Sparse auto-regressive modeling for scene generation from multi-view images

Thomas Lucas[1⋆], Maxime Pietrantoni[1⋆], Philippe Weinzaepfel[1], Wonjune Cho[2], Bardienus Pieter Duisterhof[3], Vincent Leroy[1], and Jerome Revaud[1]

[1] NAVER LABS Europe, France
[2] NAVER LABS, South Korea
[3] CMU, USA

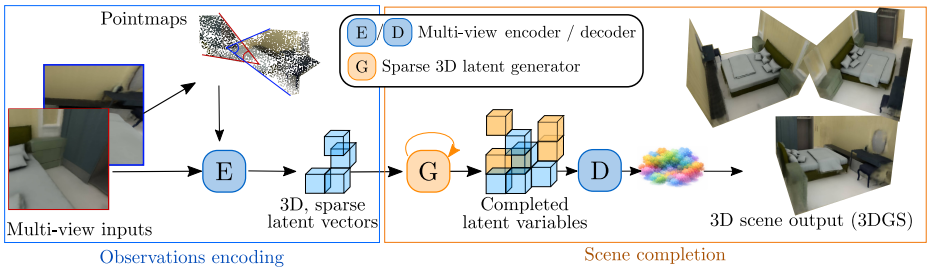

Fig. 1: Overview of SPAR3S. An encoder extracts voxel-aligned conditioning 3D latent tokens (in blue) from sparse (*e.g.* 2) input views. This encoder is trained together with a decoder that outputs 3D Gaussians that can be rendered via splatting. The partially filled 3D latent voxel grids are iteratively expanded by a 3D masked auto-regressive transformer that jointly predicts the occupancy of the remaining voxel grids, and their corresponding latent tokens (in yellow). The generated latent tokens can be decoded into 3D Gaussians using the scene decoder.

Abstract. Generating complete 3D scenes from sparse, unconstrained views is a fundamental challenge in 3D vision which requires reasoning beyond observed content while remaining computationally tractable. Existing feed-forward reconstruction methods are inherently limited to content visible in the input images, while 3D generative modeling is hindered by the high computational cost of dense volumetric representations and the scarcity of large-scale 3D supervision. We introduce SPAR3S, a sparse voxel-aligned 3D latent generative model for conditional scene completion without requiring ground-truth 3D data for supervision. Our key insight is to formulate 3D scene generation in a structured, compact, voxel-aligned 3D latent space where only occupied voxels are represented. We learn this sparse latent space directly from multi-view images using photometric supervision via differentiable 3D Gaussian Splatting. Given a partial set of observed voxels encoded from sparse input views, scene completion reduces to predicting the missing latent tokens and their spatial support within the voxel grid. To this end, we train a masked

⋆ Equal contribution.

autoregressive transformer that jointly models voxel occupancy and latent token values, enabling efficient and spatially consistent generation of unseen regions. We demonstrate the effectiveness of our method on synthetic indoor scenes, achieving higher novel-view quality than prior work. We further validate its generalization on RealEstate10k, highlighting its applicability to real-world data.

1 Introduction

Recent advances in novel view synthesis, *e.g.* 3D Gaussian Splatting (3DGS) [19], have enabled real-time, photorealistic rendering of 3D scenes, optimized through differentiable photometric supervision. In parallel, feed-forward multi-view stereo and pointmap regression methods such as DUSt3R [52] have demonstrated that 3D geometry can be recovered directly from images without iterative optimization. Combined, these paradigms lead to *pixel-aligned* 3D Gaussian regression from sparse views [6,9,56]. However, pixel-aligned predictions are fundamentally limited to the content visible in the input images.

To complete scenes, a recent line of work combines multi-view diffusion models to generate 2D novel content with pixel-aligned Gaussian Splats regression further lifted in 3D [2,16,46]. Such approaches require known cameras to generate new content or strong assumptions about them, *e.g.* with object-centric tasks or 2D panoramic image generation. This leads to a chicken-and-egg problem, where sampling plausible cameras requires knowing the scene and vice-versa. Additionally, multi-view diffusion models lack geometric consistency across novel views because of a lack of explicit 3D representation. By contrast, generating content directly in 3D does not require a known set of novel camera poses beforehand, and inherently enforces geometric consistency while naturally handling unconstrained camera positions. Therefore, in this work we aim to generate unseen content in the form of 3DGS representations directly in 3D from sparse unconstrained input views.

Generating complete 3D scenes from sparse, unconstrained views requires reasoning beyond observations, and modeling the distribution of plausible unseen geometry and appearance. To do so, we propose to lift the problem into a *voxel-aligned 3D latent space*. Concretely, a first model learns to map multi-view observations of a scene to voxel-aligned 3D latent variables. Then, a generative model is trained to learn a distribution over the 3D latent space; when parts of a 3D scene are not observed from a set of multi-view inputs, corresponding voxels can be sampled from the generative model to complete the 3D scene. The completed latent representation is then decoded into a 3DGS representation, see Fig. 1 for an overview. Designing such generative models in 3D is challenging for two main reasons. First, dense volumetric representations scale cubically with spatial resolution, making high-resolution generative modeling computationally prohibitive. Second, large-scale curated 3D supervision is scarce unlike with images or text, limiting the applicability of fully-supervised 3D generative learning.

Our key idea to address scalability is to only materialize occupied voxels in our structured voxel-aligned latent space. Our encoder-decoder architecture maps arbitrary sets of multi-view input images to a *sparse* and *compact* voxel-based 3D latent representation, which is decoded into a 3DGS scene with voxel-aligned Gaussians. To model a distribution over such latent representations, we introduce a *sparse autoregressive transformer* operating over sequences of 3D tokens. Scene completion thus reduces to predicting both voxel occupancy and latent token features. By exploiting sparsity, spatial compression, and structured 3D token orderings, our approach is efficient while preserving volumetric coherence. To address raw 3D data scarcity, our sparse latent space is learned directly from multi-view images using *photometric supervision*, via differentiable 3D Gaussian Splatting. Importantly, this learning process does not require any ground-truth 3D supervision.

We validate our proposed 3D generative paradigm, called SPAR3S on synthetic indoor scenes, where it achieves improved novel-view synthesis quality compared to prior feed-forward 3DGS regression methods as well as 3D generative methods. We further demonstrate generalization to real-world data on RealEstate10K [63].

Our contributions may be summarized as follows:

- We introduce a sparse voxel-aligned 3D latent space learned from multi-view photometric supervision (Section 3).
- We propose an occupancy-aware masked autoregressive transformer for conditional scene completion which operates in this latent space (Section 4).
- We demonstrate improved novel-view synthesis quality over prior feed-forward 3DGS generative and deterministic approaches, with validation on both synthetic and real-world datasets (Section 5).

2 Related work

Observed 3D scene reconstruction aims to regress observed scene geometry from multi-view inputs. The resulting scene representations only cover the parts of the scene that are observed in the input images. We rely on such partial representations, but additionally aim to complete unobserved parts of the scene with a generative model and capture visual content together with geometry. Recent methods for scene reconstruction like DUSt3R [52] and MASt3R [24] represent a scene by predicting a dense per-pixel 3D point map $P(u, v) \in \mathbb{R}^3$ mapping each image pixel directly to its 3D coordinate in camera space. We use such pointmaps, associated to a set of multi-view inputs, to initialize a sparse voxel aligned latent space and remove voxels that do not contain anything when encoding targets. Conveniently, these methods have been extended to accommodate more than 2 views, either by introducing some scene-level memory [4, 49, 51] or by scaling attention layers to a larger number of tokens from multi-views [50].

Feed-Forward 3D Gaussian splatting. Feed-forward 3D Gaussians models have been introduced to reconstruct a scene from a sparse number of input

views [6,9,33,44,45,47,56,57,60]. The base principle lies in regressing 3D Gaussians parameters from a pixel-aligned feature map [45]. PixelSplat [6] predicts 3D Gaussians parameters from a pair of images by relying on epipolar cues while MVSplat [9] aggregates multi-view information in a cost volume before predicting Gaussians, implicitly learning to match and triangulate. To overcome the ambiguity of reconstructing texture-less regions, [33,44,56] further introduce monocular depth priors as regularization. Instead of using such priors, [47,57,60] directly predict a set of 3D Gaussians from images using a large transformer model, at the cost of a significant increase in training budget. These methods lack structured 3D representations, which limits geometric consistency across views as well as their ability to handle very sparse inputs and extreme viewpoint variations. In contrast, our model is designed around a latent sparse 3D space to handle these difficult cases. Furthermore, standard feed-forward 3DGS approaches are deterministic and fail to reconstruct ambiguous regions that may be occluded and not observed in the input views. We now present several distinct lines of work that introduce generative modeling to address this challenge.

Multi-view diffusion over image space. The strong generative priors contained in image and video diffusion models have been used to generate novel views conditioned on camera poses [13, 28, 34, 62]. To improve geometric consistency of the generated images, some approaches [8, 32, 59] directly condition the denoising process on strong 3D scene priors [52]. In contrast to our task, these methods do not produce an explicit 3D scene representation as output, but rather sets of novel views. Thus, generated content cannot be rendered in real-time in 3D, and the procedure needs to be re-run for each additional set of views. Another drawback is that there is no guaranteed spatial coherence between views, inducing flickering and geometric drift.

Iterative 3DGS optimization using diffusion priors. In [27, 48] a Score Distillation Sampling loss is introduced to leverage 2D diffusion model priors and directly optimize a 3DGS scene representation. In addition, [58] initialize the Gaussians with a diffusion model to obtain a rough initial geometry which facilitate downstream convergence. Any scene reconstruction is therefore based on iterative optimization and takes a significant amount of time.

2D generation with 3DGS prediction. Given a set of posed conditioning images and a set of novel camera poses, this class of methods outputs a set of 3DGS parameters for each novel camera pose by using 2D generative models. In the context of object-centric reconstruction, [35, 55] finetune a StableDiffusion model [42] to output Gaussian parameters and [2,46] optimize a 3DGS prediction head on top of a pretrained controlled video diffusion model. The head predicts pixel-wise 3DGS parameters for each novel view sampled from the diffusion model. In [16, 30] a 2D encoder-decoder is trained to encode images in a pixel-aligned latent space that can be further decoded into pixel-aligned Gaussians. In a second stage they optimize a multi-view diffusion model in this latent space. Such approaches require *known camera* as input to sample novel content in 2D, or alternatively need to make simplifying assumptions such as object-centric targets or 2D panoramic scenes to easily sample cameras. In contrast, we tackle

the more general problem of generating content directly in 3D, without requiring known camera poses or making simplifying assumptions about the scene.

3D Generative feed-forward 3DGS. The following methods directly apply diffusion in 3D. In [38] a 3D diffusion model is optimized to sample 3DGS, but requires a teacher model and ground-truth splats for training; this limits scalability as such data requires dense sets of views and a slow optimization procedure per scene. Closer to our work, [53] adopts a VAE formulation with variational Gaussians that encode uncertainty in a latent space, and these can be rendered and decoded in 2D to allow for easy photometric optimization. [29] instead applies diffusion in a 3D latent space to model the conditional distribution of latents that can be decoded to 3D Gaussian splats based on conditioning views. In [10] novel images are sampled with diffusion conditioned on latents rendered from a unified 3D representations. These 3D diffusion approaches are used to sample a fixed set of Gaussian parameters which limits the extent of the generated scene and scalability. In contrast, we sample latents from a sparse voxel grid and apply an autoregressive model which provides greater flexibility and representational power. More similar to our approach, SCube [41] and XCube [40] both sample occupancy grids in a voxelized latent space, but with generative formulations that do not encode appearance and geometry jointly in a latent space.

Autoregressive latent modeling. Our approach is based on auto-regressively predicting sparse 3D latent tokens, and borrows from autoregressive models primarily developed for images. In particular, discrete latent approaches such as VQ-VAE [37] and its hierarchical extension VQ-VAE-2 [39] represent images using compact codebooks and train autoregressive priors over these quantized latents. We follow the same overall paradigm, but adapt it to 3D. Recently, masked and autoregressive [5, 25, 26]. transformers have emerged as flexible alternatives to raster-order PixelCNN-style [36] models: they allow inference over tokens in arbitrary order. We extend this flexibility to sparse 3D voxel grids, leveraging arbitrary token orderings to operate directly on sparse voxel representations. Other applications of auto-regressive models to 3D include [23], which uses a multiview 2D transformer, and [7] adapts it to object-centric generation. In contrast, we directly operate in a 3D voxel-aligned latent space which enforces geometric consistency across views and does not require simplifying assumptions about camera distribution.

3 3D latent representation from multi-view data

Our scene encoder-decoder model follows the general framework of auto-encoders: it encodes input views into a latent representation, and is trained to reconstruct these inputs by decoding the latent variables. However, in our case the inputs and outputs are only a *proxy* to the quantity that we seek to model, which is 3D scenes represented via 3DGS. To achieve this, the flow of information in our encoder-decoder is *constrained to go through a single 3DGS representation per scene*, from which all input views should be reconstructed. This constraint can be seen as a type of bottleneck in the general auto-encoder framework as all the

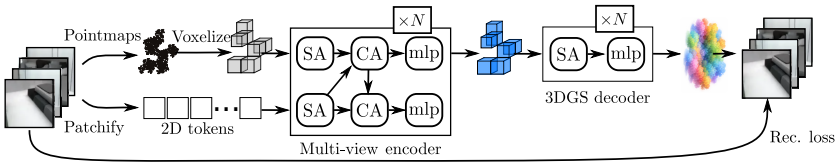

Fig. 2: Learning a 3D latent space from multi-view input data. Multi-view inputs are patchified and used to initialize voxel-aligned tokens from pointmaps. A bidirectional cross-attention block updates 2D and 3D tokens. The final 3D latent representation is decoded into a 3D scene represented as a set of Gaussian splats. The model is trained to reconstruct the inputs via differentiable rendering. No explicit 3D scene ground truth is required, yet this produces 3D representations.

information in the inputs has to be compressed into a single 3D representation. Additionally, we require that this 3DGS representation should be computed from a *single, voxel aligned* 3D latent representation. With this construction it is possible to generate scene content by first generating a 3D latent representation, then decoding it with the scene decoder.

Let $\mathcal{I}=\{I_i\}_i$ be a set of multi-view input RGB images and E_θ an encoder that maps multi-view inputs into a latent 3D representation $z_{3\text{D}}$. Also let D_ϕ a decoder that maps such a 3D latent variable to a 3DGS scene representation denoted G, and let $\mathcal{R}$ a rendering operation that reconstructs $\hat{\mathcal{I}}$ from G via differentiable rendering. Our scene encoder-decoder can be summarized as:

$$Z_{3\text{D}} = E_\theta(\mathcal{I}), \quad G = D_\phi(Z_{3\text{D}}), \quad \hat{\mathcal{I}} = \mathcal{R}(G). \tag{1}$$

E_θ and D_ϕ are optimized with a photometric reconstruction loss; see Fig.2 for an overview. The two 3D representations $Z_{3\text{D}}$ and G are produced as a byproduct of this auto-encoding problem, yet modeling these quantities is the real purpose of our latent generative model.

Learning a sparse compact 3D latent space. In practice, processing a full 3D token grid is computationally prohibitive unless voxel resolution is very low. Hence, most of the key design choices made for our model aim at addressing this bottleneck. First, we design our auto-encoder such that it operates on a sparse 3D representation: only tokens that correspond to voxels *with content* (ie. non empty) will be processed by the scene encoder. Second, we build a hierarchical encoder-decoder such that the latent space is more compact than the scene. The encoder progressively downsamples 3D voxels until the bottleneck. The decoder then upsamples this latent space back to the original voxel resolution. Both of these aspects drastically alleviates the computational load and allow us to derive a model operating fully in 3D.

Sparse 3D voxel initialization. Our encoder model processes sets of multi-view input RGB images $\mathcal{I}=\{I_i\}_i$. First, pointmaps [52] $\{P_i\}_i$ are estimated from a scene with *unconstrained* viewpoints, here using [11]. The union of pointmaps $\mathcal{P} = \bigcup_i P_i$ is rescaled and translated to fit a predefined volume $\mathcal{V}$ with a transformation T, such that $T(\mathcal{P}) \subseteq \mathcal{V} = [0,1]^3$. Then 3D latent queries $z^0_{3\text{D}}$ are

initialized using the pointmaps to filter out empty voxels; the 2D multi-view inputs are patchified and tokenized into 2D tokens denoted z^0_{2D}.

Sparse attention. A stack of L bidirectional cross-attention blocks, denoted CA and composed of a self-attention layer, a cross-attention layer and an MLP, is used to update both 3D and 2D tokens:

$$\left(z^{k+1}_{\text{3D}}, z^{k+1}_{\text{2D}}\right) = \left(CA(z^k_{\text{3D}}, z^k_{\text{2D}}), CA(z^{k+1}_{\text{3D}}, z^k_{\text{2D}})\right). \tag{2}$$

The 2D tokens are then discarded, and z^L_{3D} is decoded with a transformer model D_ϕ, into a set of M 3D Gaussians per occupied voxels. Denoting V_{occ} the set of occupied voxels and G the set of parameters of one 3D Gaussian splat, we have:

$$\{G_i\}_{i=1}^{M\times|V_{\text{occ}}|} = D_\phi(Z). \tag{3}$$

Throughout these computations, empty voxels are never considered.

Token resampling and sparsification. The 3D voxel tokens are downsampled through the layers of the encoder into the bottleneck latent space and upsampled through the layers of the decoder. When downsampling, tokens within a local neighborhood are aggregated. When upsampling, a mechanism to maintain spatial sparsity is needed. We use a binary cross-entropy (BCE↑) head over upsampled voxel positions to predict voxel occupancy before computing the upsampled token values. During training, the ground-truth (GT) occupancy mask is substituted, augmented with false-positive noise injection to improve the model's robustness to classification errors at inference time. This can be formalized as:

$$z^{\text{up}}_{\text{3D}} = \text{Upsample}\left(z_{\text{3D}}, \mathcal{M}\right) \quad \text{where} \quad \mathcal{M} = \begin{cases} \text{BCE}\uparrow(z_{\text{3D}}) & \text{at inference} \\ \text{GT} \cup \epsilon_{\text{FP}} & \text{at training} \end{cases} \tag{4}$$

where ϵ_{FP} represents the sampled false-positive noise distribution.

Geometry-guided attention. To facilitate learning, attention maps in the CA blocks incorporate geometry-aware correspondence information, derived from the pointmap P projections onto patches and a bincount between patches and voxels. Attention logits are computed as a weighted average between learned attention logits from voxel v to patch p, and bincount attention scores:

$$A^{\text{guided}}_{v,p} = \alpha \cdot \sigma(A^{\text{learned}}_{v,:})_p + (1-\alpha) \cdot \sigma(A^{bincount}_{v,:})_p, \tag{5}$$

where $\sigma(\cdot)$ denotes a softmax and $\alpha \in [0,1]$.

Photometric supervision. Input images are reconstructed by projecting $D_\phi(Z)$ onto corresponding cameras, using camera parameters C obtained together with the pointmaps P, with a differentiable Gaussian splatting operation $\mathcal{R}$ [20]. The model is trained end-to-end with an L_2 reconstruction loss. Following [42], we sample Z using the reparametrization trick [22] and regularize the latent space of the encoder E_θ using a Kullback–Leibler divergence term with a low coefficient and the total loss is thus:

$$\mathcal{L}(\theta,\phi) = \frac{1}{|\mathcal{I}|}\sum_{i=1}^{|\mathcal{I}|}\left\| I_i - \mathcal{R}\Big(D_\phi\big(E_\theta(I_i,P_i)\big), C_i\Big)\right\|_2^2 + \beta\,\mathcal{L}_{\text{KL}}(\theta). \tag{6}$$

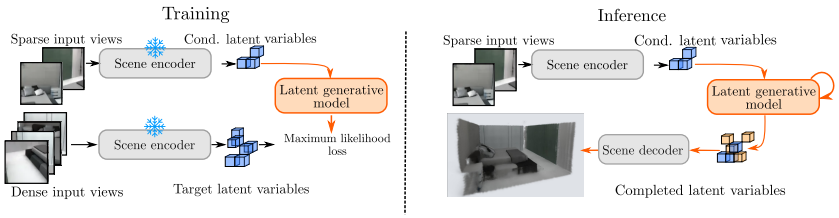

Fig. 3: Learning a latent generative model. For training, a dense set of multi-view inputs is aggregated into a 3D latent representation used at target for the generative model. This target is predicted conditionally on an other latent representation, encoded from sparse observed views. Inference starts from the observed latent variables which are iteratively expanded to complete the scene in latent space before decoding it to 3DGS.

The use of a probabilistic framework accounts for the fact that multiple sets of Gaussians can yield the same projections, and provides a well behaved latent space.

Latent targets and conditioning. We use E_θ in two capacities: given a dense set of input views, $Z_{\text{target}} = E_\theta(\mathcal{I}_{\text{dense}})$ can provide targets for the 3D latent generative model. Given a single view or a small set of views $\mathcal{I}_{\text{sparse}}$, $Z_{\text{cond}} = E_\theta(\mathcal{I}_{\text{sparse}})$ provides model conditioning. Thus, E_θ is trained to work with variable input set sizes.

4 Sparse Auto-Regressive Modeling in 3D Latent Space

The encoder-decoder predicts 3D Gaussians covering only the observed parts of the scene. To generate geometry and scene content in unobserved parts, we introduce our sparse 3D latent generative model, see Fig. 3 for an overview.

Background: masked auto-regression. Our generative model is built on Masked Auto-regressive models [5, 25, 26]. Let $\mathbf{z} = (z_1, \dots z_n)$, auto-regressive models decompose $P(\mathbf{z})$ using the chain rule of probabilities: $P(\mathbf{z}) = \prod_{i=1}^{N} P(z_i | z_1, \dots z_{i-1})$. Masked auto-regressive models in particular can predict the chain in any order, and predict any group of variables from any other. They achieve this by training the model to take random visible sets of input and predicting the rest. To model the output distribution, a popular choice is a categorical distribution over discretized values; alternatively, [26] proposed a token-wise diffusion loss to model continuous values.

Predicting occupancy and latent values. Scene completion in a voxel space is a challenging task as sequences of unknown voxels can be of arbitrary length and voxels may be occupied or empty, making sequences spatially sparse. We thus propose a formulation where a masked auto-regressive model predicts both *where* the 3D latent vectors through occupancy prediction, and their *content* through latent prediction. Concretely, our model H_ψ takes as input a sequence of conditioning voxel tokens from the encoder-decoder, a sequence of unknown voxel positions and predicts their occupancy as well as their latent values if they

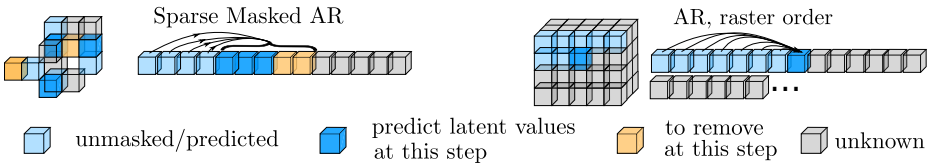

Fig. 4: Sparse 3D latent completion. Sparse masked auto-regression is applied by jointly predicting latent token values as well as occupancies for a random observed subset of tokens. By contrast, raster-order auto-regressive models consider all tokens sequentially in the possible volume, which does not scale well.

are occupied. It is composed of two bidirectional attention-based blocks (H^1_ψ, H^2_ψ) and two lightweight voxel-wise heads: an occupancy head h_{occ} and a denoising diffusion head h_ϵ [26]. The first block aggregates information from conditioning and observed latent vectors while for the second block, masked voxel embeddings $\mathbf{z}_\text{m}$ are added to the aggregated context tokens, and completed by H^2_ψ, inspired by the MAE architecture [15]. Finally, occupancy $\mathbf{o}$ and latent vector $\mathbf{z}$ are predicted independently per masked voxel using the prediction heads.

Training procedure. Training our generative model does not require any 3D ground-truth information and simply requires supervision in the form of target tokens generated from encoding dense views through our encoder. To ease notations in what follows, we assume that these target tokens $\mathbf{z}$ are obtained by encoding a dense set of multi-view images $\mathbf{z} = E_\theta(\mathcal{I}_\text{target})$. An important point is that target sequences can be *incomplete*. Indeed, using a dense set of multi-view inputs $\mathcal{I}_\text{target}$ does not *guarantee* that the entire scene will be observed. Thus at training, our *target* token positions are separated into three categories: $\mathcal{F}$ the set of occupied positions in $\mathbf{z}$, $\mathcal{E}$ the set of positions observed to be empty, and $\mathcal{U}$ the set of unobserved positions. Their disjoint union covers the full volume:

$$\mathcal{V} = \mathcal{F} \sqcup \mathcal{E} \sqcup \mathcal{U}. \tag{7}$$

In full generality, masked auto-regressive models are trained by selecting a mask, *i.e.* a *random* subset $\mathcal{O}$ of observed tokens *from the target*, and predicting the unobserved *target* positions $\mathcal{T} = \mathcal{V} \setminus \mathcal{O}$. In our setting, we exploit the mask in one additional way: at training, we use the mask to *hide* unobserved positions in $\mathcal{U}$, *i.e.* we constrain positions in $\mathcal{U}$ to always be in $\mathcal{T}$. They are thus predicted by the network; however, they will be masked from the loss, as the target is unknown. In addition, our model also observes a sparse set of inputs through their encoded representation, denoted $\mathbf{z}^c = E_\theta(\mathcal{I}_\text{cond.})$; this observation is provided to the model by adding tokens in $\mathbf{z}^c$ to its inputs when their corresponding positions are not included in $\mathbf{z}_\mathcal{O}$. See Fig.4 for an overview. Denoting $p_\mathcal{T}$ the target positions, a training forward pass can be summarized as:

$$\hat{\mathbf{o}}_\mathcal{T},\ \hat{\mathbf{z}}_\mathcal{T} = H_\psi(\mathbf{z}_\mathcal{O}, \mathbf{z}^c, p_\mathcal{T})\,, \tag{8}$$

with $\mathcal{O} \cap \mathcal{T} = \varnothing$, $\mathcal{O} \cup \mathcal{T} \subset \mathcal{V}$, $\mathcal{U} \subset \mathcal{T}$, and $\mathcal{U}$ masked from the loss.

As cameras are randomly placed, we assume that the unknown set $\mathcal{U}$ is random and thus the model will learn to cover full latent representations on average.

Inference. At inference, positions are grouped into disjoint subsets $\mathcal{T}_0 \sqcup \cdots \sqcup \mathcal{T}_n = \mathcal{V}$ and the model is run iteratively, where predictions on $\mathcal{T}_k$ are added to the conditioning set before predicting $\mathcal{T}_{k+1}$. Given conditioning information $\mathbf{z}^c$ we select a subset $\mathcal{T}_1$ of masked target voxels; H_ψ outputs occupancy scores and latent predictions. If the occupancy score is above a threshold τ, the latent prediction is added to the set of observed voxels; otherwise it is discarded. This is repeated iteratively by selecting a new subset $\mathcal{T}_i$ of masked target voxels at each iteration. Thus, the model refines the scene geometry while predicting scene content.

This construction can work with any random ordering. We opt for a region growing approach, where subsets are selected around the previous set until all positions are exhausted. We perform mask selection (*i.e.* autoregressive order) by building a sparse symmetric k-NN graph with observed voxels $\mathbf{z}^o$ taken as seeds, from which a breadth-first search (BFS) assigns a depth level $l(v)$ to each voxel (length of the shortest path from a seed to v). Candidate voxels are then partitioned into depth levels and the resulting joint latent probability is:

$$P(\mathbf{z}) = \prod_{l=1}^{\max_v(l(v))} P\left(\mathbf{z}_{\{l(v)=l\}} | \mathbf{z}_{\{l(v)<l\}}\right). \tag{9}$$

This ordering promotes spatial consistency at inference.

Training. The occupancy head is optimized through a masked binary cross entropy loss between $\mathbf{o}_{\text{target}}$ and $\mathbf{o}_{\text{pred}}$: $\mathcal{L}_{\text{occ}} = BCE(\hat{\mathbf{o}}_{\mathcal{T}}, \mathbf{o}_{\mathcal{T}})$. Inspired by [26], we parametrize the latent prediction head as a tokenwise denoising network and optimize it with a tokenwise diffusion loss, effectively learning to sample plausible unobserved tokens conditioned on observed tokens. It is based on a DDPM [18] schedule with an epsilon formulation, leading to the following loss: $\mathcal{L}_{\text{diff}} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I),t} \|\epsilon - h_\epsilon(Z^{\text{mask}}_{\text{target}} | f^2_t, t)\|^2_2$, where ϵ denotes the corruption noise and t us the timestep in the noise schedule. We apply this loss on the set of unobserved target positions $\mathcal{T}$. In parallel, we introduce a second diffusion head to refine conditioning tokens based on all decoded information $\mathbf{z}^c$, it is optimized with the same $\mathcal{L}_{\text{diff}}$ loss but applied on conditioning tokens only. As this task is much more constrained than the main novel token prediction task, we apply a *stopgrad* operation before the conditioning token refinement head. We keep the cost of this tokenwise diffusion part at minimum (a few token-wise MLP layers) to maintain scalability, but empirically observe an improvement compared to a fully auto-regressive formulation and keep it by default.

Hierarchical occupancy prediction To further improve efficiency, we propose to perform occupancy prediction in a hierarchical coarse-to-fine manner. In that case both stages share the masked auto-regressive overall architecture and principles defined earlier, but differ in the following way: the coarse stage performs a dense coarse occupancy prediction from all the voxels in the scene while the fine stage takes as input this coarse set of occupied voxels and predicts a refined set of occupancies. The coarse model is used with a *high recall* occupancy threshold at inference while the fine model is used with a threshold chosen for a good precision-recall tradeoff, thus optimizing compute and accuracy.

Table 1: Novel view synthesis evaluation with *two* conditioning views (very wide baseline) at resolution 224×224 on the 3DFront and RealEstate10k datasets.

Category	Method	3DFront				RealEstate10k			
		FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓
Reconstruction	3DGS [20]	260	8.65	0.14	0.79	271	7.92	0.12	0.79
	PixelSplat [6]	165	9.07	0.18	0.67	155	13.73	0.41	0.50
	DepthSplat [56]	110	13.76	0.49	0.55	82	13.25	0.46	0.41
Generation	DiffusioNeRF [54]	229	12.80	0.19	0.73	-	-	-	-
	MVSplat360 [10]	111	9.72	0.35	0.70	66	15.40	0.49	0.40
	LatentSplat [53]	180	13.92	0.33	0.61	53	16.09	0.48	0.39
	SPAR3S (ours)	**59**	**15.18**	**0.62**	**0.50**	**41**	**16.72**	**0.57**	**0.36**

5 Experiments

We first present training details and evaluation protocol (Sec. 5.1), which primarily consists in evaluating scene reconstruction from unconstrained cameras, and sparse sets of conditioning views, *e.g.* 2 views. In this context where with 2 cameras randomly sampled, input views often have no overlap, typically making baselines struggle, we show the benefits of SPAR3S through quantitative evaluations and qualitative examples (Sec. 5.3). Finally we provide a comprehensive set of ablations to justify our design choices (Sec. 5.4).

5.1 Experimental details

Datasets. We train and evaluate our models on a synthetic indoor dataset generated with 3DFront [12], as well as the real-world RealEstate10K dataset [63]. Our synthetic dataset, referred to as 3DFront, comprises over 10000 indoor scenes covering bedrooms, living rooms, dining rooms, and libraries, with diverse layouts, furniture, and textures. For each scene, cameras are first distributed randomly. A subset of cameras that maximizes overall scene coverage is then selected as the final scene viewpoints. In the sparse-view setting, this naturally reduces the overlap between viewpoints, making novel-view synthesis significantly more challenging. We use ground-truth camera poses, and initial pointmaps are obtained by backprojecting pixels using rendered depth maps.

To assess our method's ability to handle real data without ground-truth camera information, we also evaluate on RealEstate10K, which consists of over 80k real estate video clips, predominantly captured indoors. We estimate camera poses, intrinsics, and initial pointmaps using MASt3R-SfM [11]. Owing to their video origin, the camera paths in RealEstate10K follow smooth trajectories and produce densely sampled views with substantial overlap, making the novel-view synthesis task comparatively easier.

Training details. Our models are trained and evaluated at a resolution of 224x224. Training is done from scratch on a single A100 GPU; training both the scene encoder-decoder and the latent prediction model takes approximately

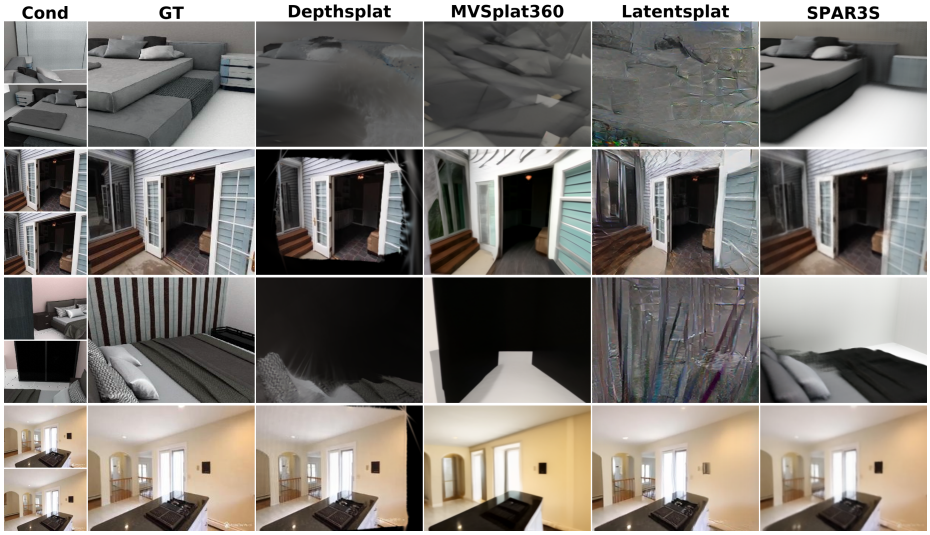

Fig. 5: Visual comparison of novel rendered views from 2 conditioning images (left column). Overall, our approach handles conditioning views with little overlap and large novel view extrapolation (lines 1 and 3) while other 3D generative approaches fail. On easier setups (lines 2 and 4), our model still provides better scene completions.

4 days each in that setting. Given a training scene, we randomly sample 4 conditioning views and 12 target views. Masking ratio in the target set of voxels is uniformly sampled in $[0.5, 1]$. We refer to the supplementary material for further training details.

Evaluation protocol. We evaluate the ability to reconstruct scene geometry and appearance by performing novel view synthesis from a sparse set of observed images. We report standard image-level reconstruction metrics—PSNR and SSIM—along with the perceptual metric LPIPS [61], all computed on rendered novel views. To further assess the visual quality and diversity, we measure FID [17] and KID [3] across datasets. The number of conditioning views varies among $2, 4, 8, 12, 16$ depending on the experiment, and we explicitly indicate the chosen setting for each evaluation. Given a scene, conditioning views are first randomly sampled, novel views are then randomly sampled among the remaining images. We deliberately adopt this random sampling strategy for both our evaluation datasets such that the splits span a wide range of difficulty, from easy novel views containing overlap with the conditioning set to challenging novel views that share little or no overlap (as opposed to 'easier' splits that may be found in the literature [6] that contain substantial overlaps).

Baselines. We primarily compare our approach against recent state-of-the-art feed-forward generative 3D Gaussian scene reconstruction methods: LatentSplat [53] which employs a variational latent representation within a VAE framework, enabling more reliable reconstruction outside the observed camera frustums as well as MVSplat360 [10] which samples images with a video diffusion

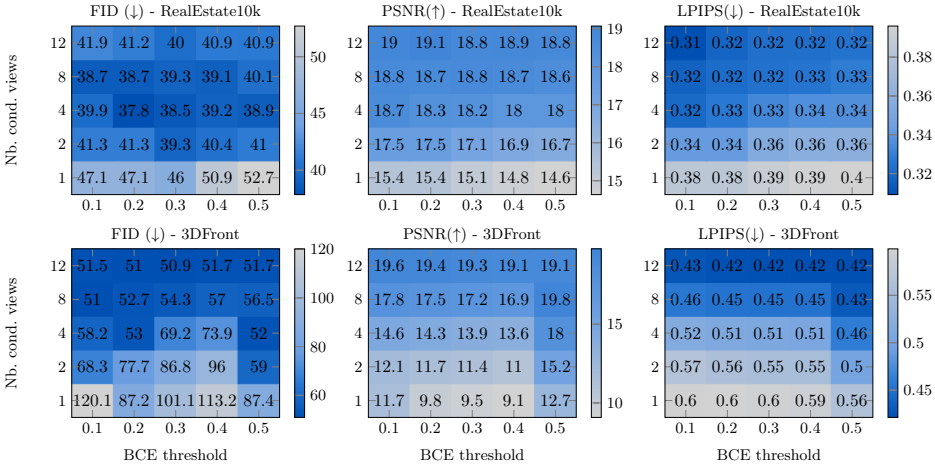

Fig. 6: Impact of the number of conditioning views and BCE occupancy thresholds. We study the impact on performance of varying the threshold used for binary occupancy prediction (in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$) for various numbers of conditioning views ($\{1, 2, 4, 8, 12\}$). Increasing the number of conditioning views consistently improves performance, while lower threshold tend to give gains. With PSNR and LPIPS we see a diagonal pattern: with more conditioning views, the threshold can be increased as geometry prediction gets more accurate.

model operating in a latent space spanned by rendering 3DGS latent vectors predicted from a feed-forward 3DGS model. We further include the 3D generative DiffusioNeRF [54] which leverages generative priors from diffusion models when reconstructing scenes with neural radiance fields. For the sake of completeness, we include the following pixel-aligned methods, PixelSplat [6], a well-established model that regresses 3D Gaussians along viewing rays from paired context images as well as Depthsplat [56] which leverages geometric priors resulting in more accurate reconstruction. Finally, vanilla 3DGS [20] is also included as a reference.

5.2 Two-view NVS

We first evaluate scene reconstruction from two conditioning views, a challenging setting in which most of the scene is unobserved and must be inferred from minimal context. Novel-view synthesis results on 3DFront and RealEstate10K are reported in Tab. 1. As expected, both optimization-based [20] and feed-forward 3DGS [6] methods fail to produce meaningful reconstructions, due to their limited ability to extrapolate beyond the frustums of conditioning views, and their difficulty in reproducing observed regions from novel views with significant viewpoint variation. Depthsplat [56], with geometric priors, is able to better handle large viewpoint changes but still inherently cannot reconstruct unobserved areas. LatentSplat [53] benefits from its generative formulation and shows improved reconstructions in novel areas. However, it remains constrained by its epipolar-based encoder architecture which prevents it from handling overly

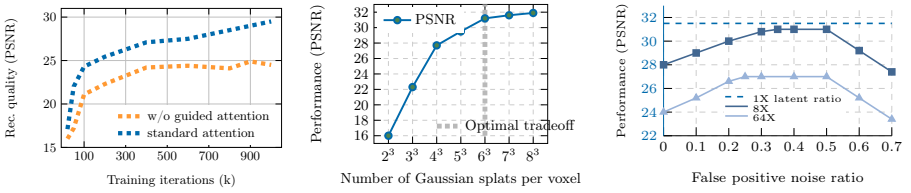

Fig. 7: Ablation studies. Left: Impact of guided attention on training dynamics — adding attention logits is beneficial. **Middle:** Impact of the number of splats per voxel — diminishing returns are observed; we use 6^3 splats per voxel. **Right:** Latent compression ratio and false positives in learned upsampling: approximately 25% of false positives is optimal.

wide viewpoint variations (as illustrated by the clear performance drop between Re10k and 3DFront). Similarly, MVSplat360 diffusion model [10] struggles to handle unconstrained cameras poses which leads to inconsistent reconstructions. In contrast, the combination of an autoregressive generative formulation and a sparse voxelized latent space allows SPAR3S to substantially outperform all 3D generative baselines on this challenging wide-two views scene reconstruction.

5.3 Multi-view NVS

In Fig. 6 we increase the amount of contextual information the model receives by increasing the number of conditioning views provided to the models, using $n = \{1, 2, 4, 8, 12\}$. The reconstruction task thus becomes more constrained, inducing easier scene completion in unobserved areas. This is indeed illustrated by the improving NVS metrics as the number of conditioning views increases. We also vary the threshold used in binary occupancy prediction. Overall, performance tends to be better with low thresholds: missing out parts of the room degrades metrics more than adding spurious parts. We notice a diagonal pattern: the model makes more accurate occupancy predictions with more views.

5.4 Ablations

Encoder-Decoder. We study the main design choices of the scene encoder-decoder. First, in Fig. 7 (left) we show the impact of removing the bincount-based guided attention mechanism, which results in slower training and lower performance. Second, we show in the middle plot the impact of the number of Gaussian regressed on PSNR. While the performance improves monotonically, there is a clear diminishing return and thus we chose a value of 6^3. Third, we show in the right plot that when spatial compression is used, injecting false positives at training is key to obtain optimal performance. Around 25% is a robust value for both $8\times$ and $64\times$ compression ratios. Finally, we evaluate different information bottlenecks on the encoded latent variables; KL coefficients are taken in $\beta \in \{0.001, 0.01, 0.1, 1\}$ and latent dimension in $\{16, 32, 64, 128\}$. A KL coefficient stronger than 0.1 or a latent dimension below 32 both significantly degrade performance; we pick the tighter possible bottleneck: $\beta = 0.1$ and $\dim(z) = 32$.

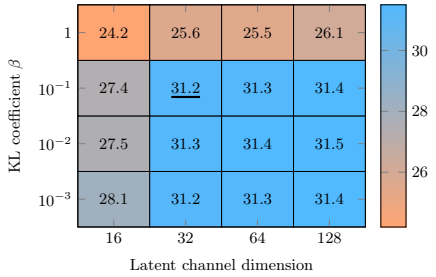

Fig. 8: Information bottlenecks on the latent variable. Impact measured with PSNR. A clear trend emerges: a KL coefficient stronger than 0.1 or a latent dimension below 32 both significantly degrade performance; otherwise reconstruction accuracy is stable over a wide range.

Table 2: Ablation study of the autoregressive diffusion model on the 3DFRONT dataset with four conditioning views. Each row removes one component of the full model. Metrics are reported on novel views.

Configuration	PSNR↑	SSIM↑	LPIPS↓
SPAR3S (full model)	**15.18**	**0.62**	**0.50**
Ablations (each removes one component)			
w/o diff.	13.90	0.53	0.54
w/o 3D RPE	13.74	0.49	0.58
w/o BFS ordering	14.81	0.59	0.53
w/o occ. reference	14.48	0.56	0.56
Additional Oracle Variant			
SPAR3S w/ gt. occ.	17.90	0.68	0.45

Autoregressive diffusion model. We ablate different components of the autoregressive diffusion model and report novel view rendering results in Tab. 2. Decoding latent vectors with a linear head (*wo. diff*) instead of a diffusion head degrades the model's ability to sample coherent tokens in unobserved areas. Similarly, injecting relative positional encoding in self-attention layers proves important for spatial reasoning when decoding the scene (*w/o. 3D RPE*). During inference, using a random autoregressive order (*w/o. BFS ordering*) and not performing occupancy refinement (*w/o. BFS ordering*) also degrades the consistency and quality of the predicted Gaussians. Finally, in order to evaluate the model's ability to reconstruct latent tokens without the influence of occupancy predictions, we decode the scenes using ground truth occupancies (*SPAR3S w. gt. occ.*). Unsurprisingly, this improves the quality of the results.

6 Conclusion

We introduce SPAR3S, a novel generative model for 3D scene reconstruction. SPAR3S processes sparse and unconstrained input views and encodes them into a 3D latent space associated with a voxel-aligned 3DGS encoder-decoder. SPAR3S further performs full scene completion in this 3D latent space by leveraging an efficient autoregressive architecture that jointly infers geometry and latent content to exploit scene sparsity. Our results demonstrate the potential of autoregressive latent modeling for 3D scene generation. As such, our work opens promising avenues for applying masked autoregressive models to 3D data and for advancing generative reconstruction in sparse-view scenarios.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Bahmani, S., Shen, T., Ren, J., Huang, J., Jiang, Y., Turki, H., Tagliasacchi, A., Lindell, D.B., Gojcic, Z., Fidler, S., et al.: Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. arXiv preprint arXiv:2509.19296 (2025)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD GANs. In: ICLR (2018)
4. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: Must3r: Multi-view network for stereo 3d reconstruction. In: CVPR (2025)
5. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: CVPR (2022)
6. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In: CVPR (2024)
7. Chen, J., Zhu, L., Hu, Z., Qian, S., Chen, Y., Wang, X., Lee, G.H.: Mar-3d: Progressive masked auto-regressor for high-resolution 3d generation. In: CVPR (2025)
8. Chen, W., Bi, J., Huang, Y., Zheng, W., Duan, Y.: Scenecompleter: Dense 3d scene completion for generative novel view synthesis. arXiv preprint arXiv:2506.10981 (2025)
9. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T., Cai, J.: MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In: ECCV (2024)
10. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In: NeurIPS (2024)
11. Duisterhof, B.P., Žust, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In: 3DV (2025)
12. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3d-front: 3d furnished rooms with layouts and semantics. In: ICCV (2021)
13. Gao, R., Holynski, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P., Barron, J.T., Poole, B.: Cat3D: Create anything in 3D with multi-view diffusion models. In: NeurIPS (2024)
14. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
15. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
16. Henderson, P., de Almeida, M., Ivanova, D., Anciukevičius, T.: Sampling 3D gaussian scenes in seconds with latent diffusion models. arXiv preprint arXiv:2406.13099 (2024)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics (ToG) (2023)

20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) (2023)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
23. Le, T., Pham, T., Nguyen, T., Kong, D., Xie, X., Mandt, S.: Umami: Unifying masked autoregressive models and deterministic rendering for view synthesis. In: NeurIPS (2025)
24. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: ECCV (2024)
25. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. In: CVPR (2023)
26. Li, T., Tian, Y., Li, H., Deng, M., He, K.: Autoregressive image generation without vector quantization. In: NeurIPS (2024)
27. Li, X., Wang, H., Tseng, K.: GaussianDiffusion: 3D gaussian splatting for denoising diffusion probabilistic models with structured noise. arXiv preprint arXiv:2311.11221 (2023)
28. Liang, H., Cao, J., Goel, V., Qian, G., Korolev, S., Terzopoulos, D., Plataniotis, K.N., Tulyakov, S., Ren, J.: Wonderland: Navigating 3d scenes from a single image. In: CVPR (2025)
29. Liao, Z., Sayed, M., Waslander, S.L., Vicente, S., Turmukhambetov, D., Firman, M.: Complete gaussian splats from a single image with denoising diffusion models. arXiv preprint arXiv:2508.21542 (2025)
30. Lin, C., Pan, P., Yang, B., Li, Z., Mu, Y.: Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. In: ICLR (2025)
31. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
32. Liu, F., Sun, W., Wang, H., Wang, Y., Sun, H., Ye, J., Zhang, J., Duan, Y.: Reconx: Reconstruct any scene from sparse views with video diffusion model. IEEE Transactions on Image Processing (2026)
33. Liu, Y., Fan, K., Yu, W., Li, C., Lu, H., Yuan, Y.: MonoSplat: Generalizable 3d gaussian splatting from monocular depth foundation models. In: CVPR (2025)
34. Ma, B., Gao, H., Deng, H., Luo, Z., Huang, T., Tang, L., Wang, X.: You see it, you got it: Learning 3d creation on pose-free videos at scale. In: CVPR (2025)
35. Meng, X., Wang, C., Lei, J., Daniilidis, K., Gu, J., Liu, L.: Zero-1-to-G: Taming pretrained 2d diffusion model for direct 3d generation. arXiv preprint arXiv:2501.05427 (2025)
36. van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., Kavukcuoglu, K.: Conditional image generation with pixelcnn decoders. In: NeurIPS (2016)
37. van den Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. In: NeurIPS (2017)
38. Peng, C., Sobol, I., Tomizuka, M., Keutzer, K., Xu, C., Litany, O.: A lesson in splats: Teacher-guided diffusion for 3d gaussian splats generation with 2d supervision. In: ICCV (2025)
39. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. In: NeurIPS (2019)
40. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: CVPR (2024)

41. Ren, X., Lu, Y., Liang, H., Wu, Z., Ling, H., Chen, M., Fidler, S., Williams, F., Huang, J.: Scube: Instant large-scale scene reconstruction using voxsplats. In: NeurIPS (2024)
42. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
43. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864 (2021)
44. Szymanowicz, S., Insafutdinov, E., Zheng, C., Campbell, D., Henriques, J.F., Rupprecht, C., Vedaldi, A.: Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In: 3DV (2025)
45. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3D reconstruction. In: CVPR (2024)
46. Szymanowicz, S., Zhang, J.Y., Srinivasan, P., Gao, R., Brussee, A., Holynski, A., Martin-Brualla, R., Barron, J.T., Henzler, P.: Bolt3d: Generating 3d scenes in seconds. In: ICCV (2025)
47. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: LGM: Large multi-view gaussian model for high-resolution 3d content creation. In: ECCV (2024)
48. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: DreamGaussian: Generative gaussian splatting for efficient 3D content creation. In: ICLR (2023)
49. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 3DV (2025)
50. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR (2025)
51. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: CVPR (2025)
52. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3d vision made easy. In: CVPR (2024)
53. Wewer, C., Raj, K., Ilg, E., Schiele, B., Lenssen, J.E.: latentSplat: Autoencoding variational gaussians for fast generalizable 3D reconstruction. In: ECCV (2024)
54. Wynn, J., Turmukhambetov, D.: DiffusioNeRF: Regularizing Neural Radiance Fields with Denoising Diffusion Models. In: CVPR (2023)
55. Xiang, T., Li, K., Long, C., Häne, C., Guo, P., Delp, S., Adeli, E., Fei-Fei, L.: Repurposing 2d diffusion models with gaussian atlas for 3d generation. In: ICCV (2025)
56. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: DepthSplat: Connecting gaussian splatting and depth. In: CVPR (2025)
57. Xu, Y., Shi, Z., Yifan, W., Chen, H., Yang, C., Peng, S., Shen, Y., Wetzstein, G.: GRM: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In: ECCV (2024)
58. Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3D gaussians by bridging 2D and 3D diffusion models. In: CVPR (2024)
59. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)
60. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: GS-LRM: Large reconstruction model for 3d gaussian splatting. In: ECCV (2024)
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
62. Zhou, J., Gao, H., Voleti, V., Vasishta, A., Yao, C.H., Boss, M., Torr, P., Rupprecht, C., Jampani, V.: Stable virtual camera: Generative view synthesis with diffusion models. In: ICCV (2025)

63. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. In: SIGGRAPH (2018)