4}$$

We then take the negation of $\mathcal{J}$ as the loss function $\mathcal{L}(\theta; \boldsymbol{p}, \boldsymbol{y}) = -\mathcal{J}(\theta; \boldsymbol{p}, \boldsymbol{y})$ and optimize our NeRF model using gradient descent

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}, \tag{5}$$

where α denotes the learning rate.

Notice that an object is supposed to be consistent with its corresponding text description despite of different view directions, the same natural language sentence is used to guide the generation of renderings from multiple different views. Our method is summarized in Algorithm 1.

In practice, we use Vision Transformer [17] as the image encoder for better feature representations. We use state-of-the-art language models such as BERT [16, 37] or models from the GPT family [47, 48, 7] to extract language representations.

4. Experiments and Evaluations

In this section, we start with the experimental details. We then elaborate the special evaluation protocol for our problem of text-guided object generation. Finally, we present quantitative and qualitative results.

4.1. Experimental Details

We parameterize the NeRF model as a MLP with 8 layers. Each layer has 256 hidden units. We sample 64 coarse samples per ray. As for the image-text semantic model, we use CLIP variant `ViT-B/16` during training. We use Adam optimizer [32] with an initial learning rate of 5×10^{-4}. It is then exponentially decayed during the train-

Algorithm 1: Contrastive Text-guided Object Generation

Input: Learning rate α, all poses $\boldsymbol{P}$, text description $\boldsymbol{y}$, image encoder $g(\cdot)$, and text encoder $h(\cdot)$.

Randomly initialize NeRF parameters θ;
while *not done* **do**
  forall *pose* $\boldsymbol{p}$ *in all poses* $\boldsymbol{P}$ **do**
    Render image $\boldsymbol{I}(\theta, \boldsymbol{p})$ according to Equations 1, 2, and 3; Compute the semantic alignment loss for pose $\boldsymbol{p}$: $\mathcal{L}(\theta; \boldsymbol{p}, \boldsymbol{y}) = -g(\boldsymbol{I}(\theta, \boldsymbol{p}))^\top h(\boldsymbol{y})$
  end
  Collect losses from all poses $\mathcal{L}(\theta; \boldsymbol{P}, \boldsymbol{y})$;
  Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta; \boldsymbol{P}, \boldsymbol{y})$;
end

ing. We use images from *Real images of complex scenes* dataset introduced in [39] as ground-truth images. Due to compute and time constraints, we experiment with `Lego`, `Orchid`, and `Fern` in this work. We defer more comprehensive experiments to future work.

4.2. Evaluation Protocol

Our method can be evaluated qualitatively and quantitatively. We discuss our quantitative evaluation protocol.

Conventionally, 3D reconstruction methods are evaluated by comparing the learned geometry with a ground-truth reference model. However, for novel view synthesis techniques such as NeRF, we do not have such ground-truth models. Nevertheless, they can still be evaluated by comparing rendered images with pixel-aligned ground truth images from held-out sets. However, it is challenging to evaluate our method in this manner because we do not have diverse captioned multi-view data. The same problem exists in [26] as well. Instead, we use the CLIP R-Precision metric [41]. Concretely, we use a CLIP variant `Vit-B/32` [46] as an evaluator. Note that the CLIP variant used for evaluation is different from that used during training. As demonstrated in Figure 2, the evaluator computes the similarity scores between renderings generated from our trained model and the text descriptions. It also computes the same similarity scores between images from the dataset and the same text descriptions. The latter scores serve as the oracle for us to evaluate the performance of our trained model. The scores are scaled to the range of $[-1, 1]$ for better readability and interpretability. A score value of -1 means "completely dissimilar" (visually, two vectors are opposite). A score value of 0 means "orthogonal" (visually, two vectors are perpendicular). A score value of 1 means "exactly same" (visually, two vectors are parallel).

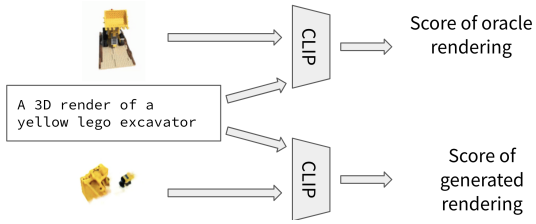

Figure 2: Quantitative evaluation protocol. We use the CLIP variant `Vit-B/32` as an evaluator. It first computes the similarity scores between renderings generated from our trained model and the text descriptions. It also computes the same similarity scores between images from the dataset and the same text descriptions.

4.3. Quantitative Results

Table 1 presents the quantitative results of our method for three text prompts, namely 1) `A 3D render of a yellow lego excavator`, 2) `A 3D render of a red orchid`, and 3) `A 3D render of a green fern`. In cases where the described objects should be rendered realistically, e.g., the yellow lego excavator, the score of generated renderings is lower than the score of oracle renderings. However, in cases where the described objects do not have complicate textures or shapes, e.g., the red orchid and green fern, scores of generated renderings are higher than scores of oracle renderings. Notice that the Cosine similarity scores of generated renderings are all above zero, indicating that our method is able to follow the text guidance to generate objects that are semantically consistent with corresponding language descriptions.

4.4. Qualitative Results

We show renderings from multiple views for `Lego`, `Orchid`, and `Fern` in Figures 3, 4, and 5, respectively. Despite of different view directions, renderings are consistent with each other, indicating that our method is able to generate renderings that are intra-consistent. We note that corresponding videos can be found in supplementary materials.

5. Discussion, Limitations, and Future Work

Recent work shows that different modalities are not orthogonal to each other. Knowledge in one domain can be transferred to another domain and improve the performance. For example, [58] shows that the knowledge in pre-trained language models can be transferred to a multimodal setting (vision + language). [51] shows that the knowledge in WikiPedia baked in language models can improve performances in offline reinforcement learning [33]. In this work, we step towards the goal of unifying modalities by showing that language models can guide the learn-

Text Prompt	A 3D render of a yellow lego excavator	A 3D render of a red orchid	A 3D render of a green fern
Oracle	**0.3717 ± 0.0230**	0.2521 ± 0.0058	0.2540 ± 0.0067
Generated	0.3234 ± 0.0155	**0.3020 ± 0.0133**	**0.3299 ± 0.0119**

Table 1: CLIP R-Precision of oracle renderings and generated renderings.

Figure 3: 3D rendering generated from text prompt `A 3D render of a yellow lego excavator`.

Figure 4: 3D rendering generated from text prompt `A 3D render of a red orchid`.

Figure 5: 3D rendering generated from text prompt `A 3D render of a green fern`.

ing of NeRFs. However, limitations still exist. First, the current implementation only associates the learning of the NeRF model and the language descriptions through back-propagated gradients that are obtained from semantic similarity loss. We have to learn separate NeRF models for different text prompts. This is compute intensive. One direction of future work is to condition NeRF models on natural language sentences such that we do not needs to train separate models for different prompts. Second, the quality of generated renderings need to be improved. Figure 3 shows that the generated renderings are not realistic enough for relatively complicated objects. [26] also includes trans-

mittance loss to encourage sparsity and tunes the architecture of NeRF models including using Mip-NeRF [2]. These are potentially helpful for better generation quality. Third, more investigation can be carried to analyze how the trained models work. One promising direction is to explore compositional generation [64, 50].

6. Conclusions

We explore the problem of text-guided object generation in this work. We propose a method that leverages image-text semantic models to guide the learning of NeRF models. Our results are both semantically consistent and intra-consistent, demonstrating the possibility to transfer knowledge baked in language models to novel view synthesis. We then identify limitations of the current implementation and suggest possible directions for future work. We highlight the compute inefficiency of the current implementation due to the fact that the NeRF model is not actually conditioned on text prompts. We also suggest possible venues for future work, particularly to improve generation quality for objects with complicate spatial or textural properties. Investigation the compositional generation is another promising direction as well. We hope this work can incite future efforts towards more efficient knowledge transfer between different modalities and domains.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017. 2

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5835–5844, 2021. 6

[3] S. Bi, Z. Xu, P. P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Havsan, Y. Hold-Geoffroy, D. J. Kriegman, and R. Ramamoorthi. Neural reflectance fields for appearance acquisition. *ArXiv*, abs/2008.03824, 2020. 2

[4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. S. Chatterji, A. S. Chen, K. Creel, J. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. E. Gillespie, K. Goel, N. D. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. F. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. S. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. P. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. F. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. H. Roohani, C. Ruiz, J. Ryan, C. R'e, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. P. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. A. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258, 2021. 1, 2

[5] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12664–12674, 2021. 2

[6] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 1

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. J. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 1, 2, 3

[8] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and In-*

teractive Techniques, SIGGRAPH '01, page 425–432, New York, NY, USA, 2001. Association for Computing Machinery. 2

[9] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. J. Belongie, N. Snavely, and B. Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020. 1

[10] E. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5795–5805, 2021. 2

[11] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. A. Funkhouser, and S. Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *ArXiv*, abs/1803.08495, 2018. 2

[12] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *ArXiv*, abs/2106.01345, 2021. 2

[13] W. Chen, J. Gao, H. Ling, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. 2

[14] E. Chu. Evolving evocative 2d views of generated 3d objects. *ArXiv*, abs/2111.04839, 2021. 2

[15] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 11–20, New York, NY, USA, 1996. Association for Computing Machinery. 2

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. 1, 3

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021. 1, 2, 3

[18] J. Flynn, M. Broxton, P. E. Debevec, M. DuVall, G. Fyffe, R. S. Overbeck, N. Snavely, and R. Tucker. Deepview: View synthesis with learned gradient descent. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2362–2371, 2019. 2

[19] G. Gafni, J. Thies, M. Zollhofer, and M. Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8645–8654, 2021. 2

[20] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman. Unsupervised training for 3d morphable model regression. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 2

[21] K. Gupta and M. Chandraker. Neural mesh flow: 3d manifold mesh generationvia diffeomorphic flows. *ArXiv*, abs/2007.10973, 2020. 1

[22] C. Hawthorne, A. Jaegle, C. Cangea, S. Borgeaud, C. Nash, M. Malinowski, S. Dieleman, O. Vinyals, M. M. Botvinick, I. Simon, H. R. Sheahan, N. Zeghidour, J.-B. Alayrac, J. Carreira, and J. Engel. General-purpose, long-context autoregressive modeling with perceiver ar. *ArXiv*, abs/2202.07765, 2022. 1, 2

[23] P. Henzler, V. Rasche, T. Ropinski, and T. Ritschel. Single-image tomography: 3d volumes from 2d x-rays. *ArXiv*, abs/1710.04867, 2017. 2

[24] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, A. Brock, E. Shelhamer, O. J. H'enaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver io: A general architecture for structured inputs & outputs. *ArXiv*, abs/2107.14795, 2021. 1, 2

[25] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 1, 2

[26] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole. Zero-shot text-guided object generation with dream fields. *ArXiv*, abs/2112.01455, 2021. 1, 4, 5

[27] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *ArXiv*, abs/2011.00362, 2020. 1, 2

[28] M. Janner, Q. Li, and S. Levine. Reinforcement learning as one big sequence modeling problem. *ArXiv*, abs/2106.02039, 2021. 2

[29] N. Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes. *ArXiv*, abs/2109.12922, 2021. 2

[30] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In *NIPS*, 2017. 2

[31] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 2

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 3

[33] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *ArXiv*, abs/2005.01643, 2020. 4

[34] Z. Li, S. Niklaus, N. Snavely, and O. Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6498–6508, June 2021. 2

[35] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[36] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7707–7716, 2019. 2

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. 1, 3

[38] B. Mildenhall, P. P. Srinivasan, R. O. Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion. *ACM Transactions on Graphics (TOG)*, 38:1 – 14, 2019. 2

[39] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2, 4

[40] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide. Neural scene graphs for dynamic scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2855–2864, 2021. 2

[41] D. H. Park, S. Azadi, X. Liu, T. Darrell, and A. Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 4

[42] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5845–5854, 2021. 2

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[44] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2065–2074, 2021. 2

[45] E. Penner and L. Zhang. Soft 3d reconstruction for view synthesis. *ACM Trans. Graph.*, 36(6), nov 2017. 2

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 4

[47] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. 2018. 1, 2, 3

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019. 1, 2, 3

[49] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683, 2020. 1, 2

[50] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021. 1, 2, 6

[51] M. Reid, Y. Yamada, and S. S. Gu. Can wikipedia help offline reinforcement learning? *ArXiv*, abs/2201.12122, 2022. 2, 4

[52] A. Sanghi, H. Chu, J. Lambourne, Y. Wang, C.-Y. Cheng, and M. Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *ArXiv*, abs/2110.02624, 2021. 2

[53] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *ArXiv*, abs/2007.02442, 2020. 2

[54] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021. 1, 2

[55] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7491–7500, 2021. 2

[56] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely. Pushing the boundaries of view extrapolation with multiplane images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 175–184, 2019. 2

[57] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *ArXiv*, abs/2106.10270, 2021. 2

[58] M. Tsimpoukelli, J. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill. Multimodal few-shot learning with frozen language models. *ArXiv*, abs/2106.13884, 2021. 2, 4

[59] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 1, 2

[60] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3d reconstructions. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 836–850, Cham, 2014. Springer International Publishing. 2

[61] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 287–296, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 2

[62] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 1

[63] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, and F. M. L. Z. C. Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *ArXiv*, abs/2109.14084, 2021. 1, 2

[64] A. Zhou, V. Kumar, C. Finn, and A. Rajeswaran. Policy architectures for compositional generalization in control. 2022. 6

[65] L. Zhou, Y. Du, and J. Wu. 3d shape generation and completion through point-voxel diffusion. *ArXiv*, abs/2104.03670, 2021. 1