

# EscherNet: A Generative Model for Scalable View Synthesis

Xin Kong<sup>1\*</sup> Shikun Liu<sup>1\*</sup> Xiaoyang Lyu<sup>2</sup> Marwan Taher<sup>1</sup>  
Xiaojuan Qi<sup>2</sup> Andrew J. Davison<sup>1</sup>

<sup>1</sup>Dyson Robotics Lab, Imperial College London      <sup>2</sup>The University of Hong Kong

\*Corresponding Authors: {x.kong21, shikun.liu17}@imperial.ac.uk

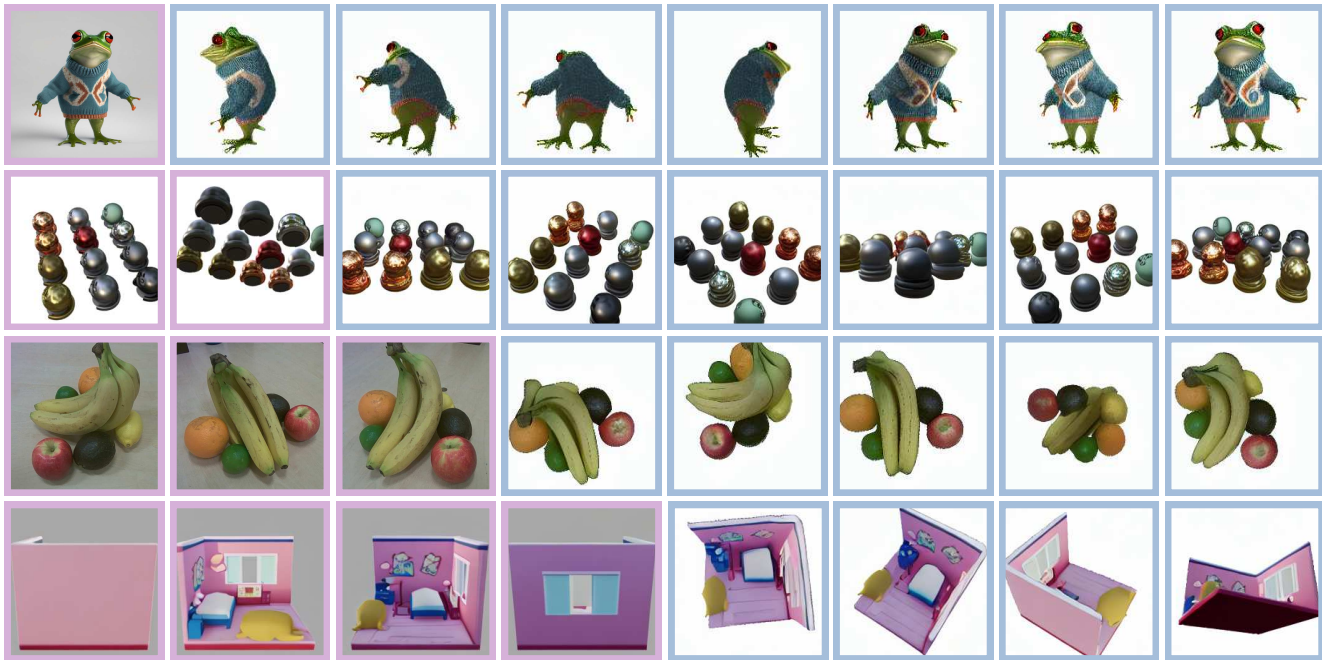


Figure 1. We introduce EscherNet, a diffusion model that can generate a flexible number of consistent target views (highlighted in blue) with arbitrary camera poses, based on a flexible number of reference views (highlighted in purple). EscherNet demonstrates remarkable precision in camera control and robust generalisation across synthetic and real-world images featuring multiple objects and rich textures.

## Abstract

We introduce EscherNet, a multi-view conditioned diffusion model for view synthesis. EscherNet learns implicit and generative 3D representations coupled with a specialised camera positional encoding, allowing precise and continuous relative control of the camera transformation between an arbitrary number of reference and target views. EscherNet offers exceptional generality, flexibility, and scalability in view synthesis — it can generate more than 100 consistent target views simultaneously on a single consumer-grade GPU, despite being trained with a fixed number of 3 reference views to 3 target views. As a result, EscherNet not only addresses zero-shot novel view synthesis, but also naturally unifies single- and multi-image 3D

reconstruction, combining these diverse tasks into a single, cohesive framework. Our extensive experiments demonstrate that EscherNet achieves state-of-the-art performance in multiple benchmarks, even when compared to methods specifically tailored for each individual problem. This remarkable versatility opens up new directions for designing scalable neural architectures for 3D vision. Project page: <https://kxhit.github.io/EscherNet>.

## 1. Introduction

View synthesis stands as a fundamental task in computer vision and computer graphics. By allowing the re-rendering of a scene from arbitrary viewpoints based on a set of reference viewpoints, this mimics the adaptability observed in

human vision. This ability is not only crucial for practical everyday tasks like object manipulation and navigation, but also plays a pivotal role in fostering human creativity, enabling us to envision and craft objects with depth, perspective, and a sense of immersion.

In this paper, we revisit the problem of view synthesis and ask: *How can we learn a general 3D representation to facilitate scalable view synthesis?* We attempt to investigate this question from the following two observations:

i) Up until now, recent advances in view synthesis have predominantly focused on training speed and/or rendering efficiency [12, 18, 30, 47]. Notably, these advancements all share a common reliance on volumetric rendering for scene optimisation. Thus, all these view synthesis methods are inherently *scene-specific*, coupled with global 3D spatial coordinates. In contrast, we advocate for a paradigm shift where a 3D representation relies solely on scene colours and geometries, learning implicit representations without the need for ground-truth 3D geometry, while also maintaining independence from any specific coordinate system. This distinction is crucial for achieving scalability to overcome the constraints imposed by scene-specific encoding.

ii) View synthesis, by nature, is more suitable to be cast as a *conditional generative modelling problem*, similar to generative image in-painting [25, 59]. When given only a sparse set of reference views, a desired model should provide multiple plausible predictions, leveraging the inherent stochasticity within the generative formulation and drawing insights from natural image statistics and semantic priors learned from other images and objects. As the available information increases, the generated scene becomes more constrained, gradually converging closer to the ground-truth representation. Notably, existing 3D generative models currently only support a single reference view [20–23, 43]. We argue that a more desirable generative formulation should flexibly accommodate varying levels of input information.

Building upon these insights, we introduce EscherNet, an image-to-image conditional diffusion model for view synthesis. EscherNet leverages a transformer architecture [50], employing dot-product self-attention to capture the intricate relation between both reference-to-target and target-to-target views consistencies. A key innovation within EscherNet is the design of camera positional encoding (CaPE), dedicated to representing both 4 DoF (object-centric) and 6 DoF camera poses. This encoding incorporates spatial structures into the tokens, enabling the model to compute self-attention between query and key solely based on their relative camera transformation. In summary, EscherNet exhibits these remarkable characteristics:

- **Consistency:** EscherNet inherently integrates view consistency thanks to the design of camera positional encoding, encouraging both *reference-to-target* and *target-to-target view consistencies*.

- **Scalability:** Unlike many existing neural rendering methods that are constrained by scene-specific optimisation, EscherNet decouples itself from any specific coordinate system and the need for ground-truth 3D geometry, without any expensive 3D operations (*e.g.* 3D convolutions or volumetric rendering), making it easier to *scale with everyday posed 2D image data*.
- **Generalisation:** Despite being trained on only a fixed number of 3 reference to 3 target views, EscherNet exhibits the capability to generate *any number of target views, with any camera poses, based on any number of reference views*. Notably, EscherNet exhibits improved generation quality with an increased number of reference views, aligning seamlessly with our original design goal.

We conduct a comprehensive evaluation across both novel view synthesis and single/multi-image 3D reconstruction benchmarks. Our findings demonstrate that EscherNet not only outperforms all 3D diffusion models in terms of generation quality but also can generate plausible view synthesis given very limited views. This stands in contrast to these scene-specific neural rendering methods such as InstantNGP [30] and Gaussian Splatting [18], which often struggle to generate meaningful content under such constraints. This underscores the effectiveness of our method’s simple yet scalable design, offering a promising avenue for advancing view synthesis and 3D vision as a whole.

## 2. Related Work

**Neural 3D Representations** Early works in neural 3D representation learning focused on directly optimising on 3D data, using representations such as voxels [26] and point clouds [39, 40], for explicit 3D representation learning. Alternatively, another line of works focused on training neural networks to map 3D spatial coordinates to signed distance functions [34] or occupancies [28, 36], for implicit 3D representation learning. However, all these methods heavily rely on ground-truth 3D geometry, limiting their applicability to small-scale synthetic 3D data [2, 54].

To accommodate a broader range of data sources, differentiable rendering functions [32, 45] have been introduced to optimise neural implicit shape representations with multi-view posed images. More recently, NeRF [29] paved the way to a significant enhancement in rendering quality compared to these methods by optimising MLPs to encode 5D radiance fields. In contrast to tightly coupling 3D scenes with spatial coordinates, we introduce EscherNet as an alternative for 3D representation learning by optimising a neural network to learn the interaction between multi-view posed images, independent of any coordinate system.

**Novel View Synthesis** The success of NeRF has sparked a wave of follow-up methods that address faster training and/or rendering efficiency, by incorporating different variants of space discretisation [3, 12, 14], codebooks [48], and



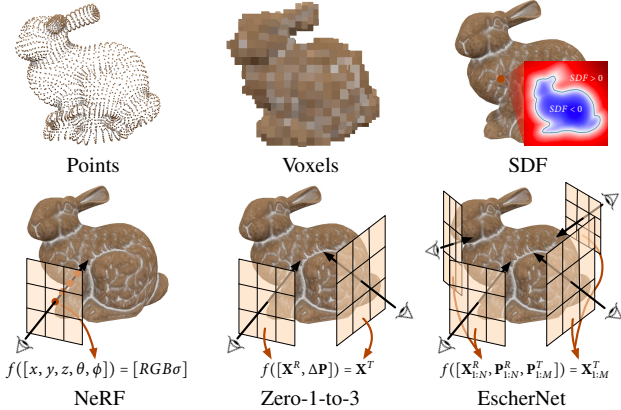


Figure 2. **3D representations overview.** EscherNet generates a set of  $M$  target views  $\mathbf{X}_{1:M}^T$  based on their camera poses  $\mathbf{P}_{1:M}^T$ , leveraging information gained from a set of  $N$  reference views  $\mathbf{X}_{1:N}^R$  and their camera poses  $\mathbf{P}_{1:N}^R$ . EscherNet presents a new way of learning implicit 3D representations by only considering the relative camera transformation between the camera poses of  $\mathbf{P}^R$  and  $\mathbf{P}^T$ , making it easier to scale with multi-view posed images, independent of any specific coordinate systems.

encodings using hash tables [30] or Gaussians [18].

To enhance NeRF’s generalisation ability across diverse scenes and in a few-shot setting, PixelNeRF [58] attempts to learn a scene prior by jointly optimising multiple scenes, but it is constrained by the high computational demands required by volumetric rendering. Various other approaches have addressed this issue by introducing regularisation techniques, such as incorporating low-level priors from local patches [33], ensuring semantic consistency [16], considering adjacent ray frequency [56], and incorporating depth signals [9]. In contrast, EscherNet encodes scenes directly through the image space, enabling the learning of more generalised scene priors through large-scale datasets.

**3D Diffusion Models** The emergence of 2D generative diffusion models has shown impressive capabilities in generating realistic objects and scenes [15, 42]. This progress has inspired the early design of text-to-3D diffusion models, such as DreamFusion [38] and Magic3D [19], by optimising a radiance field guided by score distillation sampling (SDS) from these pre-trained 2D diffusion models. However, SDS necessitates computationally intensive iterative optimisation, often requiring up to an hour for convergence. Additionally, these methods, including recently proposed image-to-3D generation approaches [8, 27, 55], frequently yield unrealistic 3D generation results due to their limited 3D understanding, giving rise to challenges such as the multi-face Janus problem.

To integrate 3D priors more efficiently, an alternative approach involves training 3D generative models directly on 3D datasets, employing representations like point clouds [31] or neural fields [4, 11, 17]. However, this de-

sign depends on 3D operations, such as 3D convolution and volumetric rendering, which are computationally expensive and challenging to scale.

To address this issue, diffusion models trained on multi-view posed data have emerged as a promising direction, designed with no 3D operations. Zero-1-to-3 [21] stands out as a pioneering work, learning view synthesis from paired 2D posed images rendered from large-scale 3D object datasets [6, 7]. However, its capability is limited to generating a single target view conditioned on a single reference view. Recent advancements in multi-view diffusion models [20, 22, 23, 43, 44, 57] focused on 3D generation and can only generate a fixed number of target views with fixed camera poses. In contrast, EscherNet can generate an unrestricted number of target views with arbitrary camera poses, offering superior flexibility in view synthesis.

### 3. EscherNet

**Problem Formulation and Notation** In EscherNet, we recast the view synthesis as a conditional generative modelling problem, formulated as:

$$\mathcal{X}^T \sim p(\mathcal{X}^T | \mathcal{X}^R, \mathcal{P}^R, \mathcal{P}^T). \quad (1)$$

Here,  $\mathcal{X}^T = \{\mathbf{X}_{1:M}^T\}$  and  $\mathcal{P}^T = \{\mathbf{P}_{1:M}^T\}$  represent a set of  $M$  target views  $\mathbf{X}_{1:M}^T$  with their global camera poses  $\mathbf{P}_{1:M}^T$ . Similarly,  $\mathcal{X}^R = \{\mathbf{X}_{1:N}^R\}$  and  $\mathcal{P}^R = \{\mathbf{P}_{1:N}^R\}$  represent a set of  $N$  reference views  $\mathbf{X}_{1:N}^R$  with their global camera poses  $\mathbf{P}_{1:N}^R$ . Both  $N$  and  $M$  can take on arbitrary values during both model training and inference.

We propose a neural architecture design, such that the generation of each target view  $\mathbf{X}_i^T \in \mathcal{X}^T$  solely depends on its relative camera transformation to the reference views  $(\mathbf{P}_j^R)^{-1}\mathbf{P}_i^T, \forall \mathbf{P}_j^R \in \mathcal{P}^R$ , introduced next.

#### 3.1. Architecture Design

We design EscherNet following two key principles: i) It builds upon an existing 2D diffusion model, inheriting its strong web-scale prior through large-scale training, and ii) It encodes camera poses for each view/image, similar to how language models encode token positions for each token. So our model can naturally handle an arbitrary number of views for *any-to-any view synthesis*.

**Multi-View Generation** EscherNet can be seamlessly integrated with any 2D diffusion model with a transformer architecture, with *no additional learnable parameters*. In this work, we design EscherNet by adopting a latent diffusion architecture, specifically StableDiffusion v1.5 [42]. This choice enables straightforward comparisons with numerous 3D diffusion models that also leverage the same backbone (more details in the experiment section).

To tailor the Stable Diffusion model, originally designed for text-to-image generation, to multi-view generation as

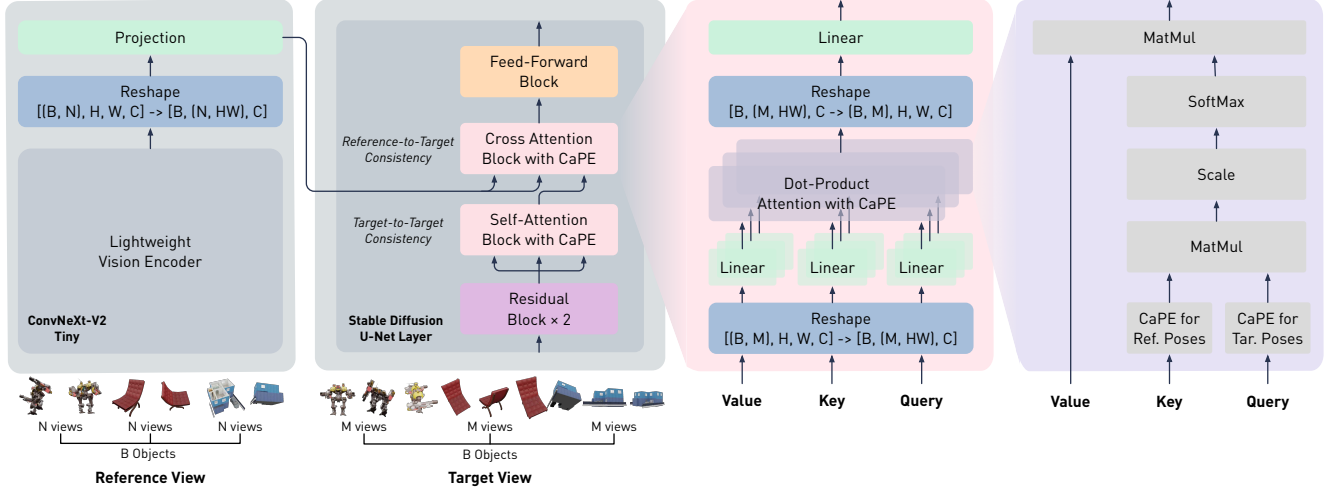


Figure 3. **EscherNet architecture details.** EscherNet adopts the Stable Diffusion architectural design with minimal but important modifications. The lightweight vision encoder captures both high-level and low-level signals from  $N$  reference views. In U-Net, we apply self-attention within  $M$  target views to encourage target-to-target consistency, and cross-attention within  $M$  target and  $N$  reference views (encoded by the image encoder) to encourage reference-to-target consistency. In each attention block, CaPE is employed for the key and query, allowing the attention map to learn with relative camera poses, independent of specific coordinate systems.

applied in EscherNet, several key modifications are implemented. In the original Stable Diffusion’s denoiser U-Net, the self-attention block was employed to learn interactions within different patches within the same image. In EscherNet, we re-purpose this self-attention block to facilitate learning interactions within distinct patches across  $M$  different target views, thereby ensuring target-to-target consistency. Likewise, the cross-attention block, originally used to integrate textual information into image patches, is repurposed in EscherNet to learn interactions within  $N$  reference to  $M$  target views, ensuring reference-to-target consistency.

**Conditioning Reference Views** In view synthesis, it is crucial that the conditioning signals accurately capture both the high-level semantics and low-level texture details present in the reference views. Previous works in 3D diffusion models [21, 22] have employed the strategy of encoding high-level signals through a frozen CLIP pre-trained ViT [41] and encoding low-level signals by concatenating the reference image into the input of the U-Net of Stable Diffusion. However, this design choice inherently constrains the model to handle only one single view.

In EscherNet, we choose to incorporate both high-level and low-level signals in the conditioning image encoder, representing reference views as sets of tokens. This design choice allows our model to maintain flexibility in handling a variable number of reference views. Early experiments have confirmed that using a frozen CLIP-ViT alone may fail to capture low-level textures, preventing the model from accurately reproducing the original reference views given the same reference view poses as target poses. While fine-tuning the CLIP-ViT could address this issue,

it poses challenges in terms of training efficiency. Instead, we opt to fine-tune a lightweight vision encoder, specifically ConvNeXtv2-Tiny [53], which is a highly efficient CNN architecture. This architecture is employed to compress our reference views to smaller resolution image features. We treat these image features as conditioning tokens, effectively representing each reference view. This configuration has proven to be sufficient in our experiments, delivering superior results in generation quality while simultaneously maintaining high training efficiency.

### 3.2. Camera Positional Encoding (CaPE)

To encode camera poses efficiently and accurately into reference and target view tokens within a transformer architecture, we introduce Camera Positional Encoding (CaPE), drawing inspiration from recent advancements in the language domain. We first briefly examine the distinctions between these two domains.

- In language, token positions (associated with each word) follow a *linear and discrete* structure, and their length can be *infinite*. Language models are typically trained with fixed maximum token counts (known as context length), and it remains an ongoing research challenge to construct a positional encoding that enables the model to behave reasonably beyond this fixed context length [13, 35].

- In 3D vision, token positions (associated with each camera) follow a *cyclic, continuous, and bounded* structure for rotations and a *linear, continuous, and unbounded* structure for translations. Importantly, unlike the language domain where the token position always starts from zero, there are no *standardised absolute global camera poses* in

a 3D space. The relationship between two views depends solely on their relative camera transformation.

We now present two distinct designs for spatial position encoding, representing camera poses using 4 DoF for object-centric rendering and 6 DoF for the generic case, respectively. Our design strategy involves directly applying a transformation on global camera poses embedded in the token feature, which allows the dot-product attention to directly encode the relative camera transformation, independent of any coordinate system.

**4 DoF CaPE** In the case of 4 DoF camera poses, we adopt a spherical coordinate system, similar to [21, 22], denoted as  $\mathbf{P} = \{\alpha, \beta, \gamma, r\}$  including azimuth, elevation, camera orientation along the look-at direction, and camera distance (radius), each position component is *disentangled*.

Mathematically, the position encoding function  $\pi(\mathbf{v}, \mathbf{P})$ , characterised by its  $d$ -dimensional token feature  $\mathbf{v} \in \mathbb{R}^d$  and pose  $\mathbf{P}$ , should satisfy the following conditions:

$$\langle \pi(\mathbf{v}_1, \theta_1), \pi(\mathbf{v}_2, \theta_2) \rangle = \langle \pi(\mathbf{v}_1, \theta_1 - \theta_2), \pi(\mathbf{v}_2, 0) \rangle, \quad (2)$$

$$\langle \pi(\mathbf{v}_1, r_1), \pi(\mathbf{v}_2, r_2) \rangle = \langle \pi(\mathbf{v}_1, r_1/r_2), \pi(\mathbf{v}_2, 1) \rangle. \quad (3)$$

Here  $\langle \cdot, \cdot \rangle$  represents the dot product operation,  $\theta_{1,2} \in \{\alpha, \beta, \gamma\}$ , within  $\alpha, \gamma \in [0, 2\pi), \beta \in [0, \pi)$ , and  $r_{1,2} > 0$ . Essentially, the relative 4 DoF camera transformation is decomposed to the relative angle difference in rotation and the relative scale difference in view radius.

Notably, Eq. 2 aligns with the formula of rotary position encoding (RoPE) [46] derived in the language domain. Given that  $\log(r_1) - \log(r_2) = \log(s \cdot r_1) - \log(s \cdot r_2)$  (for any scalar  $s > 0$ ), we may elegantly combine both Eq. 2 and Eq. 3 in a unified formulation using the design strategy in RoPE by transforming feature vector  $\mathbf{v}$  with a block diagonal rotation matrix  $\phi(\mathbf{P})$  encoding  $\mathbf{P}$ .

– **4 DoF CaPE**:  $\pi(\mathbf{v}, \mathbf{P}) = \phi(\mathbf{P})\mathbf{v}$ ,

$$\phi(\mathbf{P}) = \begin{bmatrix} \Psi & 0 & \cdots & 0 \\ 0 & \Psi & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \Psi \end{bmatrix}, \quad \Psi = \begin{bmatrix} \Psi_\alpha & 0 & \cdots & 0 \\ 0 & \Psi_\beta & 0 & \vdots \\ \vdots & 0 & \Psi_\gamma & 0 \\ 0 & \cdots & 0 & \Psi_r \end{bmatrix}. \quad (4)$$

$$\text{Rotation: } \Psi_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (5)$$

$$\text{View Radius: } \Psi_r = \begin{bmatrix} \cos(f(r)) & -\sin(f(r)) \\ \sin(f(r)) & \cos(f(r)) \end{bmatrix}, \quad (6)$$

$$\text{where } f(r) = \pi \frac{\log r - \log r_{\min}}{\log r_{\max} - \log r_{\min}} \in [0, \pi]. \quad (7)$$

Here,  $\dim(\mathbf{v}) = d$  should be divisible by  $2|\mathbf{P}| = 8$ . Note, it's crucial to apply Eq. 7 to constrain  $\log r$  within the range of rotation  $[0, \pi]$ , so we ensure the dot product monotonically corresponds to its scale difference.

**6 DoF CaPE** In the case of 6 DoF camera poses, denoted as  $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$ , each position component is *entangled*, implying that we are not able to reformulate as a multi-dimensional position as in 4 DoF camera poses.

Mathematically, the position encoding function  $\pi(\mathbf{v}, \mathbf{P})$  should now satisfy the following condition:

$$\langle \pi(\mathbf{v}_1, \mathbf{P}_1), \pi(\mathbf{v}_2, \mathbf{P}_2) \rangle = \langle \pi(\mathbf{v}_1, \mathbf{P}_2^{-1}\mathbf{P}_1), \pi(\mathbf{v}_2, \mathbf{I}) \rangle. \quad (8)$$

Let's apply a similar strategy as used in 4 DoF CaPE, which increases the dimensionality of  $\mathbf{P} \in \mathbb{R}^{4 \times 4}$  to  $\phi(\mathbf{P}) \in \mathbb{R}^{d \times d}$  by reconstructing it as a block diagonal matrix, with each diagonal element being  $\mathbf{P}$ . Since  $\phi(\mathbf{P})$  also forms a real Lie group, we may construct  $\pi(\cdot, \cdot)$  for a key and query using the following equivalence:

$$(\phi(\mathbf{P}_2^{-1}\mathbf{P}_1)\mathbf{v}_1)^\top (\phi(\mathbf{I})\mathbf{v}_2) = (\mathbf{v}_1^\top \phi(\mathbf{P}_1^\top \mathbf{P}_2^{-\top})) \mathbf{v}_2 \quad (9)$$

$$= (\mathbf{v}_1^\top \phi(\mathbf{P}_1^\top)) (\phi(\mathbf{P}_2^{-\top})\mathbf{v}_2) = (\phi(\mathbf{P}_1)\mathbf{v}_1)^\top (\phi(\mathbf{P}_2^{-\top})\mathbf{v}_2) \quad (10)$$

$$= \langle \pi(\mathbf{v}_1, \phi(\mathbf{P}_1)), \pi(\mathbf{v}_2, \phi(\mathbf{P}_2^{-\top})) \rangle. \quad (11)$$

– **6 DoF CaPE**:  $\pi(\mathbf{v}, \mathbf{P}) = \phi(\mathbf{P})\mathbf{v}$ ,

$$\phi(\mathbf{P}) = \begin{bmatrix} \Psi & 0 & \cdots & 0 \\ 0 & \Psi & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \Psi \end{bmatrix}, \quad \Psi = \begin{cases} \mathbf{P} & \text{if key} \\ \mathbf{P}^{-\top} & \text{if query} \end{cases}. \quad (12)$$

Here,  $\dim(\mathbf{v}) = d$  should be divisible by  $\dim(\mathbf{P}) = 4$ . Similarly, we need to re-scale the translation  $\mathbf{t}$  for each scene within a unit range for efficient model training.

In both 4 and 6 DoF CaPE implementation, we can efficiently perform matrix multiplication by simply reshaping the vector  $\mathbf{v}$  to match the dimensions of  $\Psi$  (8 for 4 DoF, 4 for 6 DoF), ensuring faster computation. The PyTorch implementation is attached in Appendix A.

## 4. Experiments

**Training Datasets** In this work, we focus on object-centric view synthesis, training our model on Objaverse-1.0 which consists of 800K objects [7]. This setting allows us to fairly compare with all other 3D diffusion model baselines trained on the same dataset. We adopt the same training data used in Zero-1-to-3 [21], which contains 12 randomly rendered views per object with randomised environment lighting. To ensure the data quality, we filter out empty rendered images, which make up roughly 1% of the training data.

We trained and reported results using EscherNet with both 4 DoF and 6 DoF CaPE. Our observations revealed that 6 DoF CaPE exhibits a slightly improved performance, which we attribute to its more compressed representation space. However, empirically, we found that 4 DoF CaPE yields visually more consistent results when applied to real-world images. Considering that the training data is confined within a 4 DoF object-centric setting, we present EscherNet with 4 DoF CaPE in the main paper. The results obtained with 6 DoF CaPE are provided in Appendix C.

In all experiments, we re-evaluate the baseline models by using their officially open-sourced checkpoints on the same set of reference views for a fair comparison. Further details on our experiment setting and training strategies are provided in Appendix B.



### 4.1. Results on Novel View Synthesis

We evaluate EscherNet in novel view synthesis on the Google Scanned Objects dataset (GSO) [10] and the RTMV dataset [49], comparing with 3D diffusion models for view synthesis, such as Zero-1-to-3 [21] and RealFusion [27] (primarily for generation quality with minimal reference views). Additionally, we also evaluate on NeRF Synthetic Dataset [29], comparing with state-of-the-art scene-specific neural rendering methods, such as InstantNGP [30] and 3D Gaussian Splatting [18] (primarily for rendering accuracy with multiple reference views).

Notably, many other 3D diffusion models [20, 22, 23, 43, 57] prioritise 3D generation rather than view synthesis. This limitation confines them to predicting target views with *fixed target poses*, making them not directly comparable.

**Compared to 3D Diffusion Models** In Tab. 1 and Fig. 5, we show that EscherNet significantly outperforms 3D diffusion baselines, by a large margin, both quantitatively and qualitatively. Particularly, we outperform Zero-1-to-3-XL despite it being trained on  $\times 10$  more training data, and RealFusion despite it requiring expensive score distillation for iterative scene optimisation [38]. It’s worth highlighting that Zero-1-to-3 by design is inherently limited to generating a single target view and cannot ensure self-consistency across multiple target views, while EscherNet can generate multiple consistent target views jointly and provides more precise camera control.

	Training Data	# Ref. Views	GSO-30			RTMV		
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
RealFusion	-	1	12.76	0.758	0.382	-	-	-
Zero123	800K	1	18.51	0.856	0.127	10.16	0.505	0.418
Zero123-XL	10M	1	18.93	0.856	0.124	10.59	0.520	0.401
EscherNet	800k	1	20.24	0.884	0.095	10.56	0.518	0.410
EscherNet	800k	2	22.91	0.908	0.064	12.66	0.585	0.301
EscherNet	800k	3	24.09	0.918	0.052	13.59	0.611	0.258
EscherNet	800k	5	25.09	0.927	0.043	14.52	0.633	0.222
EscherNet	800k	10	25.90	0.935	0.036	15.55	0.657	0.185

Table 1. **Novel view synthesis performance on GSO and RTMV datasets.** EscherNet outperforms Zero-1-to-3-XL with significantly less training data and RealFusion without extra SDS optimisation. Additionally, EscherNet’s performance exhibits further improvement with the inclusion of more reference views.

**Compared to Neural Rendering Methods** In Tab. 2 and Fig. 4, we show that EscherNet again offers plausible view synthesis in a zero-shot manner, without scene-specific optimisation required by both InstantNGP and 3D Gaussian Splatting. Notably, EscherNet leverages a generalised understanding of objects acquired through large-scale training, allowing it to interpret given views both semantically and spatially, even when conditioned on a limited number of reference views. However, with an increase in the number of reference views, both InstantNGP and 3D Gaussian

	# Reference Views (Less $\rightarrow$ More)							
	1	2	3	5	10	20	50	100
<b>InstantNGP (Scene Specific Training)</b>								
PSNR $\uparrow$	10.92	12.42	14.27	18.17	22.96	24.99	26.86	27.30
SSIM $\uparrow$	0.449	0.521	0.618	0.761	0.881	0.917	0.946	0.953
LPIPS $\downarrow$	0.627	0.499	0.391	0.228	0.091	0.058	0.034	0.031
<b>GaussianSplatting (Scene Specific Training)</b>								
PSNR $\uparrow$	9.44	10.78	12.87	17.09	23.04	25.34	26.98	27.11
SSIM $\uparrow$	0.391	0.432	0.546	0.732	0.876	0.919	0.942	0.944
LPIPS $\downarrow$	0.610	0.541	0.441	0.243	0.085	0.054	0.041	0.041
<b>EscherNet (Zero Shot Inference)</b>								
PSNR $\uparrow$	13.36	14.95	16.19	17.16	17.74	17.91	18.05	18.15
SSIM $\uparrow$	0.659	0.700	0.729	0.748	0.761	0.765	0.769	0.771
LPIPS $\downarrow$	0.291	0.208	0.161	0.127	0.114	0.106	0.099	0.097

Table 2. **Novel view synthesis performance on NeRF Synthetic dataset.** EscherNet outperforms both InstantNGP and Gaussian Splatting when provided with fewer than five reference views while requiring no scene-specific optimisation. However, as the number of reference views increases, both methods show a more significant improvement in rendering quality.

Splatting exhibit a significant improvement in the rendering quality. To achieve a photo-realistic neural rendering while retaining the advantages of a generative formulation remains an important research challenge.

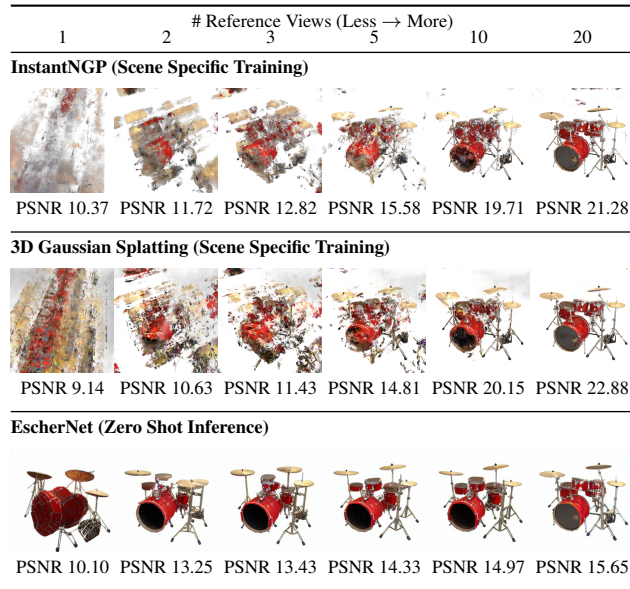


Figure 4. **Generated views visualisation on the NeRF Synthetic drum scene.** EscherNet generates plausible view synthesis even when provided with very limited reference views, while neural rendering methods fail to generate any meaningful content. However, when we have more than 10 reference views, scene-specific methods exhibit a substantial improvement in rendering quality. We report the mean PSNR averaged across all test views from the drum scene. Results for other scenes and/or with more reference views are shown in Appendix D.

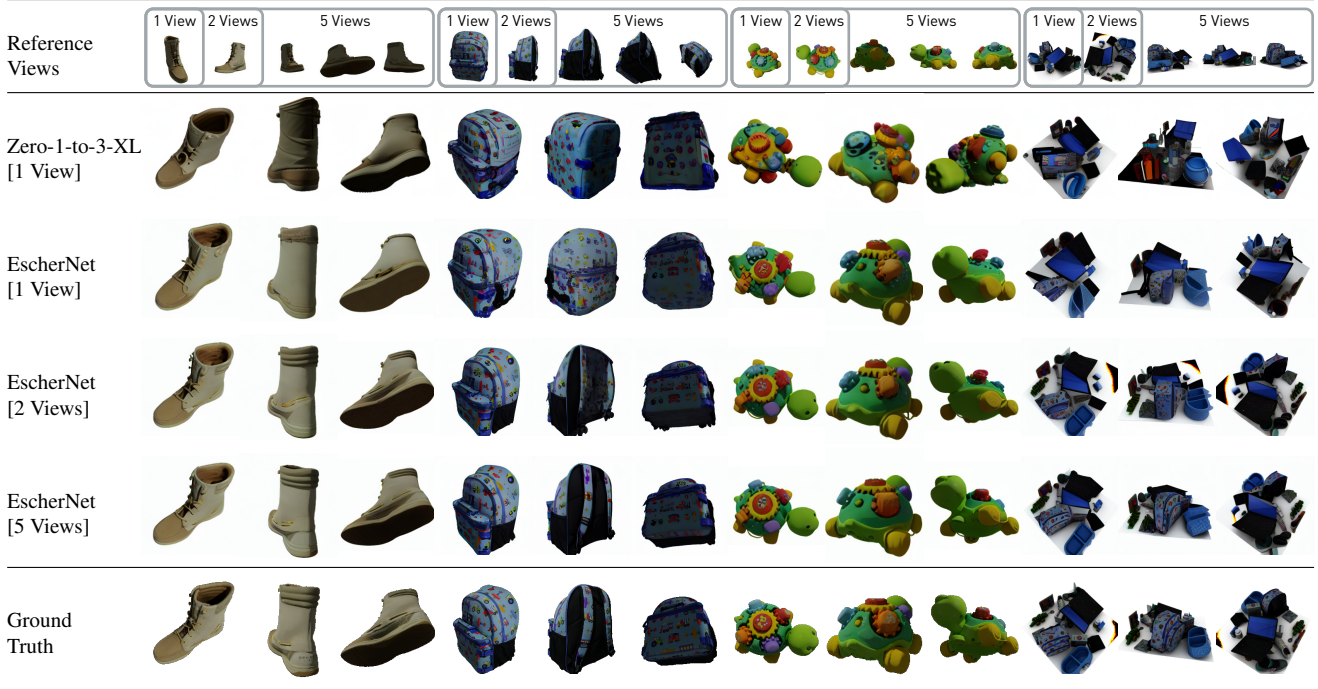


Figure 5. **Novel view synthesis visualisation on GSO and RTMV datasets.** EscherNet outperforms Zero-1-to-3-XL, delivering superior generation quality and finer camera control. Notably, when conditioned with additional views, EscherNet exhibits an enhanced resemblance of the generated views to ground-truth textures, revealing more refined texture details such as in the backpack straps and turtle shell.

## 4.2. Results on 3D Generation

In this section, we perform single/few-image 3D generation on the GSO dataset. We compare with SoTA 3D generation baselines: Point-E [31] for direct point cloud generation, Shape-E [17] for direct NeRF generation, DreamGaussian [17] for optimising 3D Gaussian [18] with SDS guidance, One-2-3-45 [20] for decoding an SDF using multiple views predicted from Zero-1-to-3, and SyncDreamer [22] for fitting an SDF using NeuS [51] from 16 consistent fixed generated views. We additionally include NeuS trained on reference views for few-image 3D reconstruction baselines.

Given any reference views, EscherNet can generate multiple 3D consistent views, allowing for the straightforward adoption with NeuS [51] for 3D reconstruction. We generate 36 fixed views, varying the azimuth from  $0^\circ$  to  $360^\circ$  with a rendering every  $30^\circ$  at a set of elevations ( $-30^\circ$ ,  $0^\circ$ ,  $30^\circ$ ), which serve as inputs for our NeuS reconstruction.

**Results** In Tab. 3 and Fig. 6, we show that EscherNet stands out by achieving significantly superior 3D reconstruction quality compared to other image-to-3D generative models. Specifically, EscherNet demonstrates an approximate 25% improvement in Chamfer distance over SyncDreamer, considered as the current best model, when conditioned on a single reference view, and a 60% improvement when conditioned on 10 reference views. This impressive performance is attributed to EscherNet’s ability to flexibly

	# Ref. Views	Chamfer Dist. ↓	Volume IoU ↑
Point-E	1	0.0447	0.2503
Shape-E	1	0.0448	0.3762
One2345	1	0.0632	0.4209
One2345-XL	1	0.0667	0.4016
DreamGaussian	1	0.0605	0.3757
DreamGaussian-XL	1	0.0459	0.4531
SyncDreamer	1	0.0400	0.5220
NeuS	3	0.0366	0.5352
NeuS	5	0.0245	0.6742
NeuS	10	0.0195	0.7264
EscherNet	1	0.0314	0.5974
EscherNet	2	0.0215	0.6868
EscherNet	3	0.0190	0.7189
EscherNet	5	0.0175	0.7423
EscherNet	10	0.0167	0.7478

Table 3. **3D reconstruction performance on GSO.** EscherNet outperforms all other image-to-3D baselines in generating more visually appealing with accurate 3D geometry, particularly when conditioned on multiple reference views.

handle any number of reference and target views, providing comprehensive and accurate constraints for 3D geometry. In contrast, SyncDreamer faces challenges due to sensitivity to elevation angles and constraints imposed by a fixed  $30^\circ$  elevation angle by design, thus hindering learning a holistic representation of complex objects. This limitation results in degraded reconstruction, particularly evident in the lower regions of the generated geometry.

## 4.3. Results on Text-to-3D Generation

EscherNet’s flexibility in accommodating any number of reference views enables a straightforward approach to the

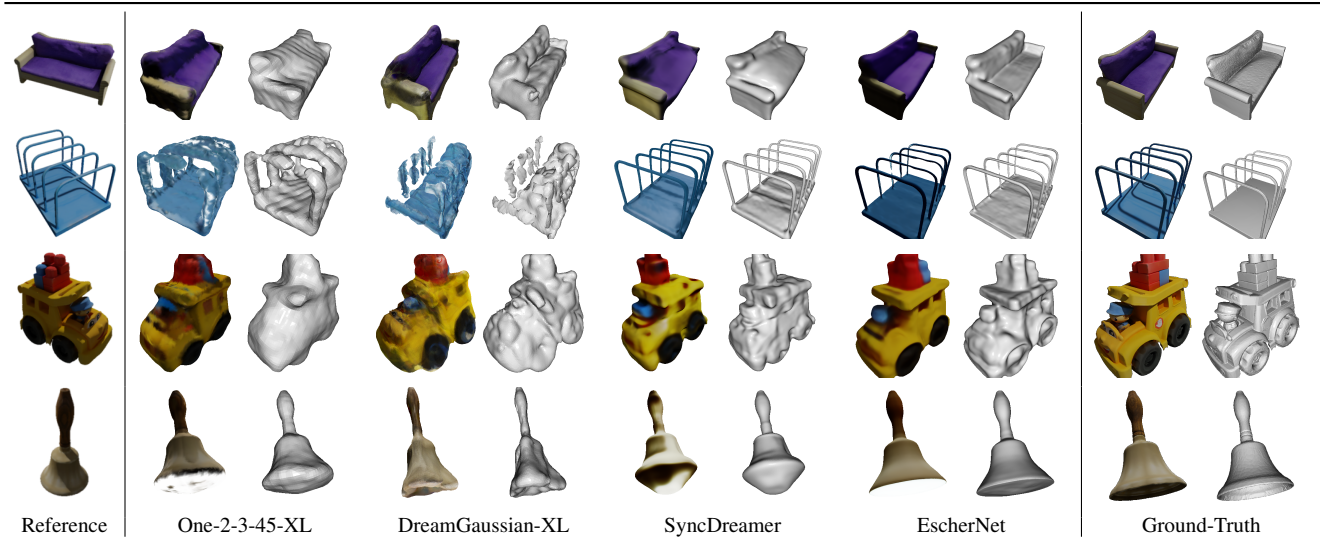


Figure 6. **Single view 3D reconstruction visualisation on GSO.** EscherNet’s ability to generate dense and consistent novel views significantly improves the reconstruction of complete and well-constrained 3D geometry. In contrast, One-2-3-45-XL and DreamGaussian-XL, despite leveraging a significantly larger pre-trained model, tend to produce over-smoothed and noisy reconstructions; SyncDreamer, constrained by sparse fixed-view synthesis, struggles to tightly constrain geometry, particularly in areas in sofa and the bottom part of the bell.

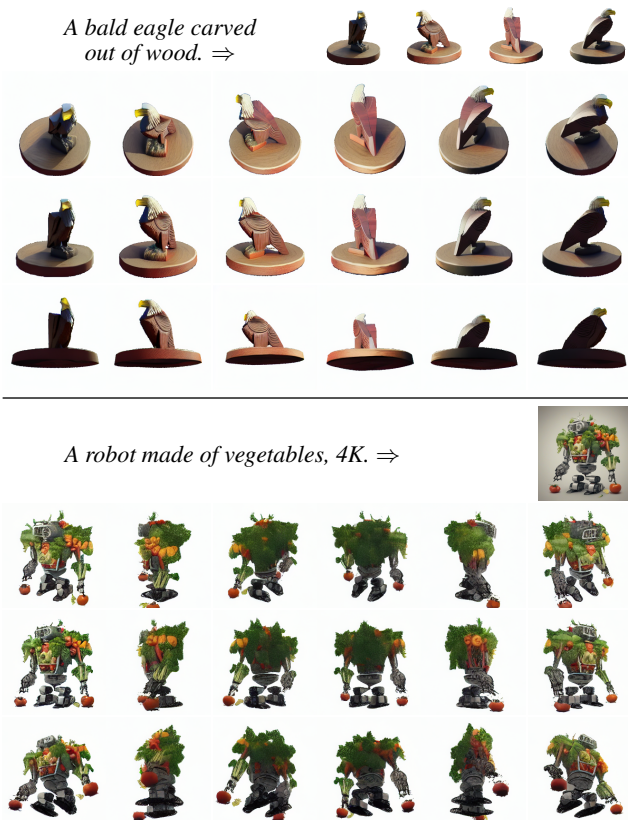


Figure 7. **Text-to-3D visualisation with MVDream (up) and SDXL (bottom).** EscherNet offers compelling and realistic view synthesis for synthetic images generated with user-provided text prompts. Additional results are shown in Appendix E.

text-to-3D generation problem by breaking it down into two stages: text-to-image, relying on any off-the-shelf text-to-image generative model, and then image-to-3D, relying on EscherNet. In Fig. 7, we present visual results of dense novel view generation using a text-to-4view model with MVDream [44] and a text-to-image model with SDXL [37]. Remarkably, even when dealing with out-of-distribution and counterfactual content, EscherNet generates consistent 3D novel views with appealing textures.

## 5. Conclusions

In this paper, we have introduced EscherNet, a multi-view conditioned diffusion model designed for scalable view synthesis. Leveraging Stable Diffusion’s 2D architecture empowered by the innovative Camera Positional Embedding (CaPE), EscherNet adeptly learns implicit 3D representations from varying number of reference views, achieving consistent 3D novel view synthesis. We provide detailed discussions and additional ablative analysis in Appendix F.

**Limitations and Discussions** EscherNet’s flexibility in handling any number of reference views allows for autoregressive generation, similar to autoregressive language models [1, 5]. While this approach significantly reduces inference time, it leads to a degraded generation quality. Additionally, EscherNet’s current capability operates within a 3 DoF setting constrained by its training dataset, which may not align with real-world scenarios, where views typically span in  $SE(3)$  space. Future work will explore scaling EscherNet with 6 DoF training data with real-world scenes, striving for a more general 3D representation.



## Acknowledgement

The research presented here was supported by Dyson Technology Ltd. Xin Kong holds a China Scholarship Council-Imperial Scholarship. We would like to thank Sayak Paul and HuggingFace for contributing the training compute that facilitated early project exploration. We would also like to acknowledge Yifei Ren for his valuable discussions on formulating the 6DoF CaPE.

## References

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 8
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2
- [3] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2
- [4] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. *arXiv preprint arXiv:2304.06714*, 2023. 3
- [5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 8
- [6] 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. 3, 17
- [7] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 5, 17
- [8] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchun Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [9] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [10] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022. 6, 13
- [11] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015*, 2023. 3
- [12] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2
- [13] Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*, 2023. 4
- [14] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3
- [16] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3
- [17] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3, 7, 13
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 2, 3, 6, 7, 13
- [19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [20] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 2, 3, 6, 7
- [21] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 3, 4, 5, 6, 13
- [22] 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. 3, 4, 5, 6, 7, 13, 17
- [23] 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](#), [3](#), [6](#), [17](#)
- [24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. [13](#)
- [25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [2](#)
- [26] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2015. [2](#)
- [27] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [3](#), [6](#)
- [28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)
- [29] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [2](#), [6](#), [13](#)
- [30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multi-resolution hash encoding. *ACM Transactions on Graphics (ToG)*, 2022. [2](#), [3](#), [6](#), [13](#)
- [31] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. [3](#), [7](#), [13](#)
- [32] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#)
- [33] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [3](#)
- [34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)
- [35] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023. [4](#)
- [36] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [2](#)
- [37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. [8](#)
- [38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. [3](#), [6](#)
- [39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [2](#)
- [41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. [4](#)
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [3](#)
- [43] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. [2](#), [3](#), [6](#)
- [44] 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. [3](#), [8](#)
- [45] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [2](#)
- [46] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. [5](#)
- [47] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [2](#)
- [48] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *Proceedings of SIGGRAPH*, 2022. [2](#)
- [49] Jonathan Tremblay, Moustafa Meshry, Alex Evans, Jan Kautz, Alexander Keller, Sameh Khamis, Thomas Müller,

- Charles Loop, Nathan Morrical, Koki Nagano, et al. Rtmv: A ray-traced multi-view synthetic dataset for novel view synthesis. *arXiv preprint arXiv:2205.07058*, 2022. 6, 13
- [50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2
- [51] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 7
- [52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 13
- [53] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4
- [54] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [55] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [56] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3
- [57] Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. *arXiv preprint arXiv:2310.03020*, 2023. 3, 6
- [58] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [59] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 2
- [60] 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 (CVPR)*, 2018. 13



## A. Python Implementation of CaPE

```
def compute_4dof_cape(v, P, s):
    """
    :param v: input feature vector with its dimension must be divisible by 8
    :param P: list = [alpha, beta, gamma, r]
    :param s: a small scalar for radius
    :return: rotated v with its corresponding camera pose P
    """
    v = v.reshape([-1, 8])
    psi = np.zeros([8, 8])
    for i in range(4):
        if i < 3:
            psi[2 * i:2 * (i + 1), 2 * i:2 * (i + 1)] = \
                np.array([[np.cos(P[i]), -np.sin(P[i]), [np.sin(P[i]), np.cos(P[i])]])
        else:
            psi[2 * i:2 * (i + 1), 2 * i:2 * (i + 1)] = \
                np.array([[np.cos(s * np.log(P[i]), -np.sin(s * np.log(P[i])),
                    [np.sin(s * np.log(P[i]), np.cos(s * np.log(P[i]))]])

    return v.dot(psi).reshape(-1)
```

Listing 1. Python implementation for 4 DoF CaPE.

```
def compute_6dof_cape(v, P, s=0.001, key=True):
    """
    :param v: input feature vector with its dimension must be divisible by 4
    :param P: 4 x 4 SE3 matrix
    :param s: a small scalar for translation
    :return: rotated v with its corresponding camera pose P
    """
    v = v.reshape([-1, 4])
    P[:3, 3] *= s
    psi = P if key else np.linalg.inv(P).T
    return v.dot(psi).reshape(-1)
```

Listing 2. Python implementation for 6 DoF CaPE.

## B. Additional Training Details and Experimental Settings

**Optimisation and Implementation** EscherNet is trained using the AdamW optimiser [24] with a learning rate of  $1 \cdot 10^{-4}$  and weight decay of 0.01 for  $[256 \times 256]$  resolution images. We incorporate cosine annealing, reducing the learning rate to  $1 \cdot 10^{-5}$  over a total of 100,000 training steps, while linearly warming up for the initial 1000 steps. To speed up training, we implement automatic mixed precision with a precision of `bfloat16` and employ gradient checkpointing. Our training batches consist of 3 reference views and 3 target views randomly sampled with replacement from 12 views for each object, with a total batch size of 672 (112 batches per GPU). The entire model training process takes 1 week on 6 NVIDIA A100 GPUs.

**Metrics** For 2D metrics used in view synthesis, we employ PSNR, SSIM [52], LPIPS [60]. For 3D metrics used in 3D generation, we employ Chamfer Distance and Volume IoU. To ensure a fair and efficient evaluation process, each baseline method and our approach are executed only once per scene per viewpoint. This practice has proven to provide stable averaged results across multiple scenes and viewpoints.

### B.1. Evaluation Details

**In NeRF Synthetic Dataset [29]**, we consider and evaluate all 8 scenes provided in the original dataset. To assess performance with varying numbers of reference views, we train all baseline methods and our approach using the same set of views randomly sampled from the training set. The evaluation is conducted on all target views defined in the test sets across all 8 scenes (with 200 views per scene). For InstantNGP [30], we run 10k steps ( $\approx 1$ min) for each scene. For 3D Gaussian Splatting [18], we run 5k steps ( $\approx 2$ min) for each scene.

**In Google Scanned Dataset (GSO) [10]**, we evaluate the same 30 objects chosen by SyncDreamer [22]. For each object, we render 25 views with randomly generated camera poses and a randomly generated environment lighting condition to construct our test set. For each object, we choose the first 10 images as our reference views and the subsequent 15 images as our target views for evaluation. It’s crucial to note that all reference and target views are rendered with random camera poses, establishing a more realistic and challenging evaluation setting compared to the evaluation setups employed in other baselines: *e.g.* SyncDreamer uses an evenly distributed environment lighting to render all GSO data, and the reference view for each object is manually selected based on human preference.<sup>1</sup> Additionally, the evaluated target view is also manually selected based on human preference chosen among four independent generations.<sup>2</sup>

In evaluating 3D generation, we randomly sample 4096 points evenly distributed from the generated 3D mesh or point cloud across all methods. Each method’s generated mesh is aligned to the ground-truth mesh using the camera pose of the reference views. Specifically in Point-E [31] and Shape-E [17], we rotate 90/180 degrees along each x/y/z axis to determine the optimal alignment for the final mesh pose. Our evaluation approach again differs from SyncDreamer, which initially projects the 3D mesh into their fixed 16 generated views to obtain depth maps. Then, points are sampled from these depth maps for the final evaluation.<sup>3</sup>

**In RTMV Dataset [49]**, we follow the evaluation setting used in Zero-1-to-3 [21], which consists of 10 complex scenes featuring a pile of multiple objects from the GSO dataset. Similar to the construction of our GSO test set, we then randomly select a fixed subset of the first 10 images as our reference views and the subsequent 10 views as our target views for evaluation.

---

<sup>1</sup><https://github.com/liuyuan-pal/SyncDreamer/issues/21>

<sup>2</sup><https://github.com/liuyuan-pal/SyncDreamer/issues/21#issuecomment-1770345260>

<sup>3</sup><https://github.com/liuyuan-pal/SyncDreamer/issues/44>

### C. Additional Results on 6 DoF CaPE

To validate the effectiveness of the 6 DoF CaPE design, we demonstrate its performance in novel view synthesis on GSO and RTMV datasets in Tab. 4a and on the NeRF Synthetic dataset in Tab. 4c. We also provide 3D reconstruction results on GSO dataset in Tab. 4b. It is evident that EscherNet with 6 DoF CaPE achieves comparable, and often, slightly improved results when compared to our 4 DoF CaPE design.

	Training Data	# Ref. Views	GSO-30			RTMV		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RealFusion	-	1	12.76	0.758	0.382	-	-	-
Zero123	800K	1	18.51	0.856	0.127	10.16	0.505	0.418
Zero123-XL	10M	1	18.93	0.856	0.124	10.59	0.520	0.401
EscherNet - 4 DoF	800k	1	20.24	0.884	0.095	10.56	0.518	0.410
EscherNet - 4 DoF	800k	2	22.91	0.908	0.064	12.66	0.585	0.301
EscherNet - 4 DoF	800k	3	24.09	0.918	0.052	13.59	0.611	0.258
EscherNet - 4 DoF	800k	5	25.09	0.927	0.043	14.52	0.633	0.222
EscherNet - 4 DoF	800k	10	25.90	0.935	0.036	15.55	0.657	0.185
EscherNet - 6 DoF	800k	1	20.89	0.886	0.093	12.30	0.569	0.332
EscherNet - 6 DoF	800k	2	23.92	0.917	0.057	14.18	0.618	0.252
EscherNet - 6 DoF	800k	3	25.21	0.927	0.045	15.06	0.643	0.217
EscherNet - 6 DoF	800k	5	26.59	0.937	0.036	15.71	0.663	0.190
EscherNet - 6 DoF	800k	10	27.75	0.947	0.030	16.58	0.688	0.160

	# Ref. Views	Chamfer Dist. ↓	Volume IoU ↑
Point-E	1	0.0447	0.2503
Shape-E	1	0.0448	0.3762
One2345	1	0.0632	0.4209
One2345-XL	1	0.0667	0.4016
DreamGaussian	1	0.0605	0.3757
DreamGaussian-XL	1	0.0459	0.4531
SyncDreamer	1	0.0400	0.5220
NeuS	3	0.0366	0.5352
NeuS	5	0.0245	0.6742
NeuS	10	0.0195	0.7264
EscherNet - 4 DoF	1	0.0314	0.5974
EscherNet - 4 DoF	2	0.0215	0.6868
EscherNet - 4 DoF	3	0.0190	0.7189
EscherNet - 4 DoF	5	0.0175	0.7423
EscherNet - 4 DoF	10	0.0167	0.7478
EscherNet - 6 DoF	1	0.0274	0.6382
EscherNet - 6 DoF	2	0.0196	0.7100
EscherNet - 6 DoF	3	0.0180	0.7348
EscherNet - 6 DoF	5	0.0176	0.7392
EscherNet - 6 DoF	10	0.0160	0.7628

(a) Novel view synthesis performance on GSO and RTMV datasets.

(b) 3D reconstruction performance on GSO.

	# Reference Views (Less → More)							
	1	2	3	5	10	20	50	100
<b>InstantNGP (Scene Specific Training)</b>								
PSNR↑	10.92	12.42	14.27	18.17	22.96	24.99	26.86	27.30
SSIM↑	0.449	0.521	0.618	0.761	0.881	0.917	0.946	0.953
LPIPS↓	0.627	0.499	0.391	0.228	0.091	0.058	0.034	0.031
<b>GaussianSplatting (Scene Specific Training)</b>								
PSNR↑	9.44	10.78	12.87	17.09	23.04	25.34	26.98	27.11
SSIM↑	0.391	0.432	0.546	0.732	0.876	0.919	0.942	0.944
LPIPS↓	0.610	0.541	0.441	0.243	0.085	0.054	0.041	0.041
<b>EscherNet - 4 DoF (Zero Shot Inference)</b>								
PSNR↑	13.36	14.95	16.19	17.16	17.74	17.91	18.05	18.15
SSIM↑	0.659	0.700	0.729	0.748	0.761	0.765	0.769	0.771
LPIPS↓	0.291	0.208	0.161	0.127	0.114	0.106	0.099	0.097
<b>EscherNet - 6 DoF (Zero Shot Inference)</b>								
PSNR↑	13.73	15.66	16.91	17.72	18.47	18.77	19.24	19.28
SSIM↑	0.664	0.712	0.745	0.762	0.779	0.786	0.795	0.796
LPIPS↓	0.294	0.197	0.149	0.120	0.103	0.095	0.085	0.084

(c) Novel view synthesis performance on NeRF Synthetic dataset.

Table 4. EscherNet 6 DoF presents a similar and sometimes improved performance than EscherNet 4 DoF.



## D. Additional Results on NeRF Synthetic Dataset

We present additional visualisation on the NeRF Synthetic Dataset using EscherNet trained with 4 DoF CaPE.

1	2	3	# Reference Views (Less → More)		20	50	100
<b>InstantNGP (Scene Specific Training)</b>							
							
PSNR 9.45	PSNR 11.41	PSNR 13.64	PSNR 19.30	PSNR 23.14	PSNR 26.18	PSNR 28.54	PSNR 28.87
							
PSNR 10.37	PSNR 11.72	PSNR 12.82	PSNR 15.58	PSNR 19.71	PSNR 21.28	PSNR 23.09	PSNR 23.78
<b>3D Gaussian Splatting (Scene Specific Training)</b>							
							
PSNR 8.07	PSNR 9.16	PSNR 11.72	PSNR 17.32	PSNR 24.19	PSNR 25.34	PSNR 26.98	PSNR 29.01
							
PSNR 9.14	PSNR 10.63	PSNR 11.43	PSNR 14.81	PSNR 20.15	PSNR 22.88	PSNR 23.49	PSNR 23.51
<b>EscherNet (Zero Shot Inference)</b>							
							
PSNR 10.86	PSNR 10.80	PSNR 15.51	PSNR 17.07	PSNR 17.40	PSNR 17.38	PSNR 17.77	PSNR 17.85
							
PSNR 10.10	PSNR 13.25	PSNR 13.43	PSNR 14.33	PSNR 14.97	PSNR 15.65	PSNR 15.70	PSNR 15.90

Table 5. Novel View Synthesis on NeRF Synthetic Dataset. We report the average PSNR per scene, conditioned on the respective number of reference views.

## E. Additional Results on Text-to-3D

We present additional visualisation on text-to-image-to-3D using EscherNet trained with 4 DoF CaPE.




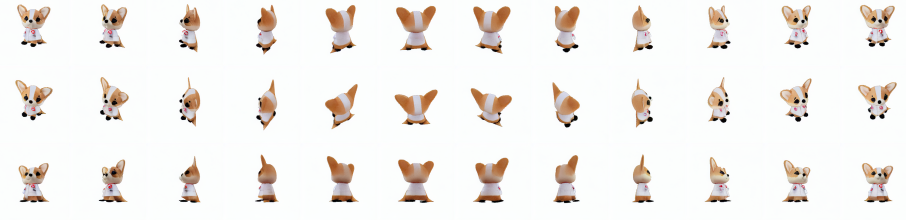

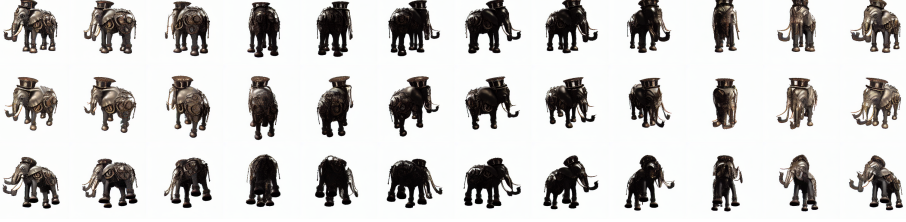






<p>A robot made of vegetables.</p> 	
<p>A nurse corgi.</p> 	
<p>A cute steampunk elephant.</p> 	
<p>A bull dog wearing a black pirate hat.</p> 	
<p>An astronaut riding a horse.</p> 	
<p>Medieval House, grass, medieval, medieval-decor, 3d asset.</p> 	

Table 6. Text-to-3D generation with SDXL (top 3) and MVDream (bottom 3).

## F. Additional Discussions, Limitations and Future Work

**Direct v.s. Autoregressive Generation** EscherNet’s flexibility in handling arbitrary numbers of reference and target views offers multiple choices for view synthesis. In our experiments, we employ the straightforward direct generation to jointly generate all target views. Additionally, an alternative approach is autoregressive generation, where target views are generated sequentially, similar to text generation with autoregressive language models.

For generating a large number of target views, autoregressive generation can be significantly faster than direct generation (e.g. more than  $20\times$  faster for generating 200 views). This efficiency gain arises from converting a quadratic inference cost into a linear inference cost in each self-attention block. However, it’s important to note that autoregressive generation may encounter a *content drifting problem* in our current design, where the generated quality gradually decreases as each newly generated view depends on previously non-perfect generated views. Autoregressive generation boasts many advantages in terms of inference efficiency and is well-suited for specific scenarios like SLAM (Simultaneous Localization and Mapping). As such, enhancing rendering quality in such a setting represents an essential avenue for future research.

**Stochasticity and Consistency in Multi-View Generation** We also observe that to enhance the target view synthesis quality, especially when conditioning on a limited number of reference views, introducing additional target views can be highly beneficial. These supplementary target views can either be randomly defined or duplicates with the identical target camera poses. Simultaneously generating multiple target views serves to implicitly reduce the inherent stochasticity in the diffusion process, resulting in improved generation quality and consistency. Through empirical investigations, we determine that the optimal configuration ensures a minimum of 15 target views, as highlighted in orange in Fig. 8. Beyond this threshold, any additional views yield marginal performance improvements.

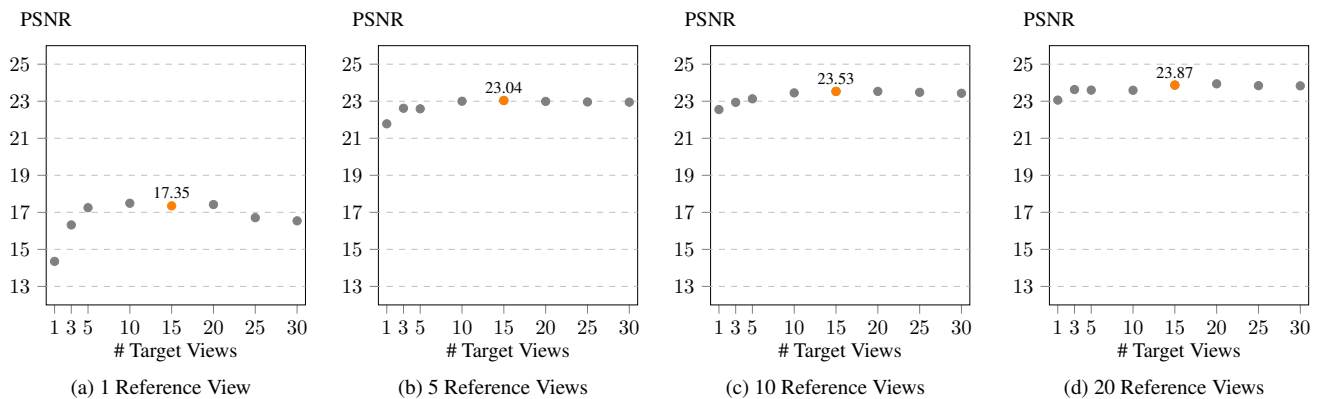


Figure 8. **Novel view synthesis with a different number of reference and target views.** We present the averaged performance of EscherNet on *one* pre-selected target view across objects in the GSO dataset. We observe a clear improvement in view synthesis quality as the number of both reference and target views increases. In this scenario, the multiple target views are essentially multiple duplicates of the initially chosen single pre-selected view, a strategy we find effective in enhancing view synthesis quality.

**Training Data Sampling Strategy** We have explored various combinations of  $N \in \{1, 2, 3, 4, 5\}$  reference views and  $M \in \{1, 2, 3, 4, 5\}$  target views during EscherNet training. Empirically, a larger number of views demand more GPU memory and slow down training speed, while a smaller number of views may restrict the model’s ability to learn multi-view correspondences. To balance training efficiency and performance, we set our training views to  $N = 3$  reference views and  $M = 3$  target views for each object, a configuration that has proven effective in practice. Additionally, we adopt a random sampling approach with replacement for these 6 views, introducing the possibility of repeated images in the training views. This sampling strategy has demonstrated a slight improvement in performance compared to sampling without replacement.

**Scaling with Multi-view Video** EscherNet’s flexibility sets it apart from other multi-view diffusion models [22, 23] that require a set of fixed-view rendered images from 3D datasets for training. EscherNet can efficiently construct training samples using just a pair of posed images. While it can benefit from large-scale 3D datasets like [6, 7], EscherNet’s adaptability extends to a broader range of posed image sources, including those directly derived from videos. Scaling EscherNet to accommodate multiple data sources is an important direction for future research.