LEARNING SHARED OBJECT PROTOTYPES FOR COMPOSITIONAL 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing 3D Gaussian Splatting (3DGS) methods typically reconstruct each scene independently, leaving recurring objects tied to their original scenes and preventing their representations from being directly reused or recombined. We present a prototype-based compositional 3DGS representation that learns globally shared Gaussian prototypes from multiple scenes. Given posed multi-view images with externally provided object masks and category labels, our method learns a canonical prototype for each recurring object category, while representing each instance using a lightweight, instance-specific similarity transformation. Individual scenes are constructed by instantiating and composing these shared prototypes according to their scene-specific configurations. To address the strong coupling between prototype parameters and instance transformations, we introduce an alternating optimization strategy together with a fixed canonical reference and prototype regularization. Experiments on GSO and OCTScenes demonstrate that the learned prototypes preserve the geometry and appearance shared by recurring objects, maintain high-quality novel-view synthesis, and can be instantiated and recombined to construct new scene configurations. Sharing global prototypes also naturally reduces repeated geometry and appearance parameters across related scenes.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) has recently emerged as an efficient representation for novel-view synthesis, offering high-quality rendering with real-time performance through an explicit set of Gaussian primitives (Kerbl et al., 2023; Lu et al., 2024; Yu et al., 2024). Compared with implicit neural fields such as NeRF (Mildenhall et al., 2020; Müller et al., 2022; Fridovich-Keil et al., 2022), its explicit representation provides a convenient basis for manipulating individual scene components. However, most existing 3DGS methods reconstruct each scene as an independent collection of Gaussians. Consequently, the learned representation remains tied to a particular scene and does not explicitly capture reusable object structures shared across multiple scenes. Even when the same object appears repeatedly with identical geometry and appearance, its representation learned in one scene cannot be directly instantiated or recombined in another.

Object-centric 3D scene representations seek to decompose scenes into individual entities whose content can be represented independently of scene composition (Yu et al., 2021; Yang et al., 2021; Liu et al., 2024). Meanwhile, recent object-aware 3DGS methods associate Gaussian primitives with semantic or instance-level information, enabling object selection, manipulation, and scene understanding (Ye et al., 2024; Zhu et al., 2025). Nevertheless, these methods primarily organize objects within independently reconstructed scenes. Although Scene-Agnostic OC-3DGS introduces a global object-centric codebook for consistent object identification across scenes (Hsu et al., 2026), it shares identity representations rather than the complete Gaussian geometry and appearance of recurring objects. Existing methods therefore do not provide explicit object representations that can be directly reused across scenes and recombined to construct new scene configurations.

A reusable and compositional representation should separate object content from scene configuration. Recurring object structures should be captured by globally shared representations, while the positions, orientations, and scales of individual instances should be modeled independently. Such a formulation would allow an object representation learned from multiple scenes to be instantiated

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

under different transformations and composed with other objects and backgrounds, rather than remaining bound to the scene in which it was originally reconstructed.

In this work, we introduce a prototype-based compositional representation for multi-scene 3D Gaussian Splatting. For each recurring object category, we learn a canonical Gaussian prototype that captures the geometry and appearance shared by its instances across all training scenes. Each instance is represented by an instance-specific similarity transformation that places the corresponding prototype into a scene. We similarly represent the background using a shared canonical prototype together with scene-specific transformations. Each scene is then constructed by composing the transformed object and background prototypes according to its instance configuration.

This prototype–instance decomposition enables object representations learned across multiple scenes to be explicitly reused and recombined. Once the global prototypes have been learned, they can be instantiated using user-specified transformations and composed into new scene configurations without reconstructing each object from scratch. The resulting representation retains the explicit editability of 3DGS and naturally supports translation, rotation, scaling, duplication, removal, and cross-scene recomposition. Because recurring geometry and appearance are shared rather than independently represented in every scene, the proposed formulation also reduces repeated parameters across related scenes.

Learning consistent global prototypes is challenging because the canonical prototypes and instance transformations are strongly coupled. For example, a change in the position, orientation, or scale of a prototype can be compensated for by an inverse change in its instance transformations, producing similar rendered images but an inconsistent canonical representation. Direct joint optimization may therefore allow prototype parameters and instance transformations to mutually compensate for each other. To alleviate this ambiguity, we adopt an alternating optimization strategy that separates the estimation of scene-specific instance parameters from the update of globally shared prototypes. We first optimize the instance transformations while fixing the prototypes and then update the prototypes while holding the instance parameters fixed. We further establish a consistent canonical frame using a fixed reference instance and regularize prototype updates to prevent excessive deviation from their initialized structures.

We conduct experiments on GSO (Downs et al., 2022) and OCTScenes (Huang et al., 2023). We evaluate the proposed representation in terms of novel-view synthesis, prototype reuse, scene composition, and representation efficiency. The results demonstrate that the learned global Gaussian prototypes preserve the geometry and appearance shared by recurring objects, can be instantiated across different scenes, and support the construction of new scene configurations through object-level recomposition. Sharing these prototypes additionally reduces repeated geometry and appearance parameters across related scenes while maintaining rendering quality.

Our main contributions are summarized as follows: 1. We propose a prototype-based compositional 3DGS representation that learns globally shared Gaussian prototypes for recurring object categories and instantiates them through instance-specific similarity transformations, enabling cross-scene object reuse and the construction of new scene configurations. 2. We develop an alternating prototype–instance optimization strategy, together with a canonical-frame constraint and prototype regularization, to alleviate the strong coupling between shared prototypes and instance transformations and facilitate the learning of consistent global prototypes. 3. We demonstrate that the learned prototypes preserve the shared geometry and appearance of recurring objects, support high-quality novel-view synthesis and flexible object-level recomposition, and naturally reduce repeated representation across related scenes.

2 Related Work

2.1 Neural 3D Scene Representations

Neural radiance fields (NeRFs) represent a scene as a continuous function of spatial position and viewing direction, enabling high-quality novel-view synthesis from posed multi-view images (Mildenhall et al., 2020). However, their geometry and appearance are implicitly encoded in network parameters, making direct manipulation and reuse of individual scene components less straightforward. In contrast, 3D Gaussian Splatting (3DGS) represents a scene using an explicit

set of anisotropic Gaussian primitives and employs differentiable, visibility-aware rasterization for real-time rendering (Kerbl et al., 2023). Its explicit representation provides a convenient basis for object-level manipulation and scene composition.

Standard 3DGS independently optimizes a complete Gaussian representation for each scene, making object representations inherently tied to their original scenes. Consequently, it does not provide an explicit mechanism for reusing or recombining recurring objects with the same geometry and appearance across scenes. Our work addresses this limitation by learning shared canonical Gaussian prototypes and representing individual instances through lightweight, instance-specific transformations. The same prototypes can therefore be instantiated across multiple scenes and recombined to construct new scene configurations, while avoiding repeated representation of their geometry and appearance.

2.2 Object-Centric 3D Scene Representations

Object-centric representations model a scene as a composition of entities that can be independently represented, manipulated, and recombined (Yuan et al., 2023; Locatello et al., 2020; Seitzer et al., 2022). uORF (Yu et al., 2021) learns unsupervised object-centric radiance fields from multi-view training images and infers foreground object fields and a background field from a single input image. Object-NeRF (Yang et al., 2021) explicitly separates objects from their surrounding scene, supporting operations such as translation, rotation, duplication, and removal. SlotLifter (Liu et al., 2024) further combines slot-based representations with multi-view feature lifting for object-centric scene decomposition and novel-view synthesis. These methods demonstrate the benefit of separating object content from scene composition, but mainly focus on object discovery, scene decomposition, novel-view synthesis, or scene editing through implicit fields and learned latent representations. Our method draws inspiration from this separation but does not address object discovery or unsupervised decomposition. Instead, given externally provided object masks and category labels, we explicitly represent recurring objects using shared Gaussian prototypes and instantiate them across scenes through instance-specific transformations, enabling their reuse and recombination into new scene configurations.

2.3 Object-Aware and Composable 3DGS

Recent methods augment 3DGS with semantic or instance-level information for object-aware scene understanding and editing. Gaussian Grouping (Ye et al., 2024) assigns identity encodings to Gaussian primitives under multi-view mask supervision and spatial consistency, allowing Gaussians belonging to the same object to be grouped and manipulated. ObjectGS (Zhu et al., 2025) models individual objects as local anchors that generate neural Gaussians and share object IDs, integrating scene reconstruction with semantic and instance understanding.

Although these methods introduce object-level structure, their object representations remain tied to independently reconstructed scenes. Object identities organize Gaussians within each scene, but the geometry and appearance of recurring objects are not shared across scenes. They therefore support object-aware manipulation within individual scenes, without providing a shared representation that can be directly reused and recombined across scenes.

Scene-Agnostic OC-3DGS (Hsu et al., 2026) moves toward cross-scene consistency by learning a global object-centric codebook. The codebook provides identity-anchored supervision for Gaussian features across scenes, enabling scene-agnostic object identification. However, it shares object identity representations rather than the complete Gaussian geometry and appearance of recurring objects.

Our method focuses on learning explicit and reusable object representations across scenes. We learn a canonical Gaussian prototype for each recurring object category and share its geometry and appearance across all corresponding instances. Each instance is represented by a lightweight, instance-specific similarity transformation, allowing the same prototype to be instantiated in different scenes and recombined with other objects and backgrounds. This prototype–instance decomposition enables the construction of new scene configurations using learned prototypes and user-specified transformations, while retaining the explicit editability of 3DGS and naturally reducing repeated representation of object geometry and appearance.

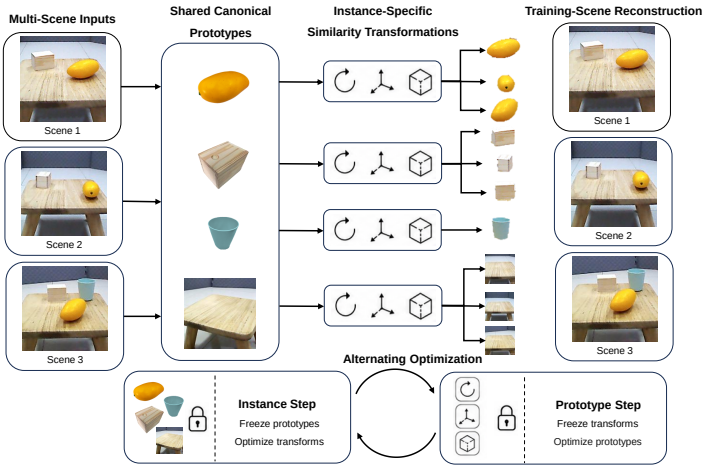


Figure 1: Overview of our prototype-based compositional 3D Gaussian Splatting framework. Shared canonical prototypes are learned from recurring objects across multiple scenes and instantiated through instance-specific similarity transformations. We alternate between optimizing instance transformations with fixed prototypes and updating shared prototypes with fixed transformations.

3 METHOD

3.1 PROBLEM FORMULATION

We consider a dataset of S static scenes, where scene s contains N_s foreground object instances drawn from C categories and is observed from V viewpoints. The input is

$$\mathcal{D} = \left\{ \left\{ \left(\mathbf{I}_{s,v}, \mathbf{K}_{s,v}, \mathbf{E}_{s,v}, \{M_{s,v,i}\}_{i=1}^{N_s} \right) \right\}_{v=1}^{V} \right\}_{s=1}^{S}, \tag{1}$$

where $\mathbf{I}_{s,v}$ is an RGB image, $\mathbf{K}_{s,v}$ and $\mathbf{E}_{s,v}$ are its camera intrinsics and extrinsics, and $M_{s,v,i}$ is the visible 2D mask of instance i. Each instance has a globally consistent category label $y_{s,i} \in \{1, \dots, C\}$. The masks and labels are provided by an external object-centric model. We assume that instances of the same category share canonical geometry and appearance and differ only in pose and scale. All scenes additionally share the same static background.

Since each scene is reconstructed in its own coordinate system, we align all scenes to a shared canonical frame using fixed per-scene similarity transformations:

$$\mathbf{x}^{\text{can}} = b_s \mathbf{R}_s^A \mathbf{x}^{(s)} + \mathbf{t}_s^A, \tag{2}$$

where $\mathbf{x}^{(s)} \in \mathbb{R}^3$ denotes a 3D point in the coordinate system of scene s, and $\mathbf{x}^{\text{can}}$ is its coordinate in the shared canonical frame. Here, $b_s > 0$, $\mathbf{R}_s^A \in \text{SO}(3)$, and $\mathbf{t}_s^A \in \mathbb{R}^3$ denote the scale, rotation, and translation of the fixed scene-alignment transformation, respectively. The camera extrinsics are transformed accordingly, and the alignment transformations remain fixed during training.

Our goal is to learn one canonical Gaussian prototype for the shared background and one for each foreground category, together with an instance-specific similarity transformation:

$$\mathcal{P} = \{\mathcal{P}_c\}_{c=0}^{C}, \qquad \boldsymbol{\theta}_{s,i} = (\mathbf{R}_{s,i}, \mathbf{t}_{s,i}, a_{s,i}), \tag{3}$$

where $\mathcal{P}_0$ denotes the background prototype and $\mathcal{P}_{y_{s,i}}$ generates instance i through rotation, translation, and scaling. The shared prototypes and instance transformations are optimized to explain the observed RGB images and object masks across all scenes and viewpoints.

3.2 PROTOTYPE-BASED SCENE COMPOSITION

For each category $c \in \{0, \dots, C\}$, we represent its canonical prototype as a set of 3D Gaussians:

$$\mathcal{P}_c = \{g_{c,j}\}_{j=1}^{J_c}, \qquad g_{c,j} = (\boldsymbol{\mu}_{c,j}, \boldsymbol{\Sigma}_{c,j}, \alpha_{c,j}, \mathbf{h}_{c,j}), \tag{4}$$

where $\boldsymbol{\mu}_{c,j}$, $\boldsymbol{\Sigma}_{c,j}$, $\alpha_{c,j}$, and $\mathbf{h}_{c,j}$ denote the Gaussian center, covariance, opacity, and spherical harmonic (SH) coefficients, respectively. As in standard 3DGS, each covariance is parameterized by a rotation and anisotropic scaling.

For foreground instance i in scene s, let $c = y_{s,i}$ be its category. Its Gaussian geometry is generated from $\mathcal{P}_c$ using the instance-specific similarity transformation $\boldsymbol{\theta}_{s,i}$, which maps the category-specific prototype frame to the shared canonical scene frame:

$$\boldsymbol{\mu}_{s,i,j} = a_{s,i}\mathbf{R}_{s,i}\boldsymbol{\mu}_{c,j} + \mathbf{t}_{s,i}, \qquad \boldsymbol{\Sigma}_{s,i,j} = a_{s,i}^2\mathbf{R}_{s,i}\boldsymbol{\Sigma}_{c,j}\mathbf{R}_{s,i}^\top. \tag{5}$$

The opacity is inherited directly from the canonical prototype, i.e., $\alpha_{s,i,j} = \alpha_{c,j}$. Thus, all instances of the same category share canonical geometry and appearance, while their transformations determine their poses and scales.

The SH coefficients remain defined in the category-specific prototype frame. To rotate view-dependent appearance consistently with the instance, we transform the viewing direction in the shared canonical scene frame back to the prototype frame:

$$\mathbf{d}_{s,v,i,j}^{\text{proto}} = \mathbf{R}_{s,i}^\top \mathbf{d}_{s,v,i,j}^{\text{can}}, \qquad \mathbf{c}_{s,v,i,j} = \text{SH}\left(\mathbf{h}_{c,j}, \mathbf{d}_{s,v,i,j}^{\text{proto}}\right), \tag{6}$$

where $\mathbf{d}_{s,v,i,j}^{\text{can}}$ is the normalized viewing direction from the camera center to the Gaussian center in the shared canonical scene frame, and $\mathbf{d}_{s,v,i,j}^{\text{proto}}$ is the corresponding direction in the category-specific prototype frame. This allows the SH coefficients to remain shared while preserving the directional appearance of each transformed instance.

The Gaussian representation of scene s is constructed by composing the shared background prototype with all transformed foreground prototypes:

$$\mathcal{G}_s = \mathcal{P}_0 \cup \bigcup_{i=1}^{N_s} \mathcal{T}\left(\mathcal{P}_{y_{s,i}}; \boldsymbol{\theta}_{s,i}\right), \tag{7}$$

where $\mathcal{T}$ applies the transformation defined above. The background prototype $\mathcal{P}_0$ is already represented in the shared canonical frame and is reused across all scenes without an instance transformation.

We render the composed scene using differentiable Gaussian splatting:

$$\widehat{\mathbf{I}}_{s,v} = \mathcal{R}\left(\mathcal{G}_s; \mathbf{K}_{s,v}, \mathbf{E}_{s,v}^{\text{can}}\right), \tag{8}$$

where $\mathbf{E}_{s,v}^{\text{can}}$ denotes the camera extrinsics after scene alignment. All foreground and background Gaussians are projected and alpha-composited together in depth order.

To render occlusion-aware instance masks, we assign a one-hot feature $\mathbf{e}_i \in \mathbb{R}^{N_s}$ to the Gaussians of instance i and the zero vector to the background:

$$\mathbf{f}_{s,i,j} = \begin{cases} \mathbf{0}, & i = 0, \\ \mathbf{e}_i, & i = 1, \ldots, N_s, \end{cases} \qquad \widehat{M}_{s,v,i} = \left[\mathcal{R}_{\text{feat}}\left(\mathcal{G}_s, \{\mathbf{f}_{s,k,j}\}; \mathbf{K}_{s,v}, \mathbf{E}_{s,v}^{\text{can}}\right)\right]_i. \tag{9}$$

The feature renderer uses the same depth ordering and alpha compositing as RGB rendering, so $\widehat{M}_{s,v,i}$ represents the visible contribution of instance i under inter-object occlusion.

3.3 Alternating Prototype–Instance Optimization

We optimize the composed scenes using RGB reconstruction and instance-mask supervision. For scene s and viewpoint v, the photometric loss is

$$\mathcal{L}_{\text{rgb}}^{s,v} = (1 - \lambda_{\text{ssim}})\left\|\widehat{\mathbf{I}}_{s,v} - \mathbf{I}_{s,v}\right\|_1 + \lambda_{\text{ssim}}\mathcal{L}_{\text{D-SSIM}}\left(\widehat{\mathbf{I}}_{s,v}, \mathbf{I}_{s,v}\right), \tag{10}$$

where the ℓ_1 term is averaged over all pixels and color channels. We supervise the instance masks using binary cross-entropy:

$$\mathcal{L}_{\text{mask}}^{s,v} = \frac{1}{N_s|\Omega|}\sum_{i=1}^{N_s}\sum_{\mathbf{p}\in\Omega}\text{BCE}\left(\widehat{M}_{s,v,i}(\mathbf{p}), M_{s,v,i}(\mathbf{p})\right), \tag{11}$$

where Ω denotes the image domain and $|\Omega|$ is the total number of pixels in an image.

The overall reconstruction objective is

$$\mathcal{L}_{\text{rec}} = \frac{1}{SV} \sum_{s=1}^{S} \sum_{v=1}^{V} \left(\mathcal{L}_{\text{rgb}}^{s,v} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}}^{s,v} \right), \tag{12}$$

where λ_{mask} weights the mask supervision.

Jointly optimizing the prototypes and instance transformations is unstable because prototype geometry and instance pose or scale can compensate for one another. We therefore alternate between optimizing the instance transformations and the shared prototypes.

We initialize the model from independently reconstructed scene components. The background prototype is initialized from the static-background Gaussians of the reference scene, which defines the shared canonical scene frame.

For each foreground instance i in scene s, let $\mathbf{z}_{s,i}$ and $d_{s,i} > 0$ denote its center and isotropic size in the local coordinate system of scene s. For each foreground category c, we select a reference instance $r_c = (s_c, i_c)$. The independently reconstructed Gaussians of the reference instance are normalized to initialize the category prototype:

$$\boldsymbol{\mu}_{c,j}^{(0)} = \frac{\boldsymbol{\mu}_{s_c,i_c,j}^{\text{rec}} - \mathbf{z}_{s_c,i_c}}{d_{s_c,i_c}}, \qquad \boldsymbol{\Sigma}_{c,j}^{(0)} = \frac{\boldsymbol{\Sigma}_{s_c,i_c,j}^{\text{rec}}}{d_{s_c,i_c}^2}. \tag{13}$$

Gaussian opacities and SH coefficients are copied directly, while the Gaussian orientations encoded in the covariance matrices remain unchanged under isotropic normalization.

Let $\overline{\mathbf{R}}_{s,i} \in \mathrm{SO}(3)$ denote the initial relative rotation from the canonical orientation defined by the reference instance of category $y_{s,i}$ to the orientation of instance i in the local coordinate system of scene s. By definition, $\overline{\mathbf{R}}_{s_c,i_c} = \mathbf{I}$ for each reference instance. The initial instance transformation combines this relative rotation, center, and size with the fixed scene alignment:

$$\mathbf{R}_{s,i}^{(0)} = \mathbf{R}_s^A \overline{\mathbf{R}}_{s,i}, \qquad \mathbf{t}_{s,i}^{(0)} = b_s \mathbf{R}_s^A \mathbf{z}_{s,i} + \mathbf{t}_s^A, \qquad a_{s,i}^{(0)} = b_s d_{s,i}. \tag{14}$$

To fix the category-specific similarity gauge, each reference-instance transformation is held fixed throughout training:

$$\boldsymbol{\theta}_{s_c,i_c} = \boldsymbol{\theta}_{s_c,i_c}^{(0)}, \qquad c = 1, \ldots, C. \tag{15}$$

Let $\boldsymbol{\Phi}$ contain all prototype parameters and $\boldsymbol{\Theta}_{\text{free}}$ contain the transformations of all non-reference instances. The fixed reference-instance transformations are treated as implicit constants in the following objectives. At alternating iteration t, we first keep the prototypes fixed and optimize the free instance transformations:

$$\boldsymbol{\Theta}_{\text{free}}^{(t+1)} = \arg\min_{\boldsymbol{\Theta}_{\text{free}}} \mathcal{L}_{\text{rec}} \left(\boldsymbol{\Phi}^{(t)}, \boldsymbol{\Theta}_{\text{free}} \right). \tag{16}$$

We then fix all instance transformations and update the shared prototypes:

$$\boldsymbol{\Phi}^{(t+1)} = \arg\min_{\boldsymbol{\Phi}} \left[\mathcal{L}_{\text{rec}} \left(\boldsymbol{\Phi}, \boldsymbol{\Theta}_{\text{free}}^{(t+1)} \right) + \mathcal{L}_{\text{reg}} \left(\boldsymbol{\Phi}; \boldsymbol{\Phi}^{(0)} \right) \right]. \tag{17}$$

Each subproblem is approximated using a fixed number of gradient-descent steps. The instance transformations are first aligned with the current prototypes, after which the prototypes aggregate reconstruction signals from all corresponding instances.

During prototype updates, we regularize only the foreground prototypes:

$$\mathcal{L}_{\text{reg}} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{J_c} \sum_{j=1}^{J_c} \ell_{c,j}^{\text{reg}}, \tag{18}$$

where the per-Gaussian regularizer is

$$\ell_{c,j}^{\text{reg}} = \lambda_{\text{xyz}} \left\| \boldsymbol{\mu}_{c,j} - \boldsymbol{\mu}_{c,j}^{(0)} \right\|_2^2 + \lambda_{\text{shape}} \mathcal{D}_{\text{shape}} \left(\boldsymbol{\Sigma}_{c,j}, \boldsymbol{\Sigma}_{c,j}^{(0)} \right) + \lambda_{\text{opacity}} \left(\alpha_{c,j} - \alpha_{c,j}^{(0)} \right)^2. \tag{19}$$

Here, $\mathcal{D}_{\text{shape}}$ penalizes changes in the scale and rotation parameters underlying the Gaussian covariance. These proximal terms suppress prototype drift by keeping Gaussian positions, shapes, and opacities close to their initialization. No regularization is imposed on the SH coefficients. The background prototype is excluded from this regularization and has no dedicated mask target, but is refined primarily through full-scene RGB supervision from all training scenes.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

4.1.1 DATASETS AND PROTOCOL

We evaluate our method on two datasets: a synthetic dataset constructed from Google Scanned Objects (GSO) (Downs et al., 2022), and a real-world dataset OCTScenes (Huang et al., 2023). These datasets contain repeated instances corresponding to the same underlying objects under varying poses, scales, and scene configurations, and therefore align well with our shared-prototype setting.

For each dataset, we select 20 multi-object scenes, ensuring that each evaluated object appears repeatedly across different scenes. Each scene contains 60 views, of which 52 are used for training and the remaining 8 are held out for evaluation. We use the same selected scenes and view splits for all compared methods. The exact scene and view indices are provided in the Appendix B.

We obtain multi-view instance masks and consistent object identities from a pretrained object-centric model and use them as image-space supervision. All instance transformations are optimized without using ground-truth object poses.

4.1.2 BASELINES

We compare with three representative baselines. **3DGS** (Kerbl et al., 2023) reconstructs each scene independently and serves as the standard scene-specific baseline for reconstruction quality and representation efficiency. **Gaussian Grouping** (Ye et al., 2024) learns identity features from 2D mask supervision for object grouping within individual scenes. **ObjectGS** (Zhu et al., 2025) uses local anchors to generate object-aware Gaussians but does not share object representations across scenes.

For scene-specific baselines, we train one model per scene and sum their Gaussian counts and storage over the 20 scenes in each dataset. When applicable, the same pseudo-masks and object identities are provided to all methods. Detailed implementation settings and baseline configurations are provided in Appendix C

4.1.3 EVALUATION METRICS

Reconstruction quality. We evaluate novel-view synthesis on the held-out views using PSNR, SSIM, and LPIPS. The metrics are averaged first across the test views of each scene and then across all evaluated scenes in each dataset.

Representation efficiency. We report the total storage required to represent all evaluated scenes, including the shared object prototypes, instance transformations, backgrounds, and any additional parameters required for rendering. We further report the compression ratio relative to independently trained 3DGS models.

Scene composition and object-centric editing. Since the newly composed and edited configurations do not have corresponding ground-truth images, we evaluate them qualitatively. We examine whether the learned prototypes preserve consistent geometry and appearance under translation, rotation, scaling, duplication, and novel spatial arrangements.

4.1.4 IMPLEMENTATION DETAILS

We first reconstruct each scene independently using standard 3DGS for 30000 iterations and partition the resulting Gaussians into foreground instances and a static background using the predicted multi-view masks. The shared background is initialized from a reference scene, while each foreground prototype is initialized from a normalized reference instance whose transformation remains fixed throughout training.

In each alternating cycle, we optimize the non-reference instance transformations for 1000 steps and the shared prototypes for 1000 steps. We perform 15 cycles using the Adam optimizer. The mask-loss weight is set to 0.1. All experiments are conducted at 256×256 on a single NVIDIA GeForce RTX 4090. Other training details are provided in Appendix D.

Table 1: Total model storage and held-out view fidelity. Comp. denotes the compression ratio. Bold indicates the best result for each metric within each dataset.

Dataset	Method	Storage (MB)↓	Comp.↑	PSNR↑	SSIM↑	LPIPS↓
GSO	3DGS	2727.81	1.00×	26.83	0.856	0.110
	Gaussian Grouping	3264.38	0.84×	27.38	0.859	0.112
	ObjectGS	1232.92	2.21×	**28.64**	**0.885**	**0.092**
	Ours	**142.69**	**19.12**×	27.08	0.860	0.104
OCTScenes	3DGS	1734.30	0.84×	23.29	0.746	0.239
	Gaussian Grouping	2107.37	0.82×	23.26	0.746	0.236
	ObjectGS	881.33	1.97×	**24.44**	**0.781**	**0.192**
	Ours	**108.77**	**15.94**×	23.89	0.747	0.240

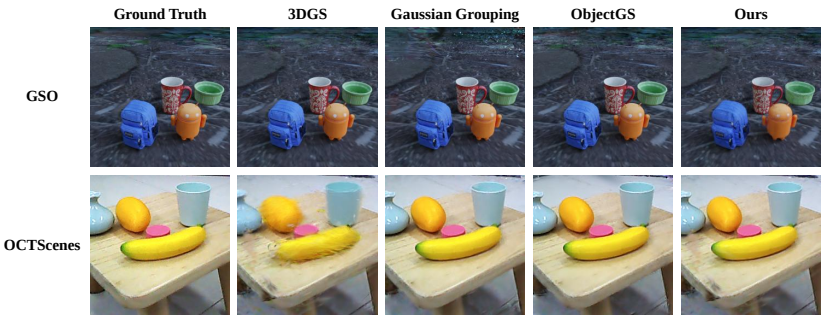


Figure 2: Held-out-view rendering comparison on GSO (top) and OCTScenes (bottom).

4.2 Storage Efficiency and Held-Out View Fidelity

We compare the total model storage across all scenes in each dataset and evaluate rendering quality on held-out views. As shown in Table 1, sharing prototypes of recurring objects across scenes substantially reduces total storage. This saving comes from storing each prototype once rather than maintaining separate object representations for every scene. Figure 2 compares representative held-out views and shows the rendering quality retained by this compact representation.

4.3 Rendering Shared Object Prototypes

Figure 3 shows turntable renderings of four learned object prototypes, with two selected from each dataset. Each row presents views distributed around one prototype, illustrating that it can be rendered independently of its original scene configuration.

4.4 Ablation Studies

We evaluate the main components of our framework, including alternating prototype–instance optimization, reference-instance freezing, mask supervision, and prototype regularization. Unless otherwise stated, all variants are evaluated on OCTScenes using the same initialization, data split, and training schedule as the full model.

Alternating prototype–instance optimization. Jointly optimizing prototype parameters and instance transformations introduces a strong ambiguity: changes in prototype geometry can be compensated by instance pose and scale, and vice versa. In contrast, alternating between the instance and prototype steps separates the two optimization objectives and produces more consistent shared prototypes.

Reference-instance freezing. The reference instance fixes the canonical coordinate frame of each shared prototype. Removing this constraint has only a limited effect on image-space reconstruction



Figure 3: Multi-view renderings of learned object prototypes. The left and right panels show two objects from GSO and OCTScenes, respectively. Each row presents one object from each dataset viewed from multiple azimuth angles.

Table 2: Ablation study of the main components of our method. Bold indicates the best result for each metric.

Variant	PSNR↑	SSIM↑	LPIPS↓
Joint optimization	23.21	0.731	0.266
Without reference freezing	23.71	0.745	0.258
Without mask supervision	23.43	0.736	0.263
Without prototype regularization	23.62	0.742	0.251
Ours	**23.89**	**0.747**	**0.240**

metrics because a transformation of the prototype can be compensated by inverse transformations of its instances. However, the learned canonical frame becomes unstable and varies across runs, producing noticeable translation, rotation, and scale drift. Freezing the reference transformation resolves this ambiguity and makes the learned prototypes directly reusable for editing and novel-scene composition.

Mask supervision. Without mask supervision, Gaussians near object boundaries can be incorrectly assigned to neighboring objects or the background. This results in blurred boundaries, floating Gaussian artifacts, and inconsistent appearance in regions where multiple objects overlap. Instance-mask supervision encourages a cleaner separation between object prototypes and improves both reconstruction quality and the visual integrity of transformed instances.

Prototype regularization. Removing prototype regularization gives the shared Gaussians excessive freedom to adapt to individual training scenes. Although this may fit some observations locally, it causes geometric distortion, opacity artifacts, and drift from the canonical initialization. Prototype regularization provides a balance between cross-scene refinement and canonical consistency, leading to more stable optimization and more reliable reuse in novel configurations.

5 Conclusion

We presented a prototype-based compositional representation for multi-scene 3D Gaussian Splatting. By learning globally shared canonical prototypes and representing individual instances through scene-specific similarity transformations, our method separates reusable object content from scene configuration. This formulation allows learned objects to be instantiated across scenes and recombined to construct new scene configurations. To address the strong coupling between prototype parameters and instance transformations, we introduced an alternating optimization strategy with a fixed canonical reference and prototype regularization. Experiments on GSO and OCTScenes demonstrate that the learned prototypes preserve the shared geometry and appearance of recurring objects while supporting high-quality novel-view synthesis and flexible object-level recomposition. Sharing prototypes also naturally reduces repeated geometry and appearance parameters across related scenes.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, sourcing and searching for information, summarizing and analyzing existing literature, identifying relevant literature and research gaps, and drafting, editing, and improving the readability of portions of the manuscript. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, clean or reformat datasets, or interpret results. We have reviewed all AI-assisted work: translations and text were revised by the authors, and literature-related suggestions were checked against the original sources before citation. The methodology, implementation, experiments, and final conclusions were developed and validated by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets and does not involve the collection of new data or studies with human participants. We use these datasets in accordance with their original licenses and terms of use. The proposed method is intended for research on reusable 3D scene representations and does not make decisions affecting individuals. For potential real-world applications, users should respect privacy, data ownership, and applicable usage restrictions when reconstructing or reusing scene content.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main paper describes the proposed prototype-based compositional representation, scene composition and transformation mechanisms, and optimization objectives. It also specifies the datasets, evaluation protocol, and metrics used in our experiments. The supplementary material provides additional implementation and training details, including hyperparameter settings, as well as further experimental results and analyses. Together, these descriptions provide the information required to reproduce the method and evaluate it under the reported experimental setting.

REFERENCES

Tonglin Chen, Yinxuan Huang, Zhimeng Shen, Jinghao Huang, Bin Li, and Xiangyang Xue. Learning global object-centric representations via disentangled slot attention. *Machine Learning*, 114 (2):40, 2025.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560. IEEE, 2022.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5501–5510, 2022.

Tsuheng Hsu, Guiyu Liu, Juho Kannala, and Janne Heikkilä. Scene-agnostic object-centric representation learning for 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 219–228, June 2026.

Yinxuan Huang, Tonglin Chen, Zhimeng Shen, Jinghao Huang, Bin Li, and Xiangyang Xue. Octscenes: A versatile real-world dataset of tabletop scenes for object-centric learning. *arXiv preprint arXiv:2306.09682*, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Yu Liu, Baoxiong Jia, Yixin Chen, and Siyuan Huang. Slotlifter: Slot-guided feature lifting for learning object-centric radiance fields. In *European Conference on Computer Vision*, pp. 270–288. Springer, 2024.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20654–20664, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pp. 405–421. Springer, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2022.

Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13779–13788, 2021.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pp. 162–179. Springer, 2024.

Hong-Xing Yu, Leonidas Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *International Conference on Learning Representations*, 2021.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19447–19456, 2024.

Jinyang Yuan, Tonglin Chen, Bin Li, and Xiangyang Xue. Compositional scene representation learning via reconstruction: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11540–11560, 2023.

Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8350–8360, 2025.

A Additional Method Details

Scene registration. We select one scene as the reference and estimate a similarity transformation that maps each remaining scene to the reference coordinate system. Registration is performed using the shared static background while excluding foreground regions. The estimated transformation is then fixed and applied to both the reconstructed Gaussians and the camera poses.

Gaussian decomposition. We decompose each independently reconstructed scene using the 3D classifier of Gaussian Grouping Ye et al. (2024), trained with the predicted multi-view masks. Each Gaussian is assigned to the class with the highest softmax probability. It is included in a foreground instance if the predicted class matches the scene-local mask ID of that instance; otherwise, it is assigned to the background.

Reference selection. For each foreground category, we manually specify one valid `scene/instance` pair in `prototype_references`. No automatic quality-based selection is used. For each reference instance, we compute a percentile bounding box from the 10th and 90th percentiles of its Gaussian centers. Its center is the midpoint of this box, and its isotropic size is the length of the longest side. The reconstructed orientation defines the canonical prototype orientation. The shared background is initialized using the static-background Gaussians of the reference scene.

B Dataset Construction

We evaluate on a synthetic dataset constructed from Google Scanned Objects (GSO) Downs et al. (2022) and the real-world OCTScenes dataset Huang et al. (2023), using 20 scenes from each.

B.1 Datasets

GSO. We construct a synthetic multi-object dataset using six textured GSO assets, one per object category (Table 3). Instances of the same category share the same asset but vary in rotation, translation, and isotropic scale across scenes. Each scene contains 3–6 object instances rendered under HDRI illumination with an environment background. We generate 20 scenes, each with 60 views rendered at 256×256 resolution.

Table 3: GSO object categories and assets.

Category	GSO asset ID
Bottle	`5_HTP`
Mug	`ACE_Coffee_Mug_Kristen_16_oz_cup`
Hat	`DPC_Handmade_Hat_Brown`
Bowl	`45oz_RAMEKIN_ASST_DEEP_COLORS`
Figurine	`Android_Figure_Orange`
Backpack	`Jansport_School_Backpack_Blue_Streak`

OCTScenes. We select 20 multi-object scenes from the OCTScenes test split. The selected scenes cover seven object categories: banana, mango, vase, box, cup, pyramid, and flat cylinder. Each scene contains 2–6 object instances and has 60 views, which are processed at 256×256 resolution. The selected scene IDs are {3101, 3102, 3106, 3117, 3118, 3119, 3121, 3125, 3132, 3143, 3144, 3148, 3154, 3155, 3156, 3159, 3167, 3179, 3184, 3190}.

For both datasets, we use the pretrained GOLD model Chen et al. (2025) to obtain visible instance masks and global object identities. The identities determine which instances share a prototype.

B.2 View Split

We follow the LLFF holdout protocol with `llffhold=8`. Specifically, the views with zero-based indices $\{0, 8, 16, 24, 32, 40, 48, 56\}$ are reserved for testing, resulting in 52 training views and 8 test views per scene.

C Baseline Implementation Details

3DGS. We use the official 3D Gaussian Splatting implementation (Kerbl et al., 2023) with its default training configuration, and train one model independently for each scene for 30,000 iterations. For storage comparison, we report the total size of the final Gaussian PLY files across all evaluation scenes.

Gaussian Grouping. We use the official Gaussian Grouping implementation (Ye et al., 2024) with its default training configuration, and train one model independently for each scene for 30,000 iterations. We use the provided per-view instance masks for its 2D supervision. Storage is the sum of the final Gaussian PLY file and associated classifier weights over all evaluation scenes.

ObjectGS. We use the official ObjectGS implementation (Zhu et al., 2025), specifically its 3DGS variant, and adopt the 3DOVS training configuration. We train one model independently for each scene for 30,000 iterations. We initialize the model from COLMAP sparse point clouds and assign object IDs to the initial points via multi-view majority voting over the training-view instance masks. We use the provided per-view instance masks for object-level supervision. For storage comparison, we sum the final anchor PLY file and the associated color, covariance, and opacity MLP files across all evaluation scenes.

D Additional Implementation Details

We use learning rates of 5×10^{-4}, 5×10^{-4}, and 2×10^{-4} for instance translation, rotation, and isotropic scale, respectively. For the shared prototypes, the learning rates for position, scale, rotation, opacity, and SH coefficients are 1×10^{-5}, 5×10^{-5}, 1×10^{-5}, 5×10^{-5}, and 2.5×10^{-4}, respectively.

E Limitations and Discussion

Our method assumes that instances sharing a prototype have similar geometry and appearance and differ mainly by a similarity transformation, limiting its ability to model large intra-category variations or deformable objects. It also relies on predicted multi-view masks, global object identities, and a shared static background; errors in these inputs may affect prototype learning. Future work will relax these assumptions and support more diverse objects and backgrounds.