

GOEnFusion: Gradient Origin Encodings for 3D Forward Diffusion Models

Animesh Karnewar

UCL, MetaAI

a.karnewar@ucl.ac.uk

Andrea Vedaldi

MetaAI

vedaldi@meta.com

Niloy J. Mitra

UCL

n.mitra@ucl.ac.uk

David Novotny

MetaAI

dnovotny@meta.com

<https://holodiffusion.github.io/goenfusion>

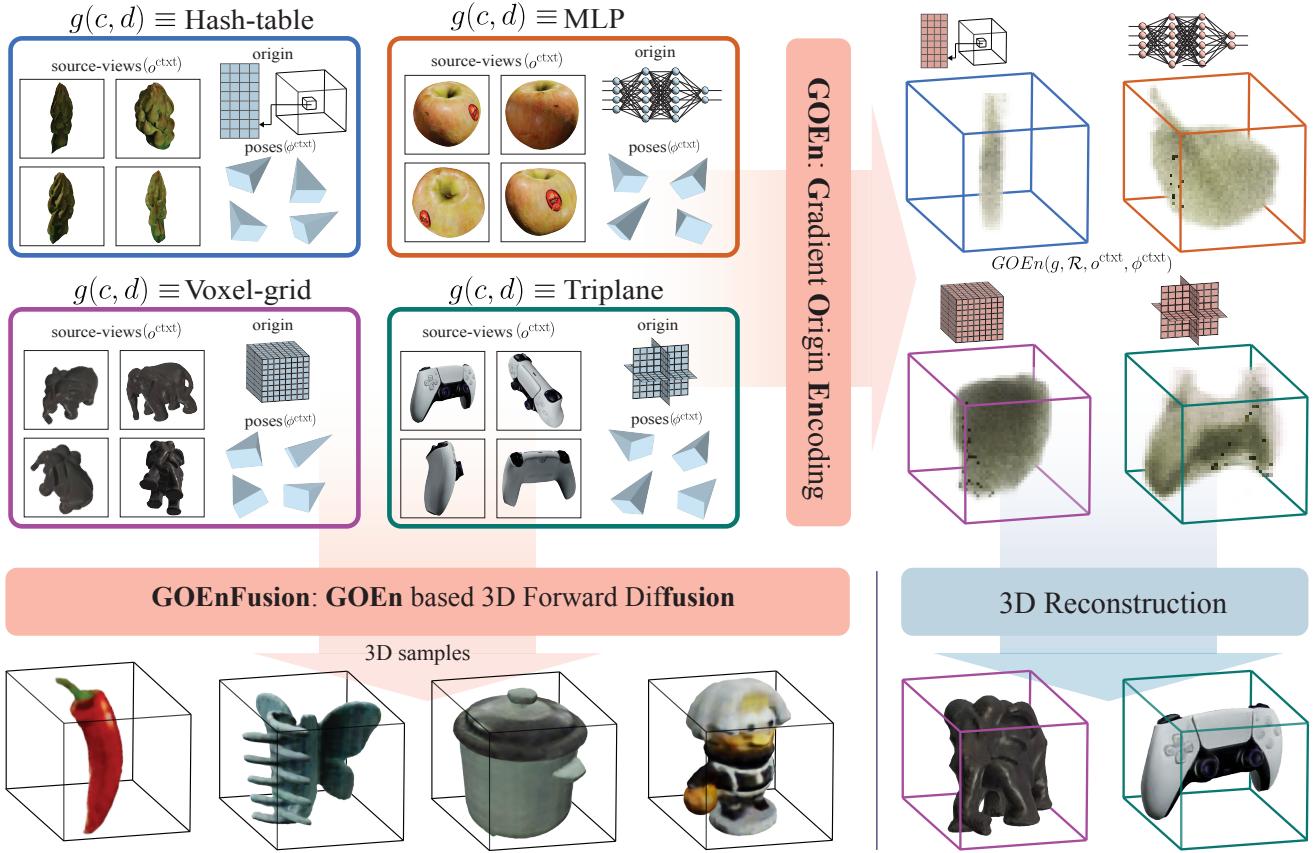


Figure 1. We propose the **GOEn** (Gradient Origin Encoding) that encodes source views (observations) into arbitrary 3D Radiance-Field representations $g(c, d)$ (sec. 3.1) while trying to maximize the transfer of source information. **GOEn** encodings can be used in the context of 3D forward diffusion models (our proposed **GOEnFusion**; sec. 3.2) and for regression-based 3D reconstruction (sec. 4.3).

Abstract

The recently introduced Forward-Diffusion method allows to train a 3D diffusion model using only 2D images for supervision. However, it does not easily generalise to different 3D representations and requires a computationally expensive auto-regressive sampling process to generate the underlying 3D scenes. In this paper, we propose **GOEn**: Gradient Origin Encoding (pronounced “gone”). **GOEn**

can encode input images into any type of 3D representation without the need to use a pre-trained image feature extractor. It can also handle single, multiple or no source view(s) alike, by design, and tries to maximise the information transfer from the views to the encodings. Our proposed **GOEnFusion** model pairs **GOEn** encodings with a realisation of the Forward-Diffusion model which addresses the limitations of the vanilla Forward-Diffusion realisation. We evaluate how much information the **GOEn** mech-

anism transfers to the encoded representations, and how well it captures the prior distribution over the underlying 3D scenes, through the lens of a partial AutoEncoder. Lastly, the efficacy of the GOEnFusion model is evaluated on the recently proposed OmniObject3D dataset while comparing to the state-of-the-art Forward and non-Forward-Diffusion models and other 3D generative models.

1. Introduction

Traditional creative applications of 3D computer graphics such as movies and video-games have grown so much that today these industries have a significant place in the world economy [46, 47]. Furthermore, various new applications such as Infotainment and Tele-Presence are on the rise due to the advancements in AR (augmented reality), VR (virtual reality), and MR (mixed reality). The key ingredient that drives all these applications is high quality 3D content. The latter is difficult to create, requiring complex equipment for scanning or hours of artistic efforts for modelling. Given how the state-of-the-art text-to-image generative models such as DALLE [32], StableDiffusion [33], Imagen[34], Giga-GAN [15], etc. and text-to-video generative models such as Imagen-video [14], Make-a-video [36], etc. have revolutionized the process of 2D content creation, we can expect that 3D generative modelling will also contribute significantly towards simplifying the process of 3D content creation.

However, there are three main challenges towards the goal of fully maturing 3D generative modelling. Firstly, there is a lack of high-quality large-scale 3D dataset; secondly, unlike 2D images, which are represented as *grids-of-pixels*, 3D assets do not have a *de-facto* digital representation; and, thirdly, transferring methods from 2D generative modelling to 3D such that they remain efficient is non-trivial. Towards the first challenge, recently, various data collection efforts [8, 9] are gathering millions of 3D textured-mesh based assets; while methods such as 3D distillation of large-scale text-to-image and text-to-video models [21, 31, 37, 40, 45, 54] and 3D reconstruction from multi-view-text-to-image generation models [20, 23, 24, 35] promise a bootstrapped approach to forming a large-scale dataset of 3D assets. Whereas, towards the second challenge, the *Neural-Fields* [50] attempts to provide a way towards unifying the disarray around 3D representations. On the contrary, solving the third one is still an open and ongoing problem. Hence, we focus our attention on this challenge and contribute substantial strides towards solving it, albeit on a small-scale 3D dataset.

The use of neural 3D representations makes the third challenge more challenging since it requires first fitting a corpus of 3D neural assets given the *raw-sensor-like* observations (images, depths, point-clouds, etc.). While re-

cent works [4, 5, 10, 16, 18, 19, 39] propose approaches to training 3D generative models using only 2D image supervision independently, the notable work, of Tewari et al., Forward-Diffusion models [41] proposes a unified mathematical framework for the *stochastic-inverse* setting of generative modelling where we only have access to partial observations of underlying ground-truth signals, but never the underlying signals themselves. However, their vanilla realisation of the proposed mathematical framework for 3D inverse-graphics (i.e. training 3D model using only 2D images) has a few limitations. Firstly, they cannot generate purely unconditional 3D samples since their realisation is a 2D view conditioned 3D forward diffusion model; secondly, they specifically make use of point-clouds of 3D features (via PixelNerf’s [52] CNN extracted image features) for the underlying 3D representation, while not generalising to other 3D neural representations trivially; and thirdly, their realisation requires a computationally expensive auto-regressive process for sampling the underlying 3D scenes.

To this end, we propose an encoding mechanism which can encode source views into arbitrary 3D representations without the need for pretrained image feature extraction models (fig. 1). Using the encodings, our proposed realisation of the mathematical framework of the 3D Forward-Diffusion models [41] overcomes the limitations of their vanilla realisation by providing a unified mathematical recipe for arbitrary 3D representations, eliminating the auto-regressive sampling procedure of the vanilla realisation, and allowing to generate unconditional and view-conditional 3D samples. In summary, our contributions are:

- We present **GOEn** as a unifying encoding for source views into different 3D representations (sec. 3.1, eq. 12).
- We present **GOEnFusion** as an improved realisation of the Forward-Diffusion framework (sec. 3.2 eqs. 14-17).
- We show that the **GOEn** is capable as a standalone mechanism for extracting different 3D representations from source images (sec. 4.1), and the **GOEnFusion** model can do 3D generation (sec. 4.2) and 3D reconstruction (sec. 4.3) robustly.

2. Preliminaries

2.1. Preliminaries: GONs (Gradient Origin Networks)

Given a dataset of observed samples $x \sim p_{\text{data}}$, where $x \in \mathbb{R}^m$, a Gradient Origin Network [2] (GON) auto-encodes the input x into reconstructed \hat{x} via a much smaller latent space $z \in \mathbb{R}^k$, where $k \ll m$, without requiring an explicit encoder network. The encoder mapping of the input x to latent z , $F_{\text{enc}} : \mathbb{R}^k \rightarrow \mathbb{R}^m$, is defined as the gradient of the log-likelihood of x under the decoder network $F_{\text{dec}} : \mathbb{R}^m \rightarrow \mathbb{R}^k$ wrt. a known fixed origin latent z_0 . In practice, the z_0 is always set to zeros, i.e., $z_0 \in \mathbb{R}_{[0]}^k$ and the mean squared

error between $F_{\text{dec}}(z_0)$ and x is used as an estimate of the log-likelihood as follows:

$$\begin{aligned} z &= F_{\text{enc}}(x) \\ &= -\nabla_{z_0} \|x - F_{\text{dec}}(z_0)\|_2^2. \end{aligned} \quad (1)$$

The decoding of z can now simply be obtained as a forward pass of the decoder network F_{dec} :

$$\begin{aligned} \hat{x} &= F_{\text{dec}}(z) \\ &= F_{\text{dec}}(-\nabla_{z_0} \|x - F_{\text{dec}}(z_0)\|_2^2). \end{aligned} \quad (2)$$

Using this encoder-less auto-encoding mechanism, the decoder network can be trained using the standard log-likelihood maximization objective (again estimated via mean squared error):

$$\begin{aligned} L^{\text{MSE}}(x, \hat{x}) &= \|x - \hat{x}\|_2^2 \\ &= \|x - F_{\text{dec}}(-\nabla_{z_0} \|x - F_{\text{dec}}(z_0)\|_2^2)\|_2^2. \end{aligned} \quad (3)$$

In the GON paper [2], Bond-Taylor and Willcocks also proposed a variational version of the GON autoencoder to enable generative sampling of the latent space, a coordinate-based implicit version of the GON as a secondary application, and a multi-step generalisation of the Gradient Origin Networks mechanism. But we only describe the auto-encoder version here, because it is the most relevant one to our proposed GOEn encodings (sec 3.1).

2.2. Preliminaries: Forward-Diffusion models

Diffusion models define the generative process through two markov chains, namely forward (diffusion) and reverse (denoising), over the observed data distribution p_{data} ¹. The most popular variant DDPM [13] defines the forward distribution $q(x_{0:T}) = q(x_0, x_1, \dots, x_t, \dots, x_T)$ as the markov chain of T Gaussian transitions:

$$q(x_{0:T}) = q(x_0) \prod_1^T q(x_t|x_{t-1})$$

where,

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I), \quad (4)$$

where the sequence $(\alpha_{1:T})$ instantiates a monotonically decreasing noise schedule such that the marginal over the last variable approximately equals a standard Gaussian distribution, i.e. $q(x_T|x_{T-1}) \approx q(x_T) \approx \mathcal{N}(0, I)$. Similar to the forward process, the reverse process is also defined as

¹please note that the forward diffusion process here is not to be confused with the forward setting of Forward-Diffusion models.

another markov chain of T Gaussian transitions:

$$p(x_{T:0}) = p(x_T) \prod_1^T p_\theta(x_{t-1}|x_t)$$

where,

$$p(x_0) = q(x_0) = p_{\text{data}}(x)$$

$$p(x_T) = q(x_T) = \mathcal{N}(0, I)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\sqrt{\alpha_{t-1}}\mathcal{D}_\theta(x_t, t), (1 - \alpha_{t-1})I), \quad (5)$$

where the mean of the Gaussian is given by a learned timestep-conditioned denoiser network $\mathcal{D}_\theta : \mathcal{R}^m \rightarrow \mathcal{R}^m$. The network \mathcal{D}_θ can be trained by minimizing the expectation of the KL divergence between [25] the forward and the reverse transition distributions of a given particular latent x_t over the timesteps t as

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_t[D_{KL}(q(x_t|x_{t-1})||p_\theta(x_t|x_{t+1}))], \quad (6)$$

and synthetic samples can be drawn from the trained model by iterating T -times over eq. 5 starting from samples x_T drawn from standard Gaussian distribution $\mathcal{N}(0, I)$. In practice, assuming that the network $D_\theta(x_t, t)$ outputs an estimate of the clean data sample $\hat{x}_0 \approx x_0$ instead of the mean of the Gaussian transition over the previous latent x_{t-1} , the KL divergence objective simplifies to the mean squared error between \hat{x}_0 and x_0 [25]

$$\mathcal{L}_{\text{simple}}(x_0, \hat{x}_0) = \mathbb{E}_t[\|x_0 - \mathcal{D}_\theta(x_t, t)\|_2^2]. \quad (7)$$

Forward-Diffusion models [41] consider the class of *stochastic-inverse* problems, like inverse 3D graphics, which pose a unique challenge where we do not have direct access to the samples $x \sim p_{\text{data}}$, but only have access to partial observations of x obtained through a differentiable forward function $o = \text{forward}(x, \phi)$, where ϕ are the parameters for obtaining the observation. The mathematical framework of Forward-Diffusion learns the conditional distribution $p(x|o, \phi)$ over the unobserved data x given the partial observations (o, ϕ) , by modelling the *push-forward* distribution $p_\theta(o^{\text{trgt}}|o^{\text{ctxt}}, \phi^{\text{ctxt}}, \phi^{\text{trgt}})$. The observations $(o^{\text{ctxt}}, \phi^{\text{ctxt}})$ form the context while $(o^{\text{trgt}}, \phi^{\text{trgt}})$ are the target observations.

$$\begin{aligned} p_\theta(o_{0:T}^{\text{trgt}}|o^{\text{ctxt}}, \phi^{\text{ctxt}}, \phi^{\text{trgt}}) &= \\ p(o_T^{\text{trgt}}) \prod_1^T p_\theta(o_{t-1}^{\text{trgt}}|o_t^{\text{trgt}}, o^{\text{ctxt}}, \phi^{\text{ctxt}}, \phi^{\text{trgt}}) \end{aligned} \quad (8)$$

In order to implicitly model the conditional distribution over the data x from the pushforward, each of the learned reverse transition $p_\theta(o_{t-1}^{\text{trgt}}|o_t^{\text{trgt}}, o^{\text{ctxt}}, \phi^{\text{ctxt}}, \phi^{\text{trgt}})$ is defined using the following three steps:

$$x_t = \text{denoise}(o^{\text{ctxt}}, o_t^{\text{trgt}}, t, \phi^{\text{ctxt}}, \phi^{\text{trgt}}) \quad (9)$$

$$\hat{o}_t^{\text{trgt}} = \text{forward}(x_t, \phi^{\text{trgt}}) \quad (10)$$

$$o_{t-1}^{\text{trgt}} \sim \mathcal{N}(\sqrt{\alpha_{t-1}}\hat{o}_t^{\text{trgt}}, (1 - \alpha_{t-1})I). \quad (11)$$

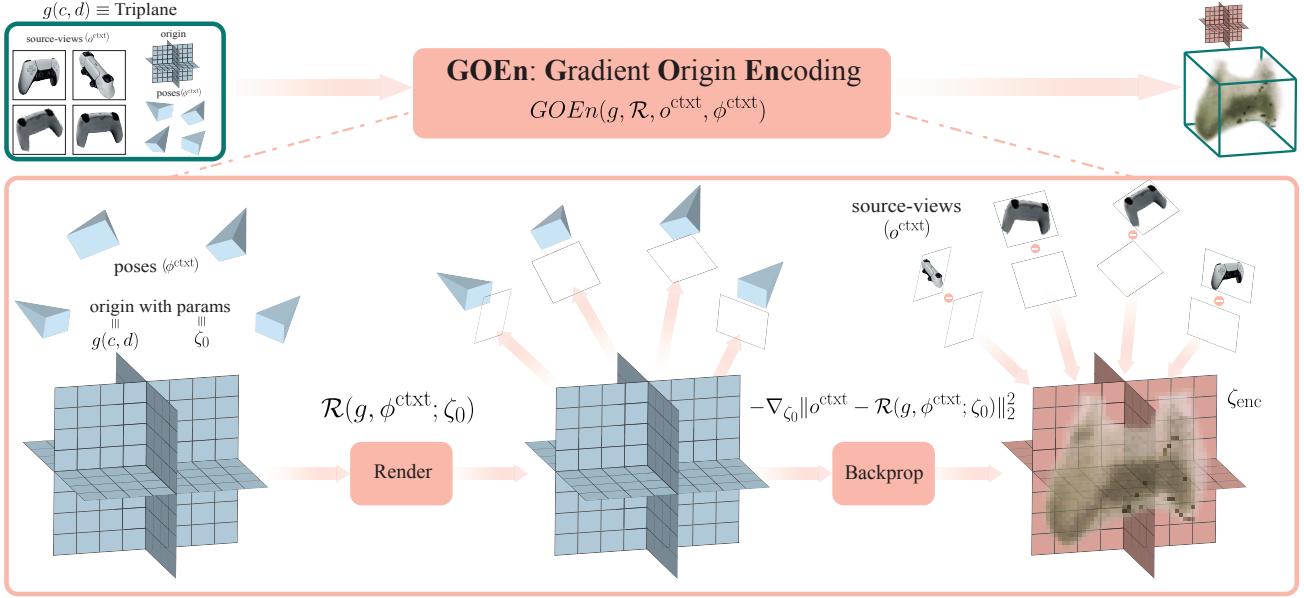


Figure 2. **Didactic illustration of the GOEn mechanism.** We demonstrate the mechanism here using the Triplane representation for $g(c, d)$, but note that this can be applied to other representations as well. The GOEn mechanism (eq. 12) consists of two steps. First we render the origin ζ_0 from the context-poses ϕ^{ctxt} into almost blank renders. Then, we compute the gradient of the MSE between the renders and the source-views o^{ctxt} wrt. the origin ζ_0 which gives us the GOEn encoded version ζ_{enc} .

Equations 9 and 10 define the same functionality as that of the \mathcal{D}_θ denoiser network, but also integrate the differentiable forward function in the process. The noisy versions of the observed target views o_t^{trgt} can be obtained by the straightforward diffusion process as defined in eq. 4 without any changes. The Forward-Diffusion model can be trained using a conditional version of the KL divergence of eq. 6 as follows:

$$\begin{aligned} \mathcal{L}_{\text{forward-diffusion}} \\ = \mathbb{E}_t[D_{KL}(q(o_t^{\text{trgt}}|o_{t-1}^{\text{trgt}}) || p_\theta(o_t^{\text{trgt}}|o_{t+1}^{\text{trgt}}, o^{\text{ctxt}}, \phi^{\text{ctxt}}), \phi^{\text{trgt}})]. \end{aligned}$$

And lastly, synthetic samples of the unobserved underlying variable x can be generated in a auto-regressive manner. First we draw a sample o^{trgt} starting from a given set of $(o^{\text{ctxt}}, \phi^{\text{ctxt}})$ context observations by iterating over equations 9, 10, and 11 T times. A partial estimate of the unobserved underlying variable x_0^0 is obtained from the last denoising step. The complete sample x can be generated by merging n different partial estimates $x_0^{0:n-1}$ by repeating the former process n -times and accumulating the drawn o^{trgt} 's together as a new context each time.

3. Method

3.1. GOEn (Gradient Origin Encodings)

We proceed with explaining the proposed GOEn encodings in the context of inverse 3D graphics, but note that they are

generally applicable. Let $g(c, d)$ represent a static 3D scene as a Radiance-Field such that $c = [x, y, z]$ ² denotes the 3D coordinates of a point in the euclidean space, $d = [\theta, \gamma]$ ³ denotes the spherical polar coordinates of an outgoing direction from the point, and the function $g : \mathbb{R}^5 \rightarrow \mathbb{R}^4$ maps each 3D coordinate and a particular outgoing direction to four values (σ, R, G, B) . Here, σ represents density and $[R, G, B]$ represents the outgoing radiance. Let $\mathcal{R}(g, \eta)$ denote the rendering functional which converts the Radiance-Field function into an image of a certain resolution from a certain camera viewpoint as described by η camera parameters. We use the differentiable Emission-Absorption Volume Raymarching algorithm for rendering [26, 27].

For building a generalisable mathematical recipe for the denoise operation of the Forward-Diffusion model (eq. 9), a mechanism which can encode the observations into a partial estimate of the unobserved underlying data sample, x , must have the following three properties:

1. should be able to encode zero or more observations alike;
2. should generalise to any parameterisation/realisation of the function g . In case of inverse 3D graphics, the encoding mechanism should be applicable irrespective of whether the Radiance-Field is represented as an MLP

²Please note that the x coordinate here is not to be confused with the x samples drawn from the p_{data} .

³We deviate from the more common use of $[\theta, \phi]$ for spherical polar coordinates in favour of $[\theta, \gamma]$ to avoid confusion with ϕ^{ctxt} and ϕ^{trgt} used in the notations of the Forward-Diffusion framework.

- [27], a hash-table [29], a triplane [6, 43], or a voxel grid [17, 38, 51], etc.;
3. and, should try to maximise the information transfer from the observations to the encoding of the underlying data sample.

We propose Gradient Origin Encodings (GOEns) for this purpose, where we define the encodings of the observations as the gradient of the log-likelihood of the observations under the differentiable forward operation of eq. 10. Without loss of generality, assuming that ζ are the parameters of the g function (i.e. the features/weights of the 3D Radiance-Field) and ζ_0 denotes the origin (zero parameters), we define the encodings ζ_{enc} as (fig. 2):

$$\begin{aligned}\zeta_{\text{enc}} &:= \text{GOEn}(g, \mathcal{R}, o^{\text{ctxt}}, \phi^{\text{ctxt}}) \\ &:= -\nabla_{\zeta_0} \|o^{\text{ctxt}} - \mathcal{R}(g, \phi^{\text{ctxt}}; \zeta_0)\|_2^2.\end{aligned}\quad (12)$$

Please note here that we replace the camera parameters η by ϕ^{ctxt} to make the notation consistent with the notations of the Forward-Diffusion models, and as done usually, we estimate the log-likelihood by the mean squared error. Note that the encoding function GOEn backpropagates through the differentiable forward functional \mathcal{R} . As apparent, these encodings satisfy the formerly stated properties 1. and 2. by design, while we minimise the following loss function similar to eq. 3:

$$\begin{aligned}\mathcal{L}^{\text{GOEn-MSE}}(o^{\text{ctxt}}, \hat{o}^{\text{ctxt}}) &:= \|o^{\text{ctxt}} - \hat{o}^{\text{ctxt}}\|_2^2 \\ &:= \|o^{\text{ctxt}} - \mathcal{R}(g, \phi^{\text{ctxt}}; \text{GOEn}(g, \mathcal{R}, o^{\text{ctxt}}, \phi^{\text{ctxt}}))\|_2^2,\end{aligned}\quad (13)$$

for maximising the information content in the encodings ζ_{enc} to satisfy property 3. We essentially re-purpose the backward pass of the rendering function to encode the information in the source views o^{ctxt} into the parameters of the 3D scene representation ζ .

As apparent, we take inspiration from GONs, but our GOEn encodings are different from them in that: (1.) while the purpose of the GON model is to obtain a compressed latent space, our GOEn mechanism is aimed at obtaining a coarse partial estimate of the plenoptic 3D scene given image observations; and (2.), our GOEn encodings are much more local in nature compared to the latent encodings obtained using the GON.

3.2. GOEnFusion

Similar to the previous sub-section, the discussion here is in the context of 3D inverse graphics, but we note that the proposed method is generally applicable. Equipped with the ingredient of the GOEn mechanism, we now present the generalisable mathematical recipe for the Forward-Diffusion models. Our unobserved x samples here correspond to the ζ parameters since we are interested in modelling the generative probability distribution over the 3D

Radiance-Fields, i.e. $x_t = \zeta_t$. We propose to generalise the equations 9, 10, and 11 into the following steps:

$$x_{\text{enc}} := \text{GOEn}(g, \mathcal{R}, o^{\text{ctxt}}, \phi^{\text{ctxt}}) \quad (14)$$

$$\hat{x}_0 := \mathcal{D}_{\theta}(x_T, T; x_{\text{enc}}) \quad (15)$$

$$\hat{x}_t \sim q(\hat{x}_t | \hat{x}_{t-1}) = q(\hat{x}_0) \prod_1^t q(\hat{x}_t | \hat{x}_{t-1}) \quad (16)$$

$$\hat{o}^{\text{trgt}} := \mathcal{R}(g, \phi^{\text{trgt}}; \mathcal{D}_{\theta}(\hat{x}_t, t; x_{\text{enc}})), \quad (17)$$

where T corresponds to the highest timestep and $x_T \sim \mathcal{N}(0, I)$ is a sample of the standard Gaussian noise distribution. Equation 14 first encodes the context observations and their parameters ($o^{\text{ctxt}}, \phi^{\text{ctxt}}$) into the GOEn encoding x_{enc} using eq. 12. Then eq. 15 uses the denoiser network to predict a pseudo-deterministic estimate of the clean 3D scene \hat{x}_0 conditioned on the GOEn encodings x_{enc} . The input to the network here contains a concatenation of pure Gaussian noise x_T and the GOEn encodings x_{enc} . While the noise can impart some degree of stochasticity to the output, the network can, in theory, completely ignore the noise. We let the network learn what to do based on the training and the data complexity, and hence call this step “pseudo”-deterministic. This is followed by obtaining a noisy version of the 3D scene \hat{x}_t through standard forward Gaussian corruption (eq. 16). And finally, the prediction for target observations can be obtained by rendering the output of the denoiser network \mathcal{D}_{θ} for \hat{x}_t conditioned on the GOEn encodings x_{enc} and the timestep t using eq. 17.

We require three different loss functions to be able to train this framework end-to-end such that each component does what it is supposed to do

$$\begin{aligned}\mathcal{L}^{\text{GOEnFusion}} &:= \mathbb{E}_{t, o^{\text{trgt}}} \|o^{\text{trgt}} - \hat{o}^{\text{trgt}}\|_2^2 \\ \mathcal{L}^{\text{PSE-DET}} &:= \mathbb{E}_{t, o^{\text{trgt}}} \|o^{\text{trgt}} - \mathcal{R}(g, \phi^{\text{trgt}}; \mathcal{D}_{\theta}(x_T, T; x_{\text{enc}}))\|_2^2 \\ \mathcal{L}^{\text{GOEn}} &:= \mathbb{E}_{t, o^{\text{ctxt}}} \|o^{\text{ctxt}} - \mathcal{R}(g, \phi^{\text{ctxt}}; x_{\text{enc}})\|_2^2.\end{aligned}$$

The final objective is simply the sum of all the three loss functions

$$\mathcal{L}^{\text{total}} := \mathcal{L}^{\text{GOEnFusion}} + \mathcal{L}^{\text{PSE-DET}} + \mathcal{L}^{\text{GOEn}}.$$

The loss function $\mathcal{L}^{\text{GOEn}}$ is required to maximise the information content in the GOEn encodings (property 3.), while the $\mathcal{L}^{\text{PSE-DET}}$ tries to maximise the reconstruction quality of the pseudo-deterministic output \hat{x}_0 , and the $\mathcal{L}^{\text{GOEnFusion}}$ loss function actually trains the revised Forward-Diffusion model. As apparent, the GOEn mechanism theoretically enables the GOEnFusion model to use any parameterisation/realisation of the g function as long as it allows differentiable rendering and can define a zero-origin over its parameters ζ . Please refer to the supplementary material for a diagrammatic illustration of the GOEnFusion model. In practice, we train the model using the

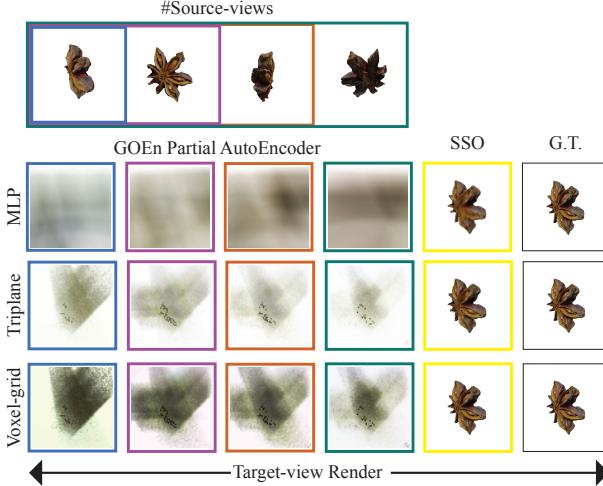


Figure 3. **Qualitative evaluation of Partial-AutoEncoding.** The rows MLP, Triplane and Voxel-grid show the renders of the GOEn encoded representations from the target-view respectively. The colour-coded columns demonstrate the effect of varying the number-of-source views (1, 2, 3, 4) used in the GOEn encoding. The SSO column shows the target render of the single-scene-overfitted representation while the G.T. column shows the mesh-render from the dataset (repeated for clarity).

classifier free guidance training scheme [12], where we dropout the GOEn conditioning randomly with a probability of 0.5, to allow for both conditional and unconditional sampling.

Lastly, samples can be generated by doing the iterative denoising using the trained model \mathcal{D}_θ either unconditionally or conditionally using the GOEn encodings of certain context observations. Note that, here we are able to directly sample (denoise) in the space of the unobserved underlying data samples x , and hence eliminate the need for the expensive auto-regressive fusion required by the vanilla Forward-Diffusion models [41].

4. Experiments

We evaluate the GOEn mechanism along 3 axes: firstly, in order to measure how much information the GOEn mechanism transfers from the source views (observations) to the encodings, we run experiments in a partial autoencoder setting (sec. 4.1); secondly, we train the GOEnFusion model to learn a 3D generative model using 2D images to evaluate its 3D generative capability (sec. 4.2); and finally, we also evaluate the GOEn mechanism in a 3D reconstruction setting (sec. 4.3).

Dataset We perform all our experiments on the recently released OmniObject3D dataset [49] which contains $\sim 6K$ 3D scans of real world objects from daily life. The dataset

contains a large-vocabulary of ~ 200 categories of daily life classes having some overlap with COCO [22]. For our Non-Forward baseline, we also use the text-captions recently released by the OmniObject3d authors at their GitHub page [48].

Metrics For the experiments done in the partial-autoencoding setting (sec. 4.1), we use the standard image reconstruction metrics **PSNR**, **LPIPS** [53] and **SSIM** [44], while for the quantitative analysis in generative modelling experiments (sec. 4.2) we use the **FID** [11] and **KID** [1] metrics following all the prior works. And, lastly for the 3D reconstruction experiments (sec. 4.3) we again use the standard image reconstruction metrics similar to the partial-autoencoding experiments. We colour code all the metric scores as first, second, and third.

4.1. Partial AutoEncoding

As described in section 3.1, in order to evaluate the information transfer from the source views to the encoded 3D representation, we train the standalone GOEn component on its own before using it in the context of 3D generation and 3D reconstruction. Given the dataset $D = \{(\mathcal{I}_i^j, \phi_i^j) | i \in [0, N] \text{ and } j \in [0, C]\}$ of N 3D scenes where each scene contains C images and camera parameters, we define the partial-autoencoder as a mechanism which encodes k source views and camera parameters, of a certain 3D scene, into the representation g (whose parameters are ζ). The encoded scene representation should be such that the rendered views from the same source cameras, should be as close as possible to the G.T. images \mathcal{I} , i.e., the partial-autoencoder $PA : \mathbb{R}^{h \times w \times c} \times \mathbb{R}^{4 \times 4} \rightarrow \mathbb{R}^{h \times w \times c}$ should minimize the following mean squared error objective:

$$\begin{aligned} \mathcal{L}^{\text{PA-MSE}} &:= \mathbb{E}_{(\mathcal{I}, \phi) \sim D} \|\mathcal{I} - PA(\mathcal{I}, \phi)\|_2^2 \\ &\text{where,} \\ PA(\mathcal{I}, \phi) &:= \mathcal{R}(g, \phi; GOEn(g, \mathcal{R}, \mathcal{I}, \phi)). \end{aligned} \quad (18)$$

Since the auto-encoder is not targeted to do the 3D reconstruction, we denote this setting by the term “*partial-autoencoding*” instead of autoencoding. We evaluate the GOEn encodings on three different 3D Radiance-Field representations, g , namely Triplanes [6], Feature-Voxel grids [17, 19, 38, 51], and MLPs [27]. Please refer to the supplementary material for the implementation details of these experiments.

We compare the standalone GOEn trained on randomly chosen 100 scenes on 1, 2, 3, and 4, number of source views. Firstly, we note that the scores are for *target-views*, and not the source-views, since we are here interested in measuring the 3D encoding ability of the GOEn mechanism, as opposed to overfitting the source views into the representation. Secondly, note that the scores are computed



Figure 4. **Qualitative evaluation of 3D Generation.** 3D samples generated by our GOEnFusion compared to the 3D GAN baselines EG3D [5], and GET3D [10]; the non-Forward-Diffusion baselines DiffRF [28], DiffTF [3], and our non-Forward-Diffusion baseline; and with the main baseline of the vanilla Forward-Diffusion model [41]. We ran the vanilla Forward-Diffusion model baseline on OmniObject3D using their released code, while used the samples for rest of the baselines from DiffTF [3].

Table 1. **Quantitative evaluation of Partial AutoEncoding.** PSNR(\uparrow), LPIPS(\downarrow) and SSIM(\uparrow) of GOEn on three different realisations of the 3D Radiance-Field g . All the metrics are evaluated for target views (different from the source views) against the G.T. mesh renders from the dataset. The Single-Scene-Overfitting (SSO) scores denote the case of individually fitting the representations to the 3D scenes.

#source views	Triplane						Voxel-Grid						MLP					
	GOEn			SSO			GOEn			SSO			GOEn			SSO		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
1	15.711	1.025	0.477				15.514	1.008	0.479				11.208	1.124	0.496			
2	15.513	1.068	0.468				15.319	1.067	0.480				11.406	1.197	0.489			
3	15.672	1.129	0.456	28.165	0.087	0.941	15.590	1.110	0.484	32.061	0.067	0.9582	11.490	1.202	0.500	27.382	0.119	0.918
4	15.919	1.150	0.472				15.755	1.145	0.486				11.599	1.195	0.505			
Average	15.703	1.093	0.468				15.544	1.082	0.482				11.425	1.179	0.497			

against the renders of the G.T. meshes, but since some quality can be lost by the choice of the representation itself, a more grounded comparison for the partial-autoencoder is the Single-Scene-Overfitting (SSO) version, where we fit the representation directly to the scene. As shown in table 1

and figure 3, the GOEn mechanism is able to obtain better 3D coarse estimate as the number of source views increases. Although, we find that different 3D representations have varying levels of encoding capabiliies with the GOEn mechanism, interestingly, we note that almost 50% of the practi-

Table 2. **3D Generation Quantitative Evaluation.** FID(\downarrow) and KID(\downarrow) scores on the OmniObject3D dataset comparing our GOEnFusion with the GAN baselines EG3D [6], and GET3D [10]; with the non-forward diffusion baselines DiffRF[28], DiffTF[3], and Our non-forward diffusion baseline; and, with the Forward-Diffusion model [41].

Method	FID	KID
EG3D [6]	41.56	1.0
GET3D [10]	49.41	1.5
DiffRF [28]	147.59	8.8
DiffTF [3]	25.36	0.8
Forward-Diffusion [41]	73.51	3.8
Ours non-forward	119.67	8.0
GOEnFusion (Ours)	61.37	2.0

cal limit of the 3D reconstruction performance is obtained in a single gradient-descent step of the GOEn encodings.

Table 3. **3D reconstruction Quantitative Evaluation.** PSNR(\uparrow), LPIPS(\downarrow) and SSIM(\downarrow) of GOEnFusion’s pseudo-deterministic 3D reconstruction output. we again include the Single-Scene-Overfitting (SSO) here for comparison.

Method	GOEn			SSO		
	PSNR	LPIPS	SSIM	PSNR	LPIPS	SSIM
GOEn Recon (Unet)	31.430	0.083	0.931			
GOEn Recon (DiT)	27.650	0.109	0.900	28.165	0.087	0.941
GOEnFusion (Ours)	26.447	0.119	0.890			

4.2. 3D Generation

Although our proposed GOEn mechanism is applicable to arbitrary 3D representations, the 3D representations to use in diffusion models are limited by their amenability to the denoising networks architectures. Also because 3D diffusion models are computationally expensive to train, we only use the Triplane [6, 43] representation in this setting, because it is very efficient and perhaps the most compatible with diffusion network architectures. We use the DiT [30] network as our diffusion-denoising backbone with no changes. Please refer to the supplementary material for more training and implementation details.

As shown in table 2, our GOEnFusion achieves better FID and KID scores than the main baseline of the vanilla Forward-Diffusion models. We note again that our improvements are not only limited to the quality of generation, but also to the benefits provided by our GOEnFusion formulation. Since, DiffTF [3] uses many architectural modifications and other tricks specific to the OmniObject3D dataset, we also train a non-forward diffusion baseline in our setup

for a fair comparison, where our single-stage GOEnFusion training approach does better quantitatively.

Although DiffTF achieves better quantitative scores than ours, qualitatively, as shown in fig. 4, their samples suffer from various artifacts such as checkerboard patterns on the surfaces and high-frequency noise. Compared to the Forward-Diffusion samples, our GOEnFusion samples have a lot more details, while are much cleaner and artifact-free when compared to our non-Forward-Diffusion samples.

4.3. 3D Reconstruction

Although the task of source-view conditioned 3D scene reconstruction has moved to the generative setting [7, 41], we still evaluate the regression based deterministic 3D reconstruction capabilities of our proposed GOEn mechanism as an ablation. Specifically, we run the proposed GOEnFusion model only till the pseudo-deterministic output as described in eq. 15 to compute the target-view reconstruction metrics. Table 3 summarises the results for this ablation. We compare the reconstruction performance of GOEnFusion against the trained GOEn regression based 3D reconstruction baselines. We use two different Triplane models, one using a standard Unet and the other using the same DiT backbone. Please refer to the supplementary material for qualitative samples and more details of these experiments.

5. Conclusion

We presented the GOEn encodings as a general-purpose mechanism for encoding the information in 2D images into arbitrary 3D scene representations and evaluated its information transfer ability with the partial-autoencoding experiments. We show that the encodings can apply to Triplanes, Voxel-grids and MLPs, but note that exploring these in the context of other popular 3D representations forms scope for future work. The GOEn encodings find a practical use in the context of improving the framework of 3D-Forward-Diffusion models, which we demonstrate through 3D generation experiments on the OmniObject3D dataset.

Although our framework applies theoretically to any arbitrary 3D Radiance-Field representation, its use in the 3D generative modelling is restricted by the representation’s compatibility with existing denoiser network architectures. We believe that Transformers [42] get close being universally applicable, but it still remains a challenge to adapt certain 3D representations such as Hash-tables [29] as input to Transformers. Nevertheless, we believe that the proposed GOEn mechanism can be applicable to various other stochastic inverse problems in diverse fields, and that it will inspire further interesting applications and theoretical insights.

6. Acknowledgements

Animesh and Niloy were partially funded by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956585. This research has been partly supported by MetaAI and the UCL AI Centre. Finally, Animesh would like to thank **Roman Shapovalov** for the insightful discussions and help with triplane fitting code.

References

- [1] Mikolaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. [6](#)
- [2] Sam Bond-Taylor and Chris G Willcocks. Gradient origin networks. *arXiv preprint arXiv:2007.02798*, 2020. [2, 3](#)
- [3] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Large-vocabulary 3d diffusion model with transformer. *arXiv preprint arXiv:2309.07920*, 2023. [7, 8](#)
- [4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. [2](#)
- [5] 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, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. [2, 7](#)
- [6] 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 *Proc. CVPR*, 2022. [5, 6, 8](#)
- [7] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. [8](#)
- [8] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. [2](#)
- [9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. [2](#)
- [10] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. [2, 7, 8](#)
- [11] 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. [6](#)
- [12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. [6](#)
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020. [3](#)
- [14] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Grigchenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. [2](#)
- [15] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023. [2](#)
- [16] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. 3inGAN: Learning a 3D generative model from images of a self-similar scene. In *Proc. 3D Vision (3DV)*, 2022. [2](#)
- [17] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy J. Mitra. ReLU fields: The little non-linearity that could. *arXiv.cs/abs/2205.10824*, 2022. [5, 6](#)
- [18] Animesh Karnewar, Niloy J Mitra, Andrea Vedaldi, and David Novotny. Holofusion: Towards photo-realistic 3d generative modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22976–22985, 2023. [2](#)
- [19] Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy Mitra. Holodiffusion: Training a 3D diffusion model using 2D images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023. [2, 6](#)
- [20] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. <https://arxiv.org/abs/2311.06214>, 2023. [2](#)
- [21] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweet-dreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596*, 2023. [2](#)
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. [6](#)
- [23] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. [2](#)
- [24] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang,

- Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 2
- [25] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 3
- [26] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 4
- [27] 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 *Proc. ECCV*, 2020. 4, 5, 6
- [28] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffirf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022. 7, 8
- [29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *Proc. SIGGRAPH*, 2022. 5, 8
- [30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 8
- [31] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2
- [32] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proc. ICML*, 2021. 2
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2
- [34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv.cs/abs/2205.11487*, 2022. 2
- [35] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2
- [36] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2
- [37] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023. 2
- [38] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. CVPR*, 2022. 5, 6
- [39] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. *arXiv preprint arXiv:2306.07881*, 2023. 2
- [40] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. <https://github.com/ashawkey/stable-dreamfusion>. 2
- [41] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B Tenenbaum, Frédéric Durand, William T Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *arXiv preprint arXiv:2306.11719*, 2023. 2, 3, 6, 7, 8
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 8
- [43] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022. 5, 8
- [44] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, pages 1398–1402. Ieee, 2003. 6
- [45] 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. 2
- [46] Wikipedia. Film industry — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Film%20industry&oldid=1185181785>, 2023. [Online; accessed 16-November-2023]. 2
- [47] Wikipedia. Video game industry — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Video%20game%20industry&oldid=1184994563>, 2023. [Online; accessed 16-November-2023]. 2
- [48] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d github code and dataset, 2023. <https://github.com/omniobject3d/OmniObject3D>. 6
- [49] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 6
- [50] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 2
- [51] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinzhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxtels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021. 5, 6

- [52] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Proc. CVPR*, 2021. [2](#)
- [53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. [6](#)
- [54] Minda Zhao, Chaoyi Zhao, Xinyue Liang, Lincheng Li, Zeng Zhao, Zhipeng Hu, Changjie Fan, and Xin Yu. Efficientdreamer: High-fidelity and robust 3d creation via orthogonal-view diffusion prior. *arXiv preprint arXiv:2308.13223*, 2023. [2](#)