G3DR: Generative 3D Reconstruction in ImageNet

Pradyumna Reddy* Ismail Elezi* Jiankang Deng
Huawei Noah's Ark Lab UK

S.1. Hyperparameters

Optimization: We train f_{trigen} using Adam optimizer with β_1 = .9; β_2 = .99, and cosine learning rate schedule with 20K warmup steps, with an initial learning rate of 2e-6 and a max learning rate of 1e-4. Model f_{rgbd} is trained using a similar setup as LDM3D [9].
Kernel: Hyperparameters s_1 and s_2 which control the height and width of the peak of the kernel are set to 1.25 and 0.03 in our experiments. Values of c_{min} and c_{max} which determine the maximum and minimum values of the kernel, are set to 0.05 and 1.0.
Losses: As mentioned in the main paper we $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ i.e weights of reconstruction, depth, vgg, clip, tv and vgg_2 losses are 1, 2, 0.5, 0.35, 0.1, 0.5. We increase λ_4(clip) from 0.02 to its maximum value of 0.35 linearly over the first 100k steps.

S.2. Alternating between novel and canonical view

We give more details on how we alternate between the novel and canonical views. At each iteration, we randomly sample between the canonical and novel views. We first generate a random number r from a uniform distribution $\mathcal{U}(0, 1)$. Then, we decide if the view is canonical or novel based on the following equation:

$$\text{view} = \begin{cases} \text{canon}, & r > min(\tau, 2 \cdot \frac{\text{iter}}{\text{num\_iter}}) \\ \text{novel}, & \text{otherwise.} \end{cases} \quad (1)$$

where τ is a hyperparameter, iter is the current iteration and num_iter is the total number of iterations. We set τ to 0.4. Using this procedure, in the early stages of training, we sample mostly canonical views, while sampling very rarely novel views. In this way, the main task of the network becomes reconstructing the input image, a much easier task than generating novel views. In the later stages of training, the probability of sampling novel views linearly increases, until converging at 0.4. Thus, the network is tasked to generate novel views, but also to not forget to reconstruct the input image (canon view).

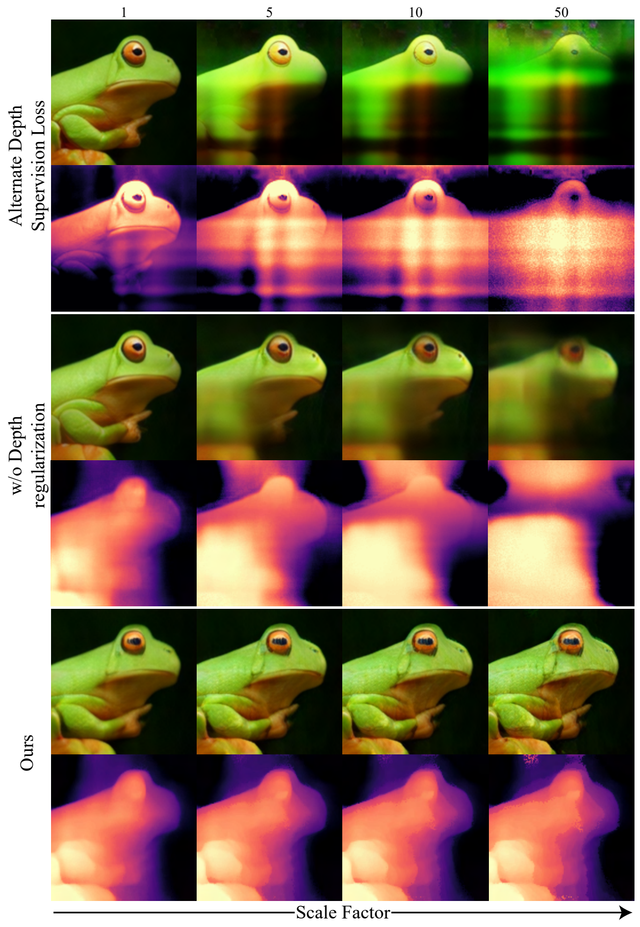

Figure S1. Visualization of f_{trigen} outputs rendered using different scale factors. Notice how the results generated by the model trained using our method degrade gracefully.

S.3. Depth regularization geometry ablation

In Fig. 5 [4] of the main paper, we show color and depth images of models trained with alternative geometric supervision loss, w/o gradient regularization and our pipeline. As alternative geometric supervision loss we use the loss function proposed in [2] which supervises the w values using the KL divergence between the w values and depth map. In the w/o gradient regularization experiment, we train by supervising the accumulated depth calculated using Eq. 4.

	Alternate Depth Supervision Loss			w/o Depth regularization			Ours		
Scaling factor	FID ↓	IS ↑	NFS ↑	FID ↓	IS ↑	NFS ↑	FID ↓	IS ↑	NFS ↑
1	44.27	83.04	28.96	13.13	116.7	36.58	13.12	151.72	36.51
5	128.64	12.26	29.85	73.34	26.72	27.97	12.98	146.30	35.24
10	174.58	5.33	26.21	104.91	14.07	25.08	14.42	131.95	35.01
50	205.15	3.58	22.42	152.52	5.94	19.38	21.37	99.59	34.54

Table S1. Table quantifying the quality of geometry learned by models trained using different methods with various scaling factors.

In both experiments, models result in naive local minima causing volume collapse, we quantify this using the depth accuracy metric in Tab. 4 of the main paper, we further quantify this phenomenon in Tab. S1.

In Tab. S1 we report FID, IS, NFS metrics for data generated by models trained using different geometric supervision and with different σ scaling factors. The scaling factor augments the predicted σ values by multiplication($\sigma_{scaled} = scale\ factor * \sigma_{predicted}$) before volume rendering. In the case of networks with volume collapse the generated 3D scenes often incorrectly model surfaces using small σ values. Scaling the σ values amplifies these small σ values essentially degrading the geometry of predictions with volume collapse.Tab. S1 shows how the network trained using our proposed depth regularization degrades gracefully compared to the other methods when increasing the scaling factor. This validates that our proposed depth regularization helps a network generate instances with better geometry than the alternatives. In Fig. S1 we visualize how the color and depth of the renders of outputs from different models change when the scaling factor is increased.

S.4. Multi-resolution sampling

In Sec. 3.3 of the main paper we mention the multi-resolution triplane sampling strategy we use to improve the generation model performance. The three levels $\{G_1, G_2, G_3\}$we use for sampling have resolution {128, 64, 32} respectively. In Fig. S2 we show a visualization of the effect of sampling from only the coarsest level to adding samples from finer levels.

S.5. Evaluation metrics

Depth accuracy: We calculate depth accuracy similar to [7]. It is defined as the MSE between rendered and pseudo-ground truth depth normalized using their respective mean and standard deviation.

In the main paper, we do not report the depth accuracy of VQ3D [6] since they normalize the predicted values using variance. This makes the final metric sensitive to their specific preprocessing and training hyperparameter setup. We will update Tab. 2 of the main paper when the code for VQ3D is available publically.

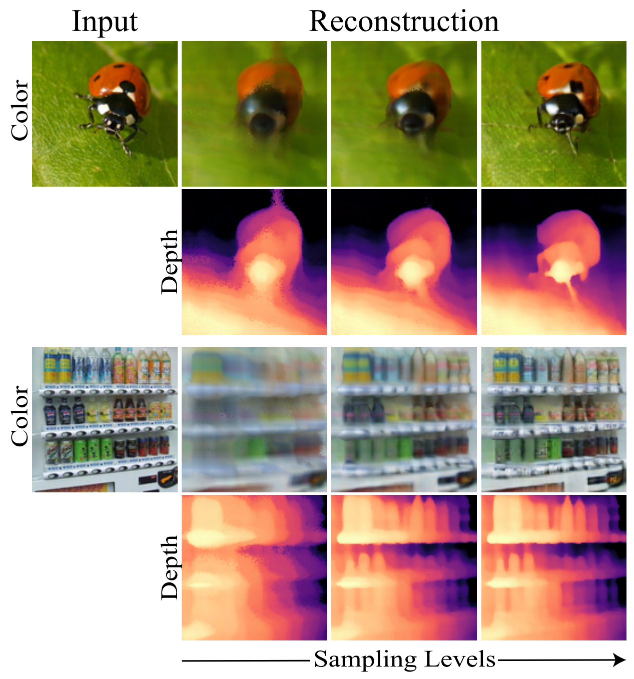

Figure S2. Images showing the effect of sampling from different levels. Left to right the images are sampled from $\{G_3, G_3, G_3\}$, $\{G_2, G_2, G_3\}$ and $\{G_1, G_2, G_3\}$.

Non-flatness score: We adopt the NFS metric from 3DGP [8] to quantify the flatness of a generator. We calculate NFS score using the eq:

$$NFS = \frac{1}{N}\sum_{i=1}^{N} exp\left[-\frac{1}{h.w}\sum_{j=1}^{B} b(d^i)_j\right] \tag{2}$$

In our experiments, we use N=256 and B=64.

FID and IS: We compute Fréchet Inception Distance(FID) [3] using similar protocol as Eg3D [1] Inception score(IS) is calculated similarly using [5].

In Tab. S2 we present all the evaluation metrics of novel view generation when using ImageNet images with f_{trigen} rather than the images generated using f_{rgbd} like in the main paper.

S.6. Video results

We include videos of results presented in the main paper and additional results. All the videos are created using cam-

	FID ↓	IS ↑	NFS ↑	DA ↑
Ours	**11.25**	**168.6**	**36.7**	**0.33**

Table S2. Quantitative evaluation of novel view generation using images from ImageNet dataset.

era parameters with yaw variation of 0.35 and pitch variation of 0.15. In the same folder, we also include the results of our method without the superresolution module in resolution 128x128. We show the same instances in as images in Fig. S4 and Fig. S5.

S.7. Other competing approaches

Our three main competing approaches are 3DGP, IVID, and VQ3D. We give a brief description of them and the main differences to our method.

3DGP [8] is the first method to tackle the problem of 3D image generation in ImageNet. Based on GANs they designed a framework that is able to generate realistically looking 3D objects from the ImageNet dataset. To do so, they used a depth estimator for geometry preservation and combined it with a flexible camera model, and a knowledge distillation module. They first sample camera parameters from a prior distribution and pass them to the camera generator. After that, they use the posterior camera parameters to render an image and its depth map, while using a depth adaptor to bridge the distribution gap between the rendered and the predicted depth. They also use an image discriminator that receives a 4-channel color-depth image, with the fake sample being color images and their adapted depth map, while the real images are Imagenet images and their estimated depth In contrast to 3DGP, our method does not use any adversarial training, knowledge distillation module, or depth adapter.

IVID [10] (ICCV 2023) uses a class-condition diffusion model to generate images. They do a forward-backward warping of rgbd images, and using another diffusion process they visually complete the images getting in process their novel-view supervision. To further improve the results, they add blur and texture erosion augmentation. During inference, they design an autoregressive process to generate the novel views, where each view is conditioned in the previous novel view. In contrast to IVID, we do not use any warping, blurring, erosion, or autoregressive processes. Their process is also prone to not estimating 3D consistent novel views.

VQ3D [6] proposes a 2-stage framework. In the first stage, they learn a model that maps images into a sequence of discrete indices that correspond to a learned latent codebook. To do so, they design several losses that ensure a good reconstructed canonical view, reasonable novel views, and correct geometry. Similar to 3DGP, the training process is done in an adversarial manner. Then in the second stage, they learn an autoregressive model over the discrete encodings produced by Stage 1 encoder, that is able to generate new 3D scenes. In contrast to them, our method does not use any adversarial training, quantized autoencoder or autoregressive modeling.

As we showed in the experimental section of the main paper, our method despite having a more simple architecture significantly outperforms the three main competing approaches in both visual quality (FID and IS) and geometry (NFS and Depth accuracy).

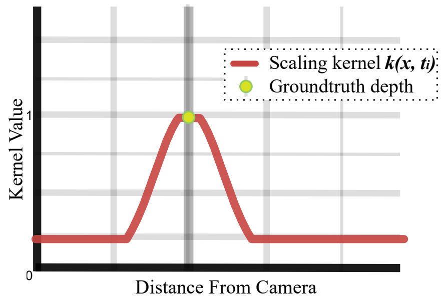

Figure S3. An illustration of our kernel shown in 1D.

S.8. 1d visual

We show a 1D illustration of the kernel in Fig. S3. The x-axis represents the distance of the sampled point along the ray from the camera, y-axis represents the magnitude of the kernel.

Broader Impact

G3DR and stronger models based on it can be used for a variety of applications, including:

- 3D modeling: G3DR can be used to generate 3D models of objects from single images. This can be useful for a variety of applications, such as product design, architectural visualization, and special effects.
- Video games: G3DR can be used to generate 3D assets for video games, such as characters, weapons, and vehicles. This can help game developers to create more immersive and engaging video games.

Unfortunately, G3DR, or more powerful models based on it can be used negatively. One potential concern is that G3DR could be used to create realistic-looking fake images or videos (DeepFakes). This could be used for malicious purposes, such as spreading misinformation or propaganda. The authors do not endorse the usage of G3DR for any malicious purpose, such as deep fakes.

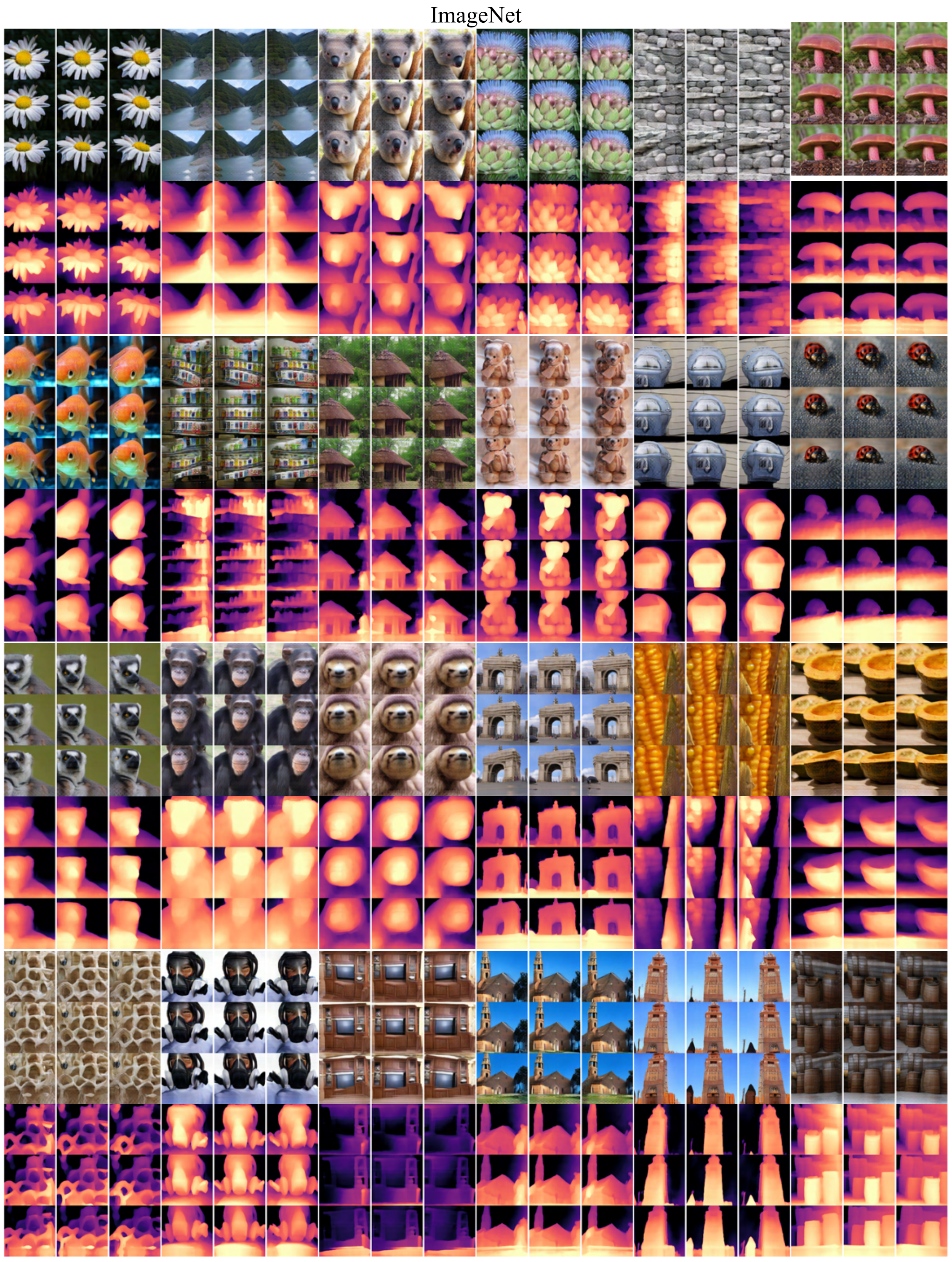

Figure S4. Visualization of results of the results provided along with the supplementary.

References

[1] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 2

[2] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster

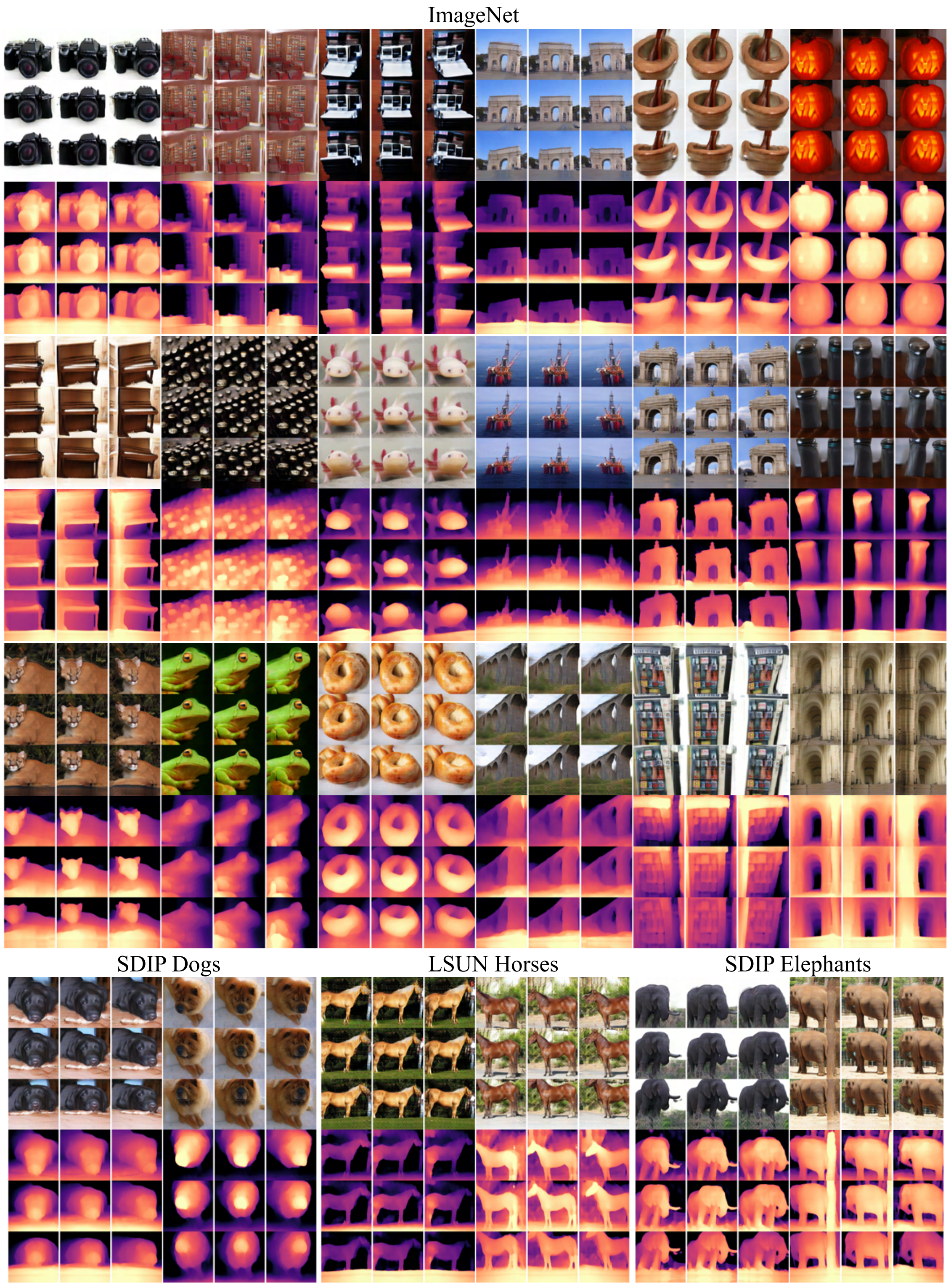

Figure S5. Visualization of results of the results provided along with the supplementary.

training for free. In *CVPR*, 2022. 1

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[4] Pradyumna Reddy, Ismail Elezi, and Jiankang Deng. G3dr: Generative 3d reconstruction in imagenet, 2024. 1

[5] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 2

[6] Kyle Sargent, Jing Yu Koh, Han Zhang, Huiwen Chang, Charles Herrmann, Pratul Srinivasan, Jiajun Wu, and Deqing Sun. VQ3D: learning a 3d-aware generative model on imagenet. In *ICCV*, 2023. 2, 3

[7] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2d stylegan for 3d-aware face generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6266, 2021. 2

[8] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. In *ICLR*, 2023. 2, 3

[9] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Müller, and Vasudev Lal. LDM3D: latent diffusion model for 3d. *CoRR*, abs/2305.10853, 2023. 1

[10] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3d-aware image generation using 2d diffusion models. In *ICCV*, 2023. 3