TASE: Truncation-Aware Semantic Embeddings for 3D Scene Understanding and Editing

Tim-Felix Faasch[1], Jochen Kall[1], Lucas Nunes[2], Jens Behley[3], and Cyrill Stachniss[3,4]

[1] Bosch Research, Hildesheim, Germany
{tim-felix.faasch, jochen.kall}@de.bosch.com
[2] Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, Germany
lucas.nunes@rwth-aachen.de
[3] Center for Robotics, University of Bonn, Bonn, Germany
{jens.behley, cyrill.stachniss}@igg.uni-bonn.de
[4] Lamarr Institute for Machine Learning and Artificial Intelligence, Bonn, Germany

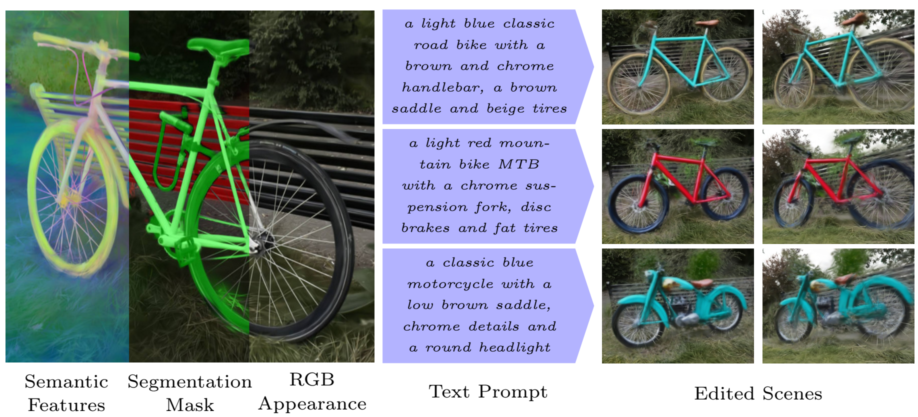


Fig. 1: We integrate truncation-aware semantic embeddings (TASE) to 3DGS, enabling semantic segmentation and text-guided edits. The figure shows the original scene, segmentation, and features (left), and the resulting localized edits (right), where only the specified object (bicycle) is modified.

Abstract. High-fidelity semantic 3D scene representations are crucial for numerous applications, including robotics, autonomous driving, and simulation. Beyond this, the ability to edit such representations enables developers to adapt these applications more easily to specific target scenarios. Current approaches provide limited support for controllable editing. We introduce TASE, a method that projects pretrained 2D semantic features into a truncation-aware embedding space to enable flexible 3D scene editing. Our method explicitly optimizes a feature space in which progressively reducing feature channels yields increasingly abstract semantic representations, while retaining more channels preserves fine-grained detail. Additionally, we improve multi-view consistency of the features using a scale- and translation-consistency loss. The resulting truncation-aware embedding space enables text-driven edits to 3D scenes, providing explicit control over how strongly edits adhere to the original scene content and allowing more substantial modifications than prior methods. Moreover, we propose a finetuning stage for the editing image generation model to mitigate artifacts caused by geometric changes. Experimental results demonstrate competitive performance in 3D scene editing, substantially outperforming prior methods on edits involving large geometric modifications.

1 Introduction

Accurate and efficient 3D scene representations, like 3D Gaussian Splatting (3DGS), are increasingly important for robotics, autonomous driving, and simulation [13, 42, 54, 56, 64, 69]. Many of these applications require the ability to efficiently modify scenes, e.g., to alter objects or to simulate weather and lighting changes [27,60]. However, current 3D representations provide limited support for structured, controllable editing. We argue that a semantic scene representation is key to efficiently enabling such edits. Recent advances in large-scale pretrained 2D feature extractors [28, 39] and image generation models [20, 29] suggest that semantically meaningful embeddings can serve as a powerful interface for controllable scene manipulation.

Incorporating semantic information into 3DGS [15, 51] has emerged as an active area of research. The integration of semantics facilitates downstream tasks such as segmentation of elements in the scene [18, 65] and editing of the scene appearance and geometry [33, 60]. Some existing works rely on discrete class or instance logits, which are inherently limited by their coarse, fixed vocabulary [60,66,67]. Open-vocabulary vision encoders such as CLIP [34] and DINO [28,39] provide rich semantic embeddings beyond discrete class logits. Recent works integrate such features into neural scene representations and 3DGS [8, 14, 17, 32, 33, 38, 52]. However, directly lifting 2D embeddings to 3D remains challenging: high feature dimensionality increases memory and compute costs [14, 32, 52], absolute positional biases hinder cross-view consistency [58, 59, 62], and existing approaches lack a principled mechanism to control semantic abstraction, i.e., the level of semantic specificity ranging from coarse class-level structure to fine-grained instance and texture detail [44].

Controllable semantic abstraction is beneficial for scene editing since it allows to define, how closely changes should be tied to the original scene content. Current 3D editing approaches still struggle with edits requiring substantial geometric change, because their edits remain too closely tied to the original scene [3, 50, 61]. Compression based approaches such as quantization [8, 14, 38, 52] or autoencoding [32] address memory, but do not directly allow for controllable abstraction. By contrast, channel-ordered representations, such as principle component analysis (PCA), appear to be effective for controlling abstraction in 2D image generation [44], yet remain under-explored for 3D scene representations.

The main contribution of this paper is a novel method that projects pretrained 2D semantic features to truncation-aware semantic embeddings (TASE) that are free from 2D positional bias. By imposing a channel-ordered structure, TASE enable controllable semantic abstraction: retaining more channels preserves fine-grained detail, while truncation of channels yields increasingly abstract representations. Removing 2D positional cues from the embedding improves cross-view consistency when features are fused in 3D. TASE support text-driven 3D scene editing via a ControlNet [63], and 3D semantic segmentation. We additionally propose using a finetuning strategy for editing image generation models to mitigate artifacts introduced by large geometry changes in the 3D scene. Our approach requires no 3D or multi-view data during training. See Fig. 1 for examples.

2 Related Work

3D Scene Representations: Explicit 3D data structures that can be used in rasterization pipelines, like meshes and point clouds, have been the standard representations for a long time [1]. Neural Radiance Fields (NeRF) [25] have been introduced as an implicit, learning-based representation that can be optimized to match posed multi-view images. This led to their widespread use in 3D reconstruction [2], simulation [42], generation [31, 37] and editing [10]. 3D Gaussian Splatting (3DGS) [15] has emerged as a faster alternative to NeRFs. It models the radiance field using 3D multivariate Gaussians and renders images via splatting, enabling significantly faster optimization and interactive rendering. Its explicit representation also facilitates editing operations such as composition and geometric transformations (e.g., scaling, translation, rotation). Consequently, 3DGS has been widely adopted for scene generation [4,9,41] and editing [3,24,45,50,61]. Similarly, we employ 3DGS as our 3D scene representation, extending it to incorporate semantic information.

3D Scene Editing: 3D scene editing is the task of changing the appearance and the geometry of an existing 3D scene to reflect a given control signal like a text prompt. Existing approaches usually leverage the guidance from an image generation model to change the parameters of a scene representation, either using score distillation sampling [16, 30, 55, 70] or pseudo views [3, 10, 21, 43, 45, 46, 48, 50, 61]. Most of these methods use current RGB [3, 10, 21, 45, 46, 48, 61] or depth [50] renderings as input to the image generation model. Even though effective, this limits the amount of geometric change that can be introduced by these methods. Distinctly, our method uses semantic features to condition the editing image generation model instead, allowing for complex edits including large changes in geometry.

Open Vocabulary Semantic Features: Weakly- and self-supervised learning for open vocabulary feature extraction has gained a lot of interest for downstream tasks such as classification, detection, or segmentation, especially where training data is sparse [11]. CLIP [34] popularized the use of a contrastive loss to create a joint embedding space for text and images. The approach has been extended to dense image features by using masked self distillation [6]. Another line of work, including DINO, uses pretraining on visual data only to produce both dense and global deep features [28, 39, 68].

A number of recent works have proposed to include pretrained image features into scene reconstruction to aid with scene understanding, allowing to segment and specifically target objects within the scene [8, 14, 32, 52, 57]. Due to the high dimensionality of the semantic features, most of these methods use compression based approaches such as quantization [8, 14, 38, 52] or a per-scene autoencoder (AE) [32]. This reduces the required memory, but does not allow for semantic abstraction. Some works investigate channel-ordered latent representations, in which earlier channels capture more important information than later ones [19, 35]. Nested dropout [35] encourages such structure by stochastically truncating later channels of the latent vector during training. Matryoshka representation learning [19] extends this idea by jointly optimizing for a fixed set of truncation levels, improving stability and robustness across different embedding sizes.

Despite their effectiveness, 2D positional cues produced by pretrained 2D feature extractors hinder their direct applicability in 3D, especially in a per-scene optimization setting [57, 58, 62]. Existing solutions either suppress these cues during scene optimization [57] or apply fine-tuning of the backbone [58, 62].

Following previous work, our method uses pretrained image features as the basis for the semantic embeddings. We use an AE trained with Matryoshka representation learning to simultaneously reduce the memory required by the features, and to enable controllable semantic abstraction. We additionally employ a scale- and translation-consistency loss to remove 2D positional bias in the latent space.

Controllable Image Generation: Latent diffusion and flow matching models have recently emerged as the primary techniques for high-fidelity image and video generation [20, 22, 36, 40]. While diffusion models progressively remove Gaussian noise, flow matching models [22] regress continuous vector fields, with both typically operating in the latent space of a variational autoencoder (VAE) to improve efficiency. During generation, the process can be conditioned on text prompts, enabling text-to-image synthesis. Some methods additionally allow to condition the generation on spatial control modalities such as edge maps, depth maps, or segmentation maps [26, 63]. ControlNet [63] does this by creating a copy of the generative model, that is trained to process the additional control modality, while the weights of the original model are kept frozen. Similarly, we use a ControlNet style network conditioned on our truncation-aware embedding space to generate edited views of the 3D scene.

We propose leveraging positional-bias-free, truncation-aware semantic embeddings (TASE) for 3D scene editing. TASE are multi-view consistent and enable controllable levels of semantic abstraction. Our pipeline supports complex text-driven edits of 3D scenes, allowing for substantial modifications of the scene appearance and geometry.

3 Preliminaries on 3D Gaussian Splatting

In 3DGS [15], the radiance field of a scene is represented as a set of N multivariate Gaussians $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$. Each Gaussian $\mathcal{G}_i$ is parametrized by a mean value $\boldsymbol{\mu}_i \in \mathbb{R}^3$ and a covariance matrix $\mathbf{\Sigma_i} \in \mathbb{R}^{3\times3}$. The value of the i^{th} Gaussian $\mathcal{G}_i$ at the position $\mathbf{x} \in \mathbb{R}^3$ is given as:

$$\mathcal{G}_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right). \tag{1}$$

During rendering, the Gaussians are projected to image space, leading to a 2-dimensional covariance $\mathbf{\Sigma}'_i \in \mathbb{R}^{2\times2}$ given the viewing transformation matrix $\mathbf{W} \in \mathbb{R}^{3\times3}$ and its Jacobian $\mathbf{J} \in \mathbb{R}^{2\times3}$ [71]:

$$\mathbf{\Sigma}'_i = \mathbf{J}\mathbf{W}\mathbf{\Sigma}_i\mathbf{W}^\top\mathbf{J}^\top. \tag{2}$$

The covariance matrix $\mathbf{\Sigma}_i$, being orthogonal, can be decomposed into a rotation $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ and a scaling matrix $\mathbf{S}_i \in \mathbb{R}^{3\times3}$ with:

$$\mathbf{\Sigma}_i = \mathbf{R}_i\mathbf{S}_i\mathbf{S}_i^\top\mathbf{R}_i^\top. \tag{3}$$

In practice, the covariance is parametrized through a rotation quaternion $\mathbf{q}_i \in \mathbb{R}^4$ and a scale vector $\mathbf{s}_i \in \mathbb{R}^3$ with $\mathbf{s}_i = \text{diag}(\mathbf{S}_i)$ which ensures orthogonality and reduces the number of parameters. The contribution of a single Gaussian $\mathcal{G}_i$ towards the overall radiance of the scene is defined by its opacity α_i and a set of spherical harmonic (SH) coefficients $\hat{\mathbf{c}}_{0,i}, ..., \hat{\mathbf{c}}_{M,i}$, which are multiplied with the value of the corresponding SH basis function $H_k(\mathbf{d})$ of the order k to arrive at a color $\mathbf{c}_i$ depending on the view direction $\mathbf{d} \in \mathbb{R}^3$. The final pixel color $\mathbf{c}_{\text{pixel}}$ is composed by z-ordered alpha blending the contribution $\mathbf{c}_i$ of all Gaussians $\mathcal{G}_i$ in $\mathcal{G}$:

$$\mathbf{c}_{\text{pixel}} = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \tag{4}$$

with:

$$T_i = \prod_{j=1}^{i-1}(1-\alpha_j) \quad \mathbf{c}_i = \sum_{k=0}^{M} \hat{\mathbf{c}}_{k,i}\, H_k(\mathbf{d}) \tag{5}$$

where M is the maximum SH order used for view-dependent color modeling. Thus each Gaussian $\mathcal{G}_i$ is parametrized by $\Theta_i = (\boldsymbol{\mu}_i, \mathbf{q}_i, \mathbf{s}_i, \alpha_i, \hat{\mathbf{c}}_{0\ldots M,i})$.

4 Our Approach to 3D Scene Editing

The task of 3D scene editing can be defined as follows: given a pretrained set of 3D Gaussians that represent a scene, and a control signal such as a text prompt, the goal is to modify the parameters of the Gaussians such that the rendered images of the modified scene reflect the control signal. To this end, we leverage truncation-aware semantic embeddings (TASE) which are detailed

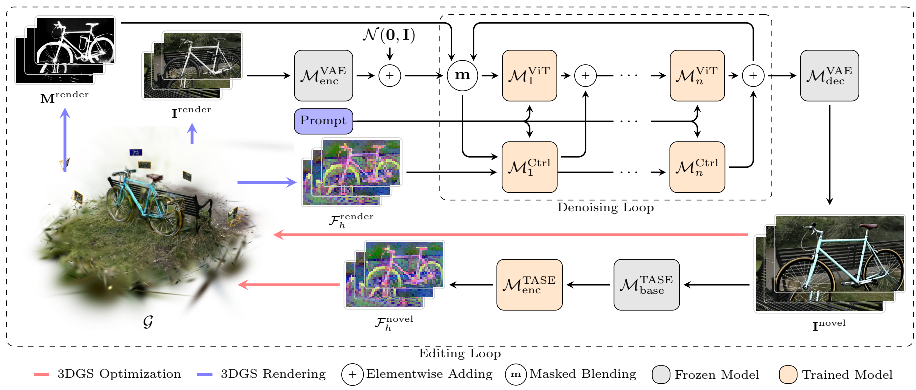


Fig. 2: Overview of the proposed pipeline: Rendered semantic feature maps $\mathcal{F}_h^{\text{render}}$ and a text prompt condition the image generation model ($\mathcal{M}^{\text{ViT}}, \mathcal{M}^{\text{Ctrl}}$) for the generation of novel views $\mathbf{I}^{\text{novel}}$. We add some noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to the latent of the current RGB render and use it as input to $\mathcal{M}^{\text{ViT}}$ and $\mathcal{M}^{\text{Ctrl}}$. For local edits, the rendered segmentation masks $\mathbf{M}^{\text{render}}$ are used as inpainting masks. From $\mathbf{I}^{\text{novel}}$, semantic features $\mathcal{F}_h^{\text{novel}}$ are extracted and optimized into the 3D Gaussian scene $\mathcal{G}$, jointly with $\mathbf{I}^{\text{novel}}$.

in Sec. 4.1. We integrate TASE into a 3DGS reconstruction of the scene as described in Sec. 4.2 to facilitate downstream segmentation and editing. We edit the 3D scene using a ControlNet conditioned on TASE, which is described in Sec. 4.4. We optimize the parameters of the Gaussians to match novel views generated by the ControlNet as detailed in Sec. 4.5. For local object edits we segment the editing target in 3D space using a simple, yet effective, similarity-based algorithm, which is described in Sec. 4.3. An overview illustration of our approach is depicted Fig. 2.

4.1 Truncation-Aware Semantic Embeddings

In order to condition the editing image generation model, we need semantic features that are multi-view consistent and that enable controllable semantic abstraction. To generate our truncation-aware semantic embeddings (TASE), we start with a pretrained image feature extraction backbone $\mathcal{M}_{\text{base}}^{\text{TASE}}$, which predicts a patch-wise feature map $\mathcal{F}_o \in \mathbb{R}^{H \times W \times N_o}$ from an image $\mathbf{I}$. We then train a symmetrical AE model with an encoder $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and a decoder $\mathcal{M}_{\text{dec}}^{\text{TASE}}$, which takes $\mathcal{F}_o$ as input. The hidden representation $\mathcal{F}_h = \mathcal{M}_{\text{enc}}^{\text{TASE}}(\mathcal{F}_o)$ will act as our projected TASE space, where $\mathcal{F}_h \in \mathbb{R}^{H \times W \times N_h}$. $\mathcal{F}_h$ has the same spatial resolution $H \times W$ as $\mathcal{F}_o$, but a lower number of channels $N_h < N_o$ to create the desired compression. We jointly train $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and $\mathcal{M}_{\text{dec}}^{\text{TASE}}$ while keeping $\mathcal{M}_{\text{base}}^{\text{TASE}}$ frozen. The reconstruction loss $\mathcal{L}_r$ between $\mathcal{F}_o$ and the reconstructed features $\mathcal{F}_r = \mathcal{M}_{\text{dec}}^{\text{TASE}}(\mathcal{F}_h)$ is defined as:

$$\mathcal{L}_r(\mathcal{F}_o, \mathcal{F}_r) = \frac{1}{HW} \sum_{x=1}^{W} \sum_{y=1}^{H} \ell_r\left(\mathbf{f}_o^{(x,y)}, \mathbf{f}_r^{(x,y)}\right), \tag{6}$$

where the loss ℓ_r between individual feature vectors $\mathbf{f}_o$ and $\mathbf{f}_r$ at patch location (x, y) is defined as:

$$\ell_r(\mathbf{f}_o, \mathbf{f}_r) = \lambda_{\cos}\left(1 - \frac{\mathbf{f}_o^\top \mathbf{f}_r}{\|\mathbf{f}_o\|\|\mathbf{f}_r\|}\right) + \lambda_{\mathrm{MSE}}\|\mathbf{f}_o - \mathbf{f}_r\|_2^2, \tag{7}$$

with the hyperparameters $\lambda_{\cos} \in \mathbb{R}$ and $\lambda_{\mathrm{MSE}} \in \mathbb{R}$.

To make $\mathcal{F}_h$ truncation aware, we employ Matryoshka representation learning [19], masking suffix channels of the embedding vectors $\mathbf{f}_h \in \mathcal{F}_h$ with zeros. We define as set of truncation levels $\mathcal{T} = \{2^n\}$ for $n = 1, \dots, \log_2(N_h)$. For each training iteration we compute the mean loss across all levels $t \in \mathcal{T}$:

$$\tilde{\mathcal{L}}_r = \frac{1}{|\mathcal{T}|}\sum_{t \in \mathcal{T}} \mathcal{L}_r(\mathcal{F}_o, \mathcal{M}_{\mathrm{dec}}^{\mathrm{TASE}}(\mathtt{mask}(\mathcal{F}_h, t))), \tag{8}$$

where $\mathtt{mask}(\mathcal{F}_h, t)$ sets the last $N_h - t$ channels of each embedding vector $\mathbf{f}_h$ in $\mathcal{F}_h$ to zero. This implicitly encourages the model to learn a hierarchical feature space, and is explained in more detail in the supplementary material.

Taking inspiration from DVT [58], we remove 2D positional cues from the embeddings by defining an additional consistency loss $\mathcal{L}_{\mathrm{cons}}$ during training:

$$\mathcal{L}_{\mathrm{cons}} = \mathcal{L}_r(\mathtt{crop}_{\mathrm{cons}}(\mathcal{F}_h), \mathcal{F}_{h,\mathrm{crop}}), \tag{9}$$

with

$$\mathcal{F}_{h,\mathrm{crop}} = \mathcal{M}_{\mathrm{enc}}^{\mathrm{TASE}}(\mathcal{M}_{\mathrm{base}}^{\mathrm{TASE}}(\mathtt{crop}(\mathbf{I}))), \tag{10}$$

where $\mathtt{crop}(\mathbf{I})$ randomly crops a section of $\mathbf{I}$ and $\mathtt{crop}_{\mathrm{cons}}(\mathcal{F}_h)$ crops the corresponding region from $\mathcal{F}_h$. The total loss $\mathcal{L}_{\mathrm{TASE}}$ for training $\mathcal{M}_{\mathrm{enc}}^{\mathrm{TASE}}$ and $\mathcal{M}_{\mathrm{dec}}^{\mathrm{TASE}}$ is then defined as:

$$\mathcal{L}_{\mathrm{TASE}} = \tilde{\mathcal{L}}_r + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}, \tag{11}$$

with the hyperparameter $\lambda_{\mathrm{cons}} \in \mathbb{R}$.

4.2 Lifting of the Embeddings into 3DGS

To use TASE for 3D scene editing, we need to integrate the semantic features $\mathcal{F}_h$ into a 3DGS scene. To this end, append a randomly initialized feature vector $\mathbf{f}_i$ to the Gaussian parameters Θ_i. We pass each input image through $\mathcal{M}_{\mathrm{base}}^{\mathrm{TASE}}$ and $\mathcal{M}_{\mathrm{enc}}^{\mathrm{TASE}}$ to obtain $\mathcal{F}_h^{\mathrm{image}}$. We then optimize $\mathbf{f}_i$ alongside the other parameters in Θ_i to minimize the feature loss $\mathcal{L}_f$ between the rendered feature maps $\mathcal{F}_h^{\mathrm{rendered}}$ and $\mathcal{F}_h^{\mathrm{image}}$:

$$\mathcal{L}_f = \mathcal{L}_r(\mathcal{F}_h^{\mathrm{image}}, \mathcal{F}_h^{\mathrm{rendered}}). \tag{12}$$

The feature rendering used the same splatting-based rendering and alpha compositing as the RGB rendering, but with the feature vectors $\mathbf{f}_i$ instead of the RGB colors.

In addition, we use the photometric loss $\mathcal{L}_{\text{l1}}$ and the SSIM loss $\mathcal{L}_{\text{SSIM}}$ [47] used in standard 3DGS reconstruction [15]. Since $\mathcal{F}_h$ has a lower spatial resolution than the input images, we add a local smoothness loss $\mathcal{L}_s$ on the feature vectors of the k-nearest Gaussians $\texttt{kNN}(\mathcal{G}_i, \mathcal{G})$ to prevent aliasing effects:

$$\mathcal{L}_s = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \sum_{j \in \texttt{kNN}(\mathcal{G}_i, \mathcal{G})} \mathcal{L}_r(\mathbf{f}_i, \mathbf{f}_j). \tag{13}$$

We additionally introduce a regularization loss $\mathcal{L}_{\text{maha}}$. This loss penalizes the Mahalanobis distance [23] between each per-Gaussian feature vector $\mathbf{f}_i$ and the feature distribution computed from per-patch features from the training dataset used to train $\mathcal{M}_{\text{enc}}^{\text{TASE}}$. Intuitively, this constrains the Gaussian features to remain close to the feature statistics observed during training and thereby discourages splatting-induced artifacts that could otherwise degrade segmentation or editing performance. The overall loss for the reconstruction is then defined as:

$$\mathcal{L}_{\text{splat}} = \lambda_{\text{l1}} \mathcal{L}_{\text{l1}} + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}} + \lambda_f \mathcal{L}_f + \lambda_s \mathcal{L}_s + \lambda_{\text{maha}} \mathcal{L}_{\text{maha}}, \tag{14}$$

with the hyperparameters $\lambda_{\text{l1}}, \lambda_{\text{SSIM}}, \lambda_f, \lambda_s, \lambda_{\text{maha}} \in \mathbb{R}$.

4.3 Segmentation

For local edits, only a specific object within the scene should be changed and therefore needs to be segmented from the rest of the scene. To enable this, the user marks the object in one or multiple views. We then extract the median feature vector $\mathbf{f}_j^{\text{anchor}}$ of the j^{th} marked area in the respective rendered feature maps $\mathcal{F}_h^{\text{rendered}}$. These act as anchors to identify corresponding Gaussians in 3D space. We compute the cosine similarity between the feature vector of each Gaussian $\mathbf{f}_i$ and the anchor vectors $\mathbf{f}_k^{\text{anchor}}$ and assign a binary segmentation label δ_i to each Gaussian $\mathcal{G}_i$ based on a similarity threshold τ_s:

$$\delta_i = \bigvee_k \frac{\mathbf{f}_i^\top \mathbf{f}_k^{\text{anchor}}}{||\mathbf{f}_i|| \, ||\mathbf{f}_k^{\text{anchor}}||} > \tau_s. \tag{15}$$

To obtain a segmentation that fully covers the target object, but no unrelated Gaussians, we incorporate spatial consistency into the process. First, the segmentation labels δ_i are iteratively propagated from the selected Gaussians to neighboring Gaussians whose $\mathbf{f}_i$ is sufficiently similar. For this we can use a relaxed threshold $\tau_p < \tau_s$ without running the risk of creating false positives outside the target object. To remove isolated, potentially misclassified Gaussians, we perform a majority vote among their k nearest neighbors and reassign their label accordingly. Optionally, the user can provide negative anchors to explicitly exclude certain regions from the segmentation that might be semantically similar. Throughout, the semantic abstraction level can be controlled by truncating the channels of $\mathbf{f}_i$ and $\mathbf{f}_k^{\text{anchor}}$.

4.4 ControlNet Training

To perform 3D scene editing, we leverage a ControlNet-style [63] image generation model that is conditioned on the TASE $\mathcal{F}_h$. The ControlNet is based on a pretrained vision transformer (ViT) $\mathcal{M}^{\text{ViT}}$ and operates in the latent space of a pretrained VAE $\mathcal{M}^{\text{VAE}}$. We construct the control branch $\mathcal{M}^{\text{Ctrl}}$ by duplicating $\mathcal{M}^{\text{ViT}}$ and introducing zero-initialized residual connections that add the output of each transformer block $\mathcal{M}_i^{\text{Ctrl}}$ to the corresponding output of $\mathcal{M}_i^{\text{ViT}}$. For each image in a training dataset, we extract the semantic features $\mathcal{F}_h$ using $\mathcal{M}_{\text{base}}^{\text{TASE}}$ and $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and use them as input to $\mathcal{M}^{\text{Ctrl}}$.

To enable controllable semantic abstraction during image generation, we mask some of the channels of $\mathcal{F}_h$ with zeros, uniformly selecting one truncation level t from $\mathcal{T}$ for each training step. Since $\mathcal{F}_h$ is already closer in spatial dimensions and channel depth to the latent space of $\mathcal{M}^{\text{VAE}}$ than to RGB images, we omit passing $\mathcal{F}_h$ through $\mathcal{M}_{\text{enc}}^{\text{VAE}}$ and instead resize it to the exact dimensions of the image latents via bilinear interpolation. We train $\mathcal{M}^{\text{Ctrl}}$ using a flow matching loss [20], while keeping $\mathcal{M}^{\text{ViT}}, \mathcal{M}_{\text{base}}^{\text{TASE}}$, and $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ frozen.

Since significant geometry changes during editing can introduce artifacts in rendered views, we propose a finetuning strategy inspired by Difix3D+ [49] to mitigate them. We construct a small dataset of clean and corrupted image triplets, including corresponding feature maps $\mathcal{F}_h$ exhibiting the same artifacts. Using the AnySplat feed-forward splatting model [12], we generate 3DGS scenes from single-view images and obtain the feature vectors $\mathbf{f}_i$ by projecting $\mathcal{F}_h$ onto the Gaussians. Subsequently, we randomly perturb $\boldsymbol{\mu}_i, \mathbf{s}_i$, and $\mathbf{q}_i$ to create corrupted renders of the scene. To emulate the local editing scenario, we apply a stronger perturbation to a spherical region around a randomly selected splat. As in Difix3D+, we treat the corrupted image as a noisy image at an intermediate denoising timestep ($t_{\text{corrupted}} = 200$), pass it through $\mathcal{M}_{\text{enc}}^{\text{VAE}}$ and add additional noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$ corresponding to the timestep t_{added} to retrieve the noisy image latent. The assumed timestep t_{assumed} of the noisy latent then computes as:

$$t_{\text{assumed}} = t_{\text{corrupted}} + \frac{t_{\text{total}} - t_{\text{corrupted}}}{t_{\text{total}}} t_{\text{added}}, \tag{16}$$

with t_{total} being the total number of optimization steps used during training. We finetune both $\mathcal{M}^{\text{ViT}}$ and $\mathcal{M}^{\text{Ctrl}}$, again using a flow matching loss [20].

4.5 3D Scene Editing

3D scene editing can be done globally, modifying all the Gaussians $\mathcal{G}$ to change, for example, the weather or the lighting of the scene. For this, we start by sampling a set of camera poses from the training poses that were used to optimize the original scene. We then render feature maps $\mathcal{F}_h^{\text{render}} \in \mathbb{R}^{H \times W \times N_h}$ from these poses and generate a set of novel views $\mathbf{I}^{\text{novel}}$ using the ControlNet ($\mathcal{M}^{\text{ViT}}, \mathcal{M}^{\text{Ctrl}}$). We optimize $\mathcal{G}$ to reflect the content of $\mathbf{I}^{\text{novel}}$, using the loss $\mathcal{L}_{\text{splat}}$ defined in Eq. (14). Subsequently, we sample a new set of poses and repeat the process. The number of iterations of this editing loop is an editing hyperparameter that depends on the desired amount of change in the scene.

For the first iterations of the editing loop, we truncate $\mathcal{F}_h^{\text{render}}$ by masking $N_h - t$ of the N_h channels with zeros, to allow the image generation model to align more closely with the text prompt, rather than with the current scene content. Over the course of the edit, we increase the number of retained channels t. During later iterations, we also integrate the current RGB renders $\mathbf{I}^{\text{render}}$ into the image generation in the same way as during the finetuning stage of the ControlNet training described in Sec. 4.4. We create schedules for t, the noise level added to $\mathbf{I}^{\text{render}}$, and the learning rates of the geometry parameters $(\boldsymbol{\mu}, \mathbf{s}, \mathbf{q})$. These schedules are editing hyperparameters that also depend on the desired amount of change in the scene.

The 3D scene can also be edited locally, changing the appearance and geometry only of a specified object in the scene. To this end, we first select a subset of Gaussians $\mathcal{G}_S \subseteq \mathcal{G}$ with the approach described in Sec. 4.3. We then sample novel object centric camera poses focused on a bounding ellipsoid that we construct around $\mathcal{G}_S$. In addition to $\mathcal{F}_h$ we also render a segmentation mask $\mathbf{M}^{\text{render}}$ that we first dilate and then use as an inpainting mask for the image generation. For local edits, we also add regularization terms as well as weight decay for the opacity α_i and the base color $\hat{\mathbf{c}}_{0,i}$ to allow for the removal of existing geometry, mitigating over-densification and over-saturation.

5 Experimental Evaluation

Our experiments evaluate the effectiveness of truncation-aware semantic embeddings (TASE) for 3D scene editing. In Sec. 5.2, we show that our semantic embeddings enable controllable 3D scene editing, including substantial geometric modifications. In Sec. 5.3, we demonstrate how channel truncation can be used to control semantic abstraction during 2D image generation conditioned on TASE. Finally, in Sec. 5.4, we show the individual effects of: our specific method to create a truncation-aware embedding, of our consistency loss aimed at removing 2D positional bias, and of the Difix3D+ [49] finetuning strategy.

5.1 Implementation Details

We use a pretrained DINOv3 model [39] as the backbone $\mathcal{M}_{\text{base}}^{\text{TASE}}$, which has been shown to contain rich semantic information. For the encoder $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and the decoder $\mathcal{M}_{\text{dec}}^{\text{TASE}}$, we use a single transformer block. The hidden channel dimensionality N_h of the autoencoder is set to 64 and we use $\mathcal{T} = [2^1, 2^2, ..., 2^6]$ as the truncation levels for the Matryoshka representation learning training. The image generation model uses FLUX.1[dev] [20] as $\mathcal{M}^{\text{ViT}}$. We train both $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and the control branch $\mathcal{M}^{\text{Ctrl}}$ on ImageNet-1k [5] for a single epoch, using a learning rate of 10^{-4} for $\mathcal{M}_{\text{enc}}^{\text{TASE}}$ and $\mathcal{M}_{\text{dec}}^{\text{TASE}}$. For the ControlNet we use a cosine learning rate schedule with a maximum learning rate of 10^{-5}. We train $\mathcal{M}^{\text{ViT}}$ and $\mathcal{M}^{\text{Ctrl}}$ randomly using an empty prompt or a prompt constructed from the class labels

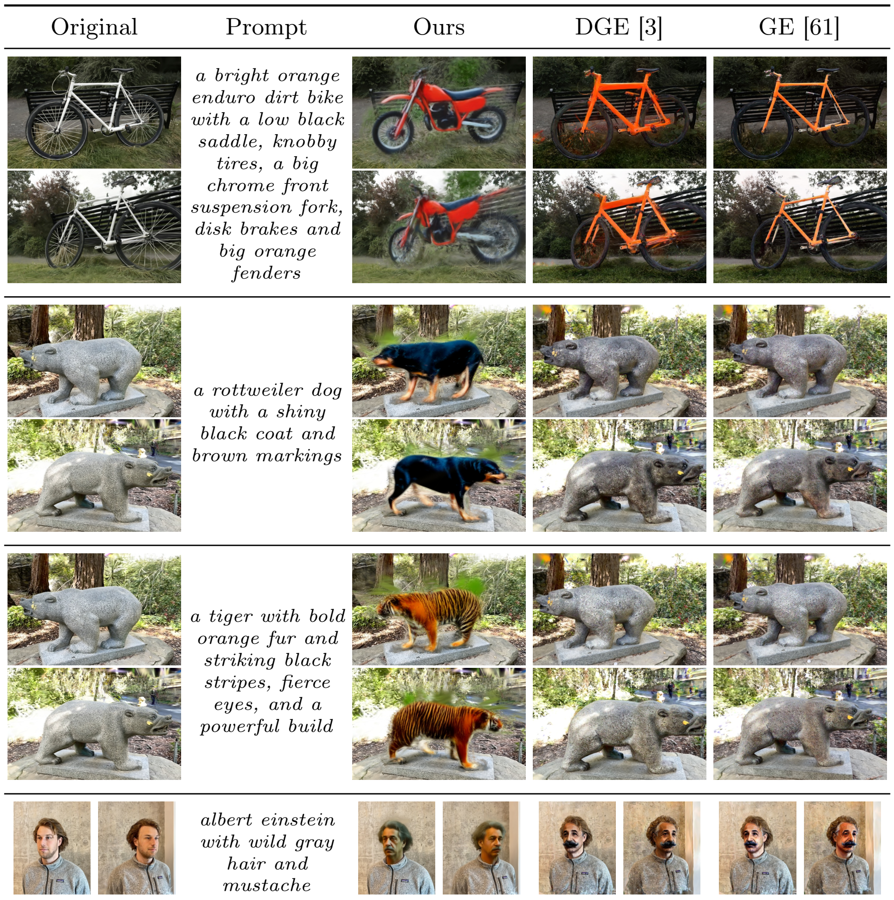


Fig. 3: Qualitative comparison of our editing results against the state-of-the-art 3D scene editing methods Direct Gaussian Editing (DGE) [3] and GaussianEditor (GE) [61] for local 3D scene editing.

associated with the image as text guidance. For the finetuning stage, we create 50,000 clean/corrupted image pairs from ImageNet-1k's train set and finetune for 4 epochs with a reduced maximum learning rate of 10^{-6} and 10^{-7} for $\mathcal{M}^{\mathrm{Ctrl}}$ and the $\mathcal{M}^{\mathrm{ViT}}$ respectively. More detail on the implementation, datasets and the used hyperparameters is provided in the supplement.

5.2 3D Scene Editing Capabilities

To evaluate our 3D scene editing capabilities, we perform diverse edits on multiple datasets. We compare against Direct Gaussian Editing (DGE) [3], GaussianEditor (GE) [61], EditSplat (ES) [21] and DreamCatalyst (DC) [16] as state-of-the-art baselines for 3DGS scene editing. While some methods use a pretrained 3D generation model to replace objects within the scene [45,53], this is not equiv-

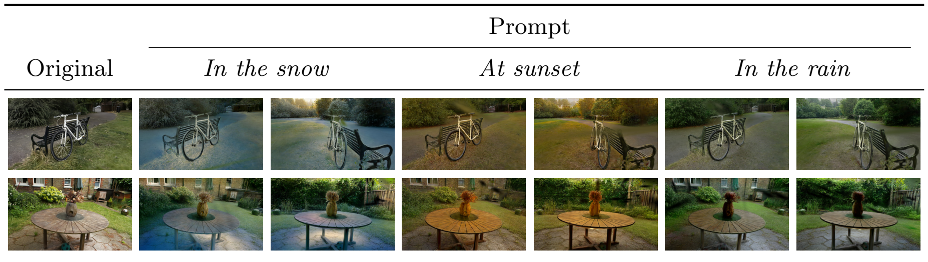


Fig. 4: Qualitative results of global 3D scene edits performed with our method showing that our method succeeds in globally changing the appearance of the scene.

Table 1: Quantitative comparison of our local scene editing results against state-of-the-art 3D scene editing methods (CLIP Directional Similarity).

	CLIP dir. sim. ↑	User Study	
		Geometry ↑	Appearance ↑
DreamCatalyst [16]	0.0985	-	-
EditSplat [21]	0.1492	-	-
GaussianEditor [61]	0.1179	16.7%	8.8%
DGE [3]	0.1247	19.1%	18.7%
Ours	**0.1542**	**64.3%**	**72.5%**

alent to manipulating the original object geometry and only changing it, where it is necessary. We therefore compare our results only to methods that change the contents of the original scene.

Fig. 3 qualitatively compares our editing results to GE [61] and DGE [3], which shows that our method is able to perform local edits with high visual fidelity and consistency across views. As seen, when large modifications to the geometry are required, e.g., changing the bicycle to a motorcycle, the baseline methods change the color of the object towards the target color to varying degrees, but fail to change the geometry sufficiently. When changing the bear into a different kind of animal, requiring a drastic change in appearance and geometry, the baselines fail to reflect the requested changes. In contrast, our approach arrives at a high quality result. This highlights the effectiveness of using TASE as a control signal, enabling substantial edits to 3D scenes. More qualitative results are provided in the supplement, including comparisons to ES [21] and DreamCatalyst [16].

To quantify the advantages of our method, in Tab. 1 we compute the CLIP dir. sim. [7] metric, which measures how well the edit direction aligns with the intent specified by the text prompt. Our method achieves the highest score, showing that our edits align much better with the provided text prompt. Additionally, we conduct a user study to assess the user preference of changes in geometry and appearance in the edited scenes in comparison to GE [61] and

(a) Original **(b)** 64 dims **(c)** 8 dims **(d)** 4 dims **(e)** 2 dims

Fig. 5: Generated images with the text prompt *"in the snow"* with TASE guidance truncated to different levels, showing close alignment to the semantic layout of the original scene for a large number of retained channels and close alignment to the text prompt for a low number of retained channels.

Table 2: Ablation study for the method of dimensionality reduction ($\mathcal{M}^{\text{TASE}}$ vs. PCA), the consistency loss $\mathcal{L}_{\text{cons}}$ and the finetuning stage (FT). The finetuning only affects the editing.

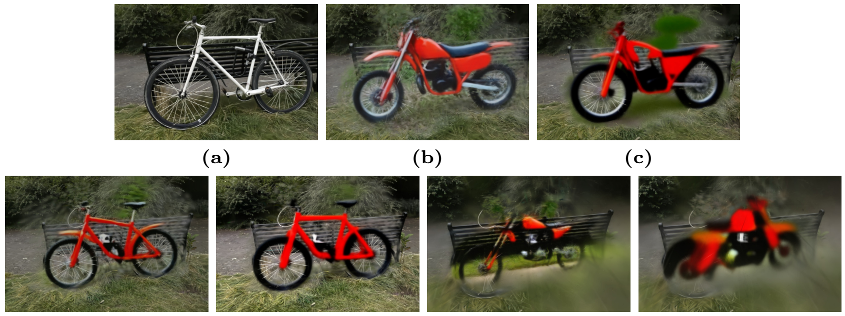
	Reconstruction			Editing
	PSNR ↑	SSIM ↑	$\mathcal{L}_f$ ↓	CLIP dir. sim. ↑
PCA	22.64	0.73	0.0879	0.1067
PCA + FT	—	—	—	0.1365
$\mathcal{M}^{\text{TASE}}$	23.89	0.76	0.1016	0.0852
$\mathcal{M}^{\text{TASE}}$ + FT	—	—	—	0.0905
$\mathcal{M}^{\text{TASE}} + \mathcal{L}_{\text{cons}}$	**24.64**	**0.78**	**0.0120**	0.1385
$\mathcal{M}^{\text{TASE}} + \mathcal{L}_{\text{cons}}$ + FT (Ours)	—	—	—	**0.1542**

DGE [3]. The details of the user study are described in the supplement. The results in Tab. 1 demonstrate that the edits performed with our method are preferred by users both in terms of geometry and in terms of appearance.

Fig. 4 presents qualitative results for global edits, where the entire scene is modified according to a text prompt. Our method is able to produce high-quality results that reflect the desired changes in the scene, such as seasonal, weather or lighting changes.

5.3 Trading off Specificity and Abstraction

When generating images conditioned on TASE, the number of retained channels controls the balance between prompt alignment and adherence to the scene structure. Using fewer dimensions yields more diverse images that follow the text prompt, while retaining the full embedding mostly preserves the semantic layout of the original image without reproducing it exactly. This behavior is illustrated in Fig. 5: with 2 dimensions, the generated image only loosely follows the original layout but strongly reflects the prompt *"in the snow"*. With 64 dimensions, the result preserves more of the scene structure in the source image, yet introduces little snow. Intermediate truncation levels provide a smooth transition between these extremes. Additional examples are provided in the supplement.



(a) (b) (c) (d) (e) (f) (g)

Fig. 6: Qualitative ablation, editing with the prompt: *a bright orange enduro dirt bike with a low black saddle, knobby tires, a big chrome front suspension fork, disk brakes and big orange fenders*: (6a): original scene, (6b): $\mathcal{M}^{\text{TASE}} + \mathcal{L}_{\text{cons}} + \text{FT}$ (Ours), (6c): $\mathcal{M}^{\text{TASE}} + \mathcal{L}_{\text{cons}}$, (6d): $\mathcal{M}^{\text{TASE}} + \text{FT}$, (6e): $\mathcal{M}^{\text{TASE}}$, (6f): $\text{PCA} + \text{FT}$, (6g): PCA.

5.4 Ablations

To assess the impact of the consistency loss $\mathcal{L}_{\text{cons}}$ described in Sec. 4.1 on editing quality, we compare our full model to a variant omitting $\mathcal{L}_{\text{cons}}$ during the training of $\mathcal{M}^{\text{TASE}}$. We further justify our truncation-aware embedding produced by $\mathcal{M}^{\text{TASE}}$ under a Matryoshka representation learning scheme (Sec. 4.1) by comparing it to a PCA of the base features $\mathcal{F}_o$ computed over 50k training samples. Finally, we evaluate all ablations before and after Difix3D+ [49] finetuning.

Tab. 2 shows the CLIP directional similarity (dir. sim.) [7] for the same local edits used for comparison to the baselines in Sec. 5.2, as well as the reconstruction metrics PSNR, SSIM [47] and the feature loss $\mathcal{L}_f$ from Eq. (12) for held out views of the original scene. The much lower $\mathcal{L}_f$ when using $\mathcal{L}_{\text{cons}}$ during the training of $\mathcal{M}^{\text{TASE}}$ indicates that $\mathcal{L}_{\text{cons}}$ is important for features that are free from 2D positional bias and therefore multi-view consistent. The large difference in the final editing quality indicates that this property is crucial when using semantic embeddings as a control signal for 3D scene editing.

Fig. 6 shows an editing example from this ablation study, more examples are shown in the supplement. When using PCA as the dimensionality reduction method (Fig. 6g, f), the edited scene often displays some of the visual elements requested in the prompt, resulting in a relatively high CLIP dir. sim. (Tab. 2), but does not converge to a sensible geometry. When $\mathcal{M}^{\text{TASE}}$ is trained without $\mathcal{L}_{\text{cons}}$, the geometry occasionally converges to a largely meaningful result, as illustrated in Fig. 6e, d. However, this behavior is not consistent. In many cases, the optimization fails entirely, as shown in the additional examples in the supplement. These frequent failures lead to the low average CLIP dir. sim. reported in Tab. 2. The Difix3D+ finetuning stage (Fig. 6b, d, f) can be observed to help the model to recover finer detail, independent of the used semantic embedding.

6 Conclusion

In this paper, we introduced TASE, a truncation-aware semantic embedding that adapts pretrained 2D features for controllable, text-driven 3D scene editing. By explicitly structuring the embedding such that channel truncation yields increasingly abstract semantics, TASE provides direct control over edit strength and adherence to the original scene content. We further improved cross-view feature consistency using a scale- and translation-consistency loss, enabling robust fusion in 3D, and proposed a finetuning stage for the editing image generation model to reduce artifacts introduced by large geometric modifications. Experiments show that our pipeline supports substantial geometry changes and achieves stronger editing quality than prior methods on challenging edits involving major geometric variations. Ablation studies validate the impact of each component, confirming the benefits of the consistency loss, the image generation finetuning stage, and our choice of an autoencoder-based truncation-aware embedding over PCA.

Limitations. Due to the diverse requirements of scene editing tasks, some applications may benefit from hyperparameter tuning, particularly for the 3DGS densification. As with other image-generation-based methods, stochastic sampling can introduce localized artifacts. Moreover, because semantics and appearance are not explicitly disentangled, color adjustments may not always match the prompt, and Gaussians may bleed into the background, causing floaters.

Acknowledgements. As part of the IPCEI ME/CT the project is supported by the Federal Ministry for Economic Affairs and Energy, by the Ministry for Economic Affairs, Labor and Tourism of Baden-Württemberg and financed by the European Union – NextGenerationEU.

References

1. Akenine-Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M., Hillaire, S.: Real-Time Rendering. CRC Press, 4 edn. (2018), see Chapter 2: The Graphics Rendering Pipeline
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
3. Chen, M., Laina, I., Vedaldi, A.: DGE: Direct Gaussian 3D Editing by Consistent Multi-view Editing. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2025)
4. Chen, S., Zhou, J., Jiang, Z., Zhang, T., Wu, Z., Hwang, J.N., Li, L.: ScalingGaussian: Enhancing 3D Content Creation with Generative Gaussian Splatting. arXiv preprint (2024), arXiv:2407.19035
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2009)
6. Dong, X., Bao, J., Zheng, Y., Zhang, T., Chen, D., Yang, H., Zeng, M., Zhang, W., Yuan, L., Chen, D.e.a.: Maskclip: Masked self-distillation advances contrastive language-image pretraining. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2023)

7. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. ACM Trans. on Graphics (TOG) **41**(4), 1–13 (2022)
8. Guo, J., Ma, X., Fan, Y., Liu, H., Li, Q.: Semantic Gaussians: Open-Vocabulary Scene Understanding with 3D Gaussian Splatting. IEEE Transactions on Circuits and Systems for Video Technology pp. 1–1 (2026)
9. Hao, J., Wang, P., Wang, H., Zhang, X., Guo, Z.: GaussVideoDreamer: 3D Scene Generation with Video Diffusion and Inconsistency-Aware Gaussian Splatting. arXiv preprint (2025), arXiv:2504.10001
10. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV) (2023)
11. Jiang, D., Liu, Y., Liu, S., Zhao, J., Zhang, H., Gao, Z., Zhang, X., Li, J., Xiong, H.: From clip to dino: Visual encoders shout in multi-modal large language models. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2024)
12. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., Lin, D., Dai, B.: AnySplat: Feed-forward 3D Gaussian Splatting from Unconstrained Views. In: Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH) Asia (2025)
13. Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H., Gao, F., Yang, Y.e.a.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In: Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH) (2024)
14. Jun-Seong, K., Kim, G., Yu-Ji, K., Wang, Y.C.F., Choe, J., Oh, T.H.: Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2025)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. on Graphics (TOG) **42**(4), 139–1 (2023)
16. Kim, J., Lee, S., Shin, J., Choi, J., Shim, H.: DreamCatalyst: Fast and High-Quality 3D Editing via Controlling Editability and Identity Preservation. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2025)
17. Kim, Y., Anagnostidis, S., Du, Y., Schönfeld, E., Kohler, J., Georgopoulos, M., Pumarola, A., Thabet, A., Sanakoyeu, A.: Autoregressive distillation of diffusion transformers. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2025)
18. Kundu, A., Genova, K., Yin, X., Fathi, A., Pantofaru, C., Guibas, L.J., Tagliasacchi, A., Dellaert, F., Funkhouser, T.: Panoptic neural fields: A semantic object-aware neural scene representation. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
19. Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P., Farhadi, A.: Matryoshka representation learning. In: Proc. of the Conf. on Neural Information Processing Systems (NeurIPS) (2022)
20. Labs, B.F.: Introducing flux.1 dev. `https://bfl.ai/blog/24-08-01-bfl` (2024), accessed 23 June 2026
21. Lee, D.I., Park, H., Seo, J., Park, E., Park, H., Baek, H.D., Shin, S., Kim, S., Kim, S.: Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2025)

22. Lipman, Y., Chen, R., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2023)
23. McLachlan, G.J.: Mahalanobis distance. Resonance **4**(6), 20–26 (1999)
24. Mei, Y., Xu, J., Patel, V.M.: Reference-based controllable scene stylization with gaussian splatting. Proc. of the Conf. on Neural Information Processing Systems (NeurIPS) (2024)
25. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. Communications of the ACM **65**(1), 99–106 (2021)
26. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI) (2024)
27. NVIDIA, Abu Alhaija, H., Alvarez, J., Bala, M., Cai, T., Cao, T., Cha, L., Chen, J., Chen, M., Ferroni, F., Fidler, S., Fox, D., Ge, Y., Gu, J., Hassani, A., Isaev, M., Jannaty, P., Lan, S., Lasser, T., Ling, H., Liu, M.Y., Liu, X., Lu, Y., Luo, A., Ma, Q., Mao, H., Ramos, F., Ren, X., Shen, T., Tang, S., Wang, T.C., Wu, J., Xu, J., Xu, S., Xie, K., Ye, Y., Yang, X., Zeng, X., Zeng, Y.: Cosmos-Transfer1: Conditional World Generation with Adaptive Multimodal Control. arXiv preprint (2025), arXiv:2503.14492
28. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Trans. on Machine Learning Research (TMLR) (2024)
29. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: SDXL: Improving latent diffusion models for high-resolution image synthesis. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2024)
30. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFUSION: Text-to-3d using 2d Diffusion. Proc. of the Intl. Conf. on Learning Representations (ICLR) (2023)
31. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.Y., Skorokhodov, I., Wonka, P., Tulyakov, S., Ghanem, B.: Magic123: One Image to High-Quality 3D Object Generation Using Both 2D and 3D Diffusion Priors. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2024)
32. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
33. Qiu, R.Z., Yang, G., Zeng, W., Wang, X.: Language-Driven Physics-Based Scene Synthesis and Editing via Feature Splatting. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J.G.K., Sutskever, I.: Learning Transferable Visual Models from Natural Language Supervision. In: Proc. of the Intl. Conf. on Machine Learning (ICML) (2021)
35. Rippel, O., Gelbart, M., Adams, R.: Learning ordered representations with nested dropout. In: Proc. of the Intl. Conf. on Machine Learning (ICML) (2014)

36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
37. Sargent, K., Li, Z., Shah, T., Herrmann, C., Yu, H.X., Zhang, Y., Chan, E.R., Lagun, D., Fei-Fei, L., Sun, D., Wu, J.: ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image. Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
38. Shi, J.C., Wang, M., Duan, H.B., Guan, S.H.: Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
39. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3. arXiv preprint (2025), arXiv:2508.10104
40. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: Proc. of the Intl. Conf. on Machine Learning (ICML) (2015)
41. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2024)
42. Tonderski, A., Lindström, C., Hess, G., Ljungbergh, W., Svensson, L., Petersson, C.: NeuRAD: Neural Rendering for Autonomous Driving. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
43. Vachha, C., Haque, A.: Instruct-gs2gs: Editing 3d gaussian splats with instructions. `https://instruct-gs2gs.github.io/` (2024), accessed 23 June 2026
44. Wang, J., Laube, K.A., Li, Y., Metzen, J.H., Cheng, S.I., Borges, J., Khoreva, A.: Label-free neural semantic image synthesis. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
45. Wang, J., Fang, J., Zhang, X., Xie, L., Tian, Q.: GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
46. Wang, Y., Yi, X., Wu, Z., Zhao, N., Chen, L., Zhang, H.: View-consistent 3d editing with gaussian splatting. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
47. Wang, Z.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Processing **13**(4), 600–612 (2004)
48. Wen, M., Wu, S., Wang, K., Liang, D.: Intergsedit: Interactive 3d gaussian splatting editing with 3d geometry-consistent attention prior. In: Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV) (2025)
49. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2025)
50. Wu, J., Bian, J.W., Li, X., Wang, G., Reid, I., Torr, P., Prisacariu, V.A.: GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
51. Wu, T., Yuan, Y.J., Zhang, L.X., Yang, J., Cao, Y.P., Yan, L.Q., Gao, L.: Recent advances in 3d gaussian splatting. Computational Visual Media **10**(4), 613–642 (2024)

52. Wu, Y., Meng, J., Li, H., Wu, C., Shi, Y., Cheng, X., Zhao, C., Feng, H., Ding, E., Wang, J., et al.: Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. Proc. of the Conf. on Neural Information Processing Systems (NeurIPS) **37** (2024)
53. Xiao, H., Chen, Y., Huang, H., Xiong, H., Yang, J., Prasad, P., Zhao, Y.: Localized gaussian splatting editing with contextual awareness. In: Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV) (2025)
54. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
55. Xiong, Z., Chen, Z., Li, Z., Xu, Y., Jacobs, N.: PanoDreamer: Consistent Text to 360-Degree Scene Generation. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops (2025)
56. Yan, C., Qu, D., Xu, D., Zhao, B., Wang, Z., Wang, D., Li, X.: Gs-slam: Dense visual slam with 3d gaussian splatting. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
57. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In: Proc. of the Intl. Conf. on Learning Representations (ICLR) (2024)
58. Yang, J., Luo, K.Z., Li, J., Deng, C., Guibas, L., Krishnan, D., Weinberger, K.Q., Tian, Y., Wang, Y.: Denoising vision transformers. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
59. Yang, X., Wen, L., Ma, Y., Mei, J., Li, X., Wei, T., Lei, W., Fu, D., Cai, P., Dou, M., Shi, B., He, L., Liu, Y., Qiao, Y.: Drivearena: A closed-loop generative simulation platform for autonomous driving. In: Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV) (2025)
60. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
61. Yiwen Chen and Zilong Chen and Chi Zhang and Feng Wang and Xiaofeng Yang and Yikai Wang and Zhongang Cai and Lei Yang and Huaping Liu and Guosheng Lin: GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
62. Yue, Y., Das, A., Engelmann, F., Tang, S., Lenssen, J.E.: Improving 2d feature representations by 3d-aware fine-tuning. In: Proc. of the Europ. Conf. on Computer Vision (ECCV) (2024)
63. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV) (2023)
64. Zheng, Y., Chen, X., Zheng, Y., Gu, S., Yang, R., Jin, B., Li, P., Zhong, C., Wang, Z., Liu, L.e.a.: Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. IEEE Robotics and Automation Letters (RA-L) **9**(9), 7827–7834 (2024)
65. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV) (2021)
66. Zhou, H., Lin, L., Wang, J., Lu, Y., Bai, D., Liu, B., Wang, Y., Geiger, A., Liao, Y.: HUGSIM: A Real-Time, Photo-Realistic and Closed-Loop Simulator for Autonomous Driving. arXiv preprint (2024), arXiv:2412.01718

67. Zhou, H., Shao, J., Xu, L., Bai, D., Qiu, W., Liu, B., Wang, Y., Geiger, A., Liao, Y.: Hugs: Holistic urban 3d scene understanding via gaussian splatting. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
68. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: iBOT: Image BERT Pre-Training with Online Tokenizer. Proc. of the Intl. Conf. on Learning Representations (ICLR) (2022)
69. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (2024)
70. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH) Asia (2023)
71. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa volume splatting. In: Proc. of Visualization (VIS) (2001)