FROM GEOMETRY TO SEMANTICS: EFFICIENT 3DGS FEATURE COMPRESSION WITH GEOMETRY-TRANSFERRED LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) enables high-fidelity rendering by representing scenes with explicit Gaussian primitives, but its extension to semantic-aware representations introduces substantial storage overhead due to high-dimensional language-aligned features. Existing compression methods, designed primarily for geometric fidelity, fail to preserve task-relevant semantics under aggressive bitrate constraints, revealing a fundamental mismatch between signal reconstruction and feature geometry preservation. In this work, we revisit semantic compression from a task-oriented perspective and propose a rate–distortion optimized framework for semantic feature compression in 3DGS. Our key insight is that effective compression must preserve the geometry of language-aligned feature spaces rather than raw signal fidelity. To this end, we transfer geometric compression priors to the semantic domain via low-rank adaptation (LoRA), enabling parameter-efficient alignment without modifying the pre-trained backbone. Furthermore, we introduce a geometry-guided attention mechanism that leverages geometric latents as zero-cost side information to enhance semantic consistency during compression. The entire framework is trained end-to-end under a hyperprior-based entropy model, enabling explicit bitrate control. Extensive experiments show that our method achieves near-lossless downstream performance (mIoU 0.812 vs. 0.816 uncompressed) at a significantly reduced bitrate (3.67 bpe), while preserving downstream capabilities such as semantic retrieval and language-guided editing.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a dominant paradigm for photorealistic 3D scene representation, enabling real-time rendering at state-of-the-art visual quality. Beyond rendering for human perception, 3D scenes are increasingly applied to machine perception systems for downstream tasks such as open-vocabulary segmentation (Zhou et al., 2024; Kerr et al., 2023), language-guided editing (Li et al., 2022; Qin et al., 2024), and scene-level querying (Kerr et al., 2023). To support these capabilities, language-augmented variants such as Feature 3DGS (Zhou et al., 2024), LangSplat (Qin et al., 2024) attach each Gaussian with language-aligned semantic features distilled from 2D foundation models, turning 3DGS into a unified representation for both human vision and machine perception (Wu et al., 2024; Li et al., 2025).
However, semantic capabilities come with substantial storage overhead: A typical indoor scene of $\sim$990K Gaussians, each carrying a 128-dimensional semantic feature alongside geometric attributes, requires over 700 MB of total storage, and the semantic features requires about 500 MB. This motivates efficient compression of language-aligned features while retaining their utility for downstream tasks. Existing 3DGS compression methods are designed primarily for human perception, focusing on rendering fidelity and geometric accuracy. Their pretrained compression transforms provide a promising starting point for semantic compression, thus raising a natural question: how can an existing 3DGS compression model be efficiently adapted to language-aligned semantic features?

This adaptation is non-trivial. High-dimensional semantic features differ from the original 3DGS attributes in their dimensionality and statistical properties. Applying FCGS (Chen et al., 2025) to semantic features causes severe degradation in downstream segmentation performance, highlighting

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.

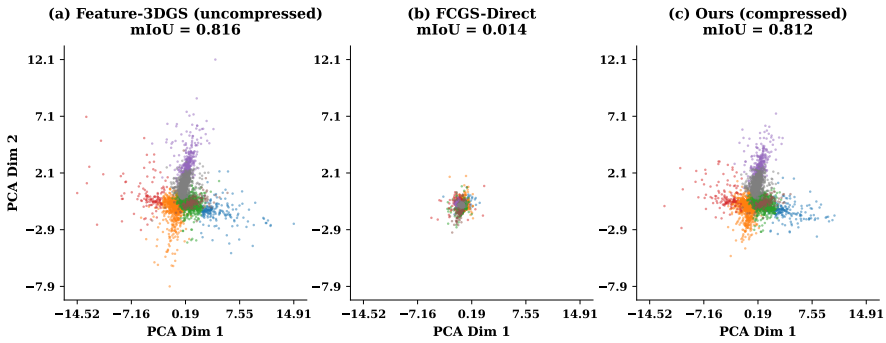

Figure 1: **Distribution mismatch in semantic feature compression.** We visualize language-aligned semantic features using PCA. **(a)** Original Feature-3DGS features exhibit clear class-wise clustering. **(b)** Directly applying FCGS destroys the feature geometry, collapsing semantic clusters and leading to severe performance degradation. **(c)** Our method preserves the underlying feature structure under compression, maintaining semantic separability and task performance.

the need to adapt its pretrained transforms to the semantic domain. More broadly, *existing compression methods optimize for raw signal fidelity, but semantic features require preserving language-aligned feature geometry to support downstream tasks.* Furthermore, the feature distributions visualized in Fig. 1 illustrate the difficulty of retaining semantic structure without adaptation. These observations motivate an adaptation framework that bridges the representation mismatch and adjusts the pretrained compression transforms to the semantic domain. Full fine-tuning offers a straightforward adaptation strategy, but requires updating the entire compression backbone. The challenge is therefore to adapt the pretrained model with a small set of trainable parameters while maintaining a favorable trade-off between bitrate and semantic performance.

In this work, we revisit semantic feature compression from a *task-oriented* perspective and propose the first end-to-end rate–distortion optimized framework centered on parameter-efficient adaptation of a pretrained 3DGS compression backbone. Our approach builds on the compression transforms learned by FCGS and adapts them to language-aligned semantic features through low-rank adaptation (LoRA). By learning low-rank updates while keeping the pretrained backbone weights frozen, we enable semantic-domain adaptation with a small set of trainable parameters. We further exploit Gaussian attributes, including appearance, opacity, scale, and rotation, as conditioning information for semantic latent refinement. A lightweight geometry-guided module incorporates this information before quantization, without requiring a separately transmitted conditioning latent.

Concretely, we make three contributions:

- **Plug-and-play LoRA adaptation for semantic compression.** We propose a LoRA-based parameter-efficient fine-tuning module that can be seamlessly inserted into a frozen geometric compression backbone. This design enables effective adaptation to semantically distinct feature distributions without modifying the original architecture, making the adaptation process lightweight and reusable. The pretrained backbone provides an entropy-efficient latent organization that LoRA refines rather than rebuilds, forming a coupled system with the hyperprior entropy model.
- **An end-to-end rate–distortion compression pipeline for 3DGS semantic features.** We present, to the best of our knowledge, the first end-to-end rate–distortion optimized compression pipeline for 3DGS semantic features. By combining projection-based domain bridging, LoRA adaptation, geometry-guided refinement, and hyperprior entropy coding, our method establishes a principled framework for semantic feature compression under true bitrate constraints.
- **Strong empirical gains across compression and downstream tasks.** Extensive experiments on Replica show that our method achieves a strong rate–distortion trade-off, reaching mIoU $= 0.812$ at bpe $= 3.67$ with minimal loss from the uncompressed upper bound. Ablation studies show that pretrained initialization improves semantic accuracy and reduces the estimated rate compared with random initialization. The compressed features also preserve downstream performance on semantic retrieval and open-vocabulary tasks.

2 Related Work

3D Gaussian Splatting and semantic fields. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) represents scenes as collections of anisotropic Gaussians optimized via differentiable rasterization, achieving real-time novel view synthesis at state-of-the-art quality. Feature 3DGS (Zhou et al., 2024) augments each Gaussian with a high-dimensional semantic feature vector distilled from 2D foundation models (*e.g.*, LSeg (Li et al., 2022), SAM (Kirillov et al., 2023)), enabling open-vocabulary segmentation and language-guided editing. Concurrent works LangSplat (Qin et al., 2024) and LEGaussians (Shi et al., 2024) similarly embed language features into 3DGS, while LERF (Kerr et al., 2023) achieves analogous functionality in the NeRF domain. Despite their semantic capabilities, all these methods store raw high-dimensional features at every Gaussian, leading to substantial memory overhead and motivating the need for efficient compression.

Compression of 3D Gaussian scenes. Existing compression works focus primarily on *geometric* attributes. FCGS (Chen et al., 2025) applies a hyperprior entropy-coding framework (Ballé et al., 2018) to Gaussian scenes, achieving significant bitrate reductions while preserving rendering quality. HAC (Chen et al., 2024) exploits inter-Gaussian context via hash-grid-assisted coding, and EAGLES (Girish et al., 2024) introduces entropy-constrained quantization. For *semantic* compression, DF-3DGS (Dai et al., 2025) decouples the semantic field from the color field and applies a VQ-based adaptive codebook with a scene-specific autoencoder. However, it imposes no entropy constraint and cannot report a principled bits-per-element metric, precluding rate–distortion analysis. Our work is the first to apply a full hyperprior entropy-coding framework to 3DGS semantic features, enabling rigorous rate–distortion evaluation comparable to the image compression literature.

Learned compression and parameter-efficient adaptation. Hyperprior-based models (Ballé et al., 2018) are the standard framework in learned compression, modeling latent distributions with side information and enabling explicit rate–distortion optimization. Widely adopted in systems such as CompressAI (Bégaint et al., 2020; Kamisli et al., 2024), they have primarily been developed for structured 2D signals. In parallel, parameter-efficient fine-tuning methods such as LoRA (Hu et al., 2022) adapt pre-trained networks via low-rank updates to weight matrices, significantly reducing the number of trainable parameters while preserving the original model structure. Originally introduced for large language models and later extended to vision and generative models (Wang et al., 2025; Ruiz et al., 2023), these methods provide an effective mechanism for transferring learned representations across domains. Building on these insights, our work applies hyperprior-based rate–distortion optimization to 3DGS semantic features and adapts geometric compression priors to the semantic domain via low-rank adaptation.

3 Method

3.1 Overview

Given a pretrained Feature-3DGS (Zhou et al., 2024) scene consisting of N Gaussians, each Gaussian is associated with a semantic feature $\boldsymbol{x}_{\text{sem}} \in \mathbb{R}^{D_s}$ distilled from a 2D foundation model. Each Gaussian also carries an attribute vector $\boldsymbol{x}_{\text{attr}} \in \mathbb{R}^{56}$, comprising 48 spherical harmonic appearance coefficients and 8 parameters for opacity, scale, and rotation. This full attribute vector is used to compute the conditioning latent for geometry-guided semantic refinement. To reduce semantic storage, we adapt the pre-trained geometric compressor FCGS (Chen et al., 2025) to the semantic domain through four lightweight components, without modifying its backbone weights. A projection network maps the semantic feature from $D_s = 128$ dimensions to the FCGS encoder input dimension $D_{\text{in}} = 56$. The projected features are processed by a LoRA-augmented FCGS encoder $\mathcal{E}^{\oplus}_{\text{fea}}$. A geometry-guided module $\mathcal{G}$ refines the semantic latent using geometric features, followed by a hyperprior entropy model $\mathcal{C}$ for rate–distortion optimization. The decoder reconstructs the latent via $\mathcal{D}^{\oplus}_{\text{fea}}$ and maps it back to the original space using $\mathcal{D}_{\text{proj}}$. Formally, the complete encoding and decoding pipeline can be summarized as:

$$\begin{aligned} \hat{\boldsymbol{x}}_{\text{sem}} &= \mathcal{D}_{\text{proj}}\big(\mathcal{D}^{\oplus}_{\text{fea}}(\hat{\boldsymbol{y}})\big)\,, \\ \hat{\boldsymbol{y}}, \boldsymbol{b} &= \mathcal{C}\big(\mathcal{G}\big(\mathcal{E}^{\oplus}_{\text{fea}}(\mathcal{E}_{\text{proj}}(\boldsymbol{x}_{\text{sem}}))\,,\, \mathcal{E}_{\text{fea}}(\boldsymbol{x}_{\text{attr}})\big)\big)\,. \end{aligned} \tag{1}$$

where $\oplus$ denotes LoRA augmentation, $\mathcal{G}$ performs attribute-conditioned semantic refinement, and $\mathcal{C}$ produces the quantized semantic latent $\hat{\boldsymbol{y}}$ and bitstream $\boldsymbol{b}$. The conditioning latent is computed from $\boldsymbol{x}_{\text{attr}}$ using the FCGS feature encoder.

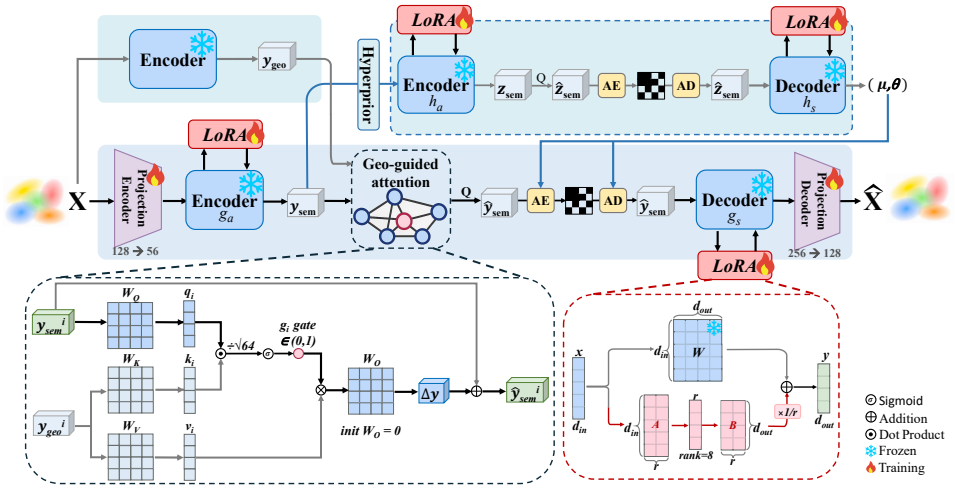

Figure 2: Overview of our semantic feature compression pipeline.Semantic features $\boldsymbol{x}_{\text{sem}}$ are projected to the FCGS input space, then encoded by the LoRA-augmented analysis transform $\mathcal{E}^{\oplus}_{\text{fea}}$. The Geometry-Guided Attention module refines the semantic latent using the geometric latent as a zero-bitrate structural prior (dashed path, detached). The hyperprior branch estimates per-element distribution parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ for entropy coding. FCGS base weights (gray) are frozen; LoRA adapters (purple), projection networks (teal), and geometry attention (coral) are trained.

Reuse of pretrained FCGS transforms. We reuse the pretrained feature analysis and synthesis transforms, $\mathcal{E}_{\text{fea}}$ and $\mathcal{D}_{\text{fea}}$, together with the hyperprior transforms, $\mathcal{E}_{\text{hyp}}$ and $\mathcal{D}_{\text{hyp}}$.

3.2 Projection Network

To bridge the dimensional and statistical mismatch, we learn a nonlinear projection encoder $\mathcal{E}_{\text{proj}} : \mathbb{R}^{D_s} \to \mathbb{R}^{56}$. The projection decoder maps the 256-dimensional output of $\mathcal{D}^{\oplus}_{\text{fea}}$ back to $\mathbb{R}^{D_s}$. Both mappings use three-layer MLPs with Layer Normalization and LeakyReLU activations, providing trainable semantic input and output mappings around the pretrained transforms. Detailed architectures are provided in Appendix A.1.

3.3 Low-Rank Adaptation (LoRA) for Semantic Compression Transfer

The pretrained FCGS transforms provide a starting point for semantic compression, but require adaptation to the statistics of language-aligned features. Keeping these transforms entirely frozen places the burden of domain alignment on the projection networks, while full fine-tuning requires updating all backbone parameters. We therefore introduce Low-Rank Adaptation (LoRA) (Hu et al., 2022) to adapt the compression transforms with a small set of trainable parameters. For each adapted linear layer, we keep the pretrained weight matrix frozen and learn a rank-r residual update:

$$\boldsymbol{W}' = \boldsymbol{W} + \frac{1}{r}\boldsymbol{B}\boldsymbol{A}, \qquad \boldsymbol{A} \in \mathbb{R}^{r \times d_{\text{in}}},\ \boldsymbol{B} \in \mathbb{R}^{d_{\text{out}} \times r}, \tag{2}$$

where $r \ll \min(d_{\text{in}}, d_{\text{out}})$ is the rank, and $1/r$ is a fixed scaling factor. $\boldsymbol{A}$ is initialized with Kaiming uniform sampling and $\boldsymbol{B}$ is initialized to zero, so that $\boldsymbol{W}' = \boldsymbol{W}$ at the start of training and each LoRA-augmented layer matches its pretrained transformation. The feature transforms and hyperprior jointly determine rate–distortion performance: the encoder produces the latent representation, while the hyperprior predicts its coding distributions. We therefore adapt both components so that changes in the semantic latent representation can be accompanied by adjustments to its probability model. Their LoRA parameters are jointly optimized under the rate–distortion objective in Stage 2.

Accordingly, we apply LoRA to all linear layers in $\mathcal{E}_{\text{fea}}$ (4 layers), $\mathcal{D}_{\text{fea}}$ (4 layers), $\mathcal{E}_{\text{hyp}}$ (3 layers), and $\mathcal{D}_{\text{hyp}}$ (3 layers), yielding 14 augmented layers with r=8, totalling 50,624 trainable parameters ($\sim$3% of the FCGS backbone). We examine the contributions of the adaptation components and pretrained initialization in Sec. 4.3.

3.4 GEOMETRY-GUIDED ATTENTION

Semantic features and appearance and geometric attributes are associated with the same Gaussian primitives, providing corresponding attribute information for semantic refinement. We compute the conditioning latent $\boldsymbol{y}_{\text{geo}} = \mathcal{E}_{\text{fea}}(\boldsymbol{x}_{\text{attr}})$ using the frozen FCGS encoder and detach it from the gradient graph. GeoGuide uses this attribute-derived latent to refine the semantic latent before quantization. Since the refined latent is encoded in the semantic stream, no separate conditioning bitstream is required. The refinement is defined as follows:

$$\boldsymbol{q}_i = \boldsymbol{W}_Q\, \boldsymbol{y}_{\text{sem},i}, \qquad \boldsymbol{k}_i = \boldsymbol{W}_K\, \boldsymbol{y}_{\text{geo},i}, \qquad \boldsymbol{v}_i = \boldsymbol{W}_V\, \boldsymbol{y}_{\text{geo},i} \tag{3}$$

$$g_i = \text{sigmoid}\left(\frac{\boldsymbol{q}_i^\top \boldsymbol{k}_i}{\sqrt{d}}\right) \tag{4}$$

$$\tilde{\boldsymbol{y}}_{\text{sem},i} = \boldsymbol{y}_{\text{sem},i} + \boldsymbol{W}_O\,(g_i \cdot \boldsymbol{v}_i) \tag{5}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d\times D_l}$, $\boldsymbol{W}_O \in \mathbb{R}^{D_l\times d}$, d=64, and i indexes individual Gaussian points. The learned scalar gate controls the contribution of the attribute-derived residual for each Gaussian.

Point-wise attention avoids the $\mathcal{O}(N^2)$ complexity, which would be prohibitive at $N \approx 10^6$ Gaussians. The output projection $\boldsymbol{W}_O$ is initialized to zero and remains unchanged during Stage 1 because the geometry-guided module is bypassed and excluded from optimization. Consequently, when the module is first activated in Stage 2, it initially contributes a zero residual, so that $\tilde{\boldsymbol{y}}_{\text{sem}} = \boldsymbol{y}_{\text{sem}}$. The module parameters are subsequently updated during joint rate–distortion optimization. $\boldsymbol{y}_{\text{geo}}$ is detached from the gradient graph throughout, ensuring gradients flow only through the semantic branch and leaving the geometric compression behavior unaffected.

3.5 HYPERPRIOR ENTROPY CODING

Prior semantic compression methods for 3DGS, including DF-3DGS (Dai et al., 2025), report storage reduction in terms of file size after lossless compression of floating-point arrays, without imposing an explicit probabilistic model on the latent representation. We instead use a hyperprior entropy model (Ballé et al., 2018) to estimate and optimize semantic coding cost. The hyperprior analysis transform maps the refined semantic latent to a hyperlatent $\boldsymbol{z}$. After quantization, the hyperlatent is used to predict Gaussian parameters for coding the main latent. The estimated semantic rate includes both streams:

$$R = \underbrace{-\sum_{i=1}^{N} \log_2 p_{\hat{\boldsymbol{y}}_i|\hat{\boldsymbol{z}}_i}(\hat{\boldsymbol{y}}_i)}_{R_y} + \underbrace{-\sum_{i=1}^{N} \log_2 p_{\hat{\boldsymbol{z}}_i}(\hat{\boldsymbol{z}}_i)}_{R_z}. \tag{6}$$

We report the estimated semantic rate as $\text{bpe} = R/(ND_s)$, where N is the number of Gaussians and D_s is the original semantic feature dimension. This likelihood-based estimate includes both latent streams but excludes the Gaussian scene bitstream and model parameter storage, which are reported separately. We vary λ_{rd} to obtain different operating points while keeping the architecture unchanged.

3.6 TRAINING STRATEGY

Stage 1: Projection warm-up. We optimize only the projection encoder and decoder, $\mathcal{E}_{\text{proj}}, \mathcal{D}_{\text{proj}}$, while keeping the FCGS backbone and LoRA parameters frozen. The forward pass omits the Gaussian attribute input, thereby bypassing the geometry-guided module. The hyperprior entropy model is not used in this stage. The projection networks are trained using semantic reconstruction loss:

$$\mathcal{L}_1 = \frac{1}{N}\sum_i w_i \|\hat{\boldsymbol{x}}_{\text{sem},i} - \boldsymbol{x}_{\text{sem},i}\|_2^2 + \lambda_{\text{cos}}\left(1 - \frac{1}{N}\sum_i \cos(\hat{\boldsymbol{x}}_{\text{sem},i}, \boldsymbol{x}_{\text{sem},i})\right) + \lambda_{\text{sm}}\,\mathcal{L}_{\text{smooth}}, \tag{7}$$

Here, $\mathcal{L}_{\text{smooth}}$ penalizes increases in adjacent feature differences under a Morton ordering of Gaussian positions; the reconstruction weights are defined as $w_i = \|\boldsymbol{x}_{\text{sem},i}\|_2 / \left(\frac{1}{N}\sum_{j=1}^{N} \|\boldsymbol{x}_{\text{sem},j}\|_2\right)$. The smoothness penalty and training hyperparameters are provided in Appendix A.2.

Stage 2 enables geometry-guided refinement by supplying $\boldsymbol{x}_{\mathrm{attr}}$ and jointly optimizes the projection networks, LoRA adapters, geometry-guided module, and trainable entropy-model parameters under:

$$\mathcal{L}_2 = \mathcal{L}_1 + \lambda_{\mathrm{rd}} \frac{R}{N \cdot D_s}. \tag{8}$$

We use AdamW (Loshchilov & Hutter, 2019) with cosine annealing; Stage 1 runs for 800 steps and Stage 2 for 1,500 steps, taking $\sim$5 minutes per scene on a single A100.

Cross-scene generalization. For target-scene adaptation, we load the source checkpoint and continue optimizing only its projection networks for 700 steps, keeping the remaining components fixed. Source and target configurations are specified in the transfer experiments and Appendix A.5.

4 Experiment

4.1 Experimental Setup

Datasets and representations. We evaluate on Replica indoor scenes and real-world scenes from the LERF dataset (Kerr et al., 2023). The main rate–distortion experiments use `room_0` and `office_3` from Replica, with 128-dimensional per-Gaussian semantic features obtained through the LSeg-based Feature-3DGS pipeline (Li et al., 2022; Zhou et al., 2024). We additionally evaluate our framework on 512-dimensional CLIP features from DF-3DGS, obtaining an mIoU of 0.764 compared with 0.774 for the reference. Details are provided in Appendix E.

Evaluation protocols. The main Replica experiments use 90 evaluation views for `room_0` and seven evaluation views (`00003`–`00009`) for `office_3`. Each method is evaluated against the corresponding uncompressed reference using the same views and semantic evaluation procedure. On LERF, we report per-Gaussian reconstruction metrics and qualitative language-query results. Transfer experiments distinguish zero-shot evaluation, which performs no target-scene optimization, from lightweight adaptation, which updates only the projection networks for 700 steps. The source and target scenes are specified alongside the corresponding transfer results.

Baselines. We use uncompressed Feature-3DGS as the reference and compare against truncated SVD at multiple ranks and two FCGS-based baselines. **FCGS-Direct** is a frozen-transform diagnostic baseline: it truncates each 128-dimensional semantic vector to its first 56 channels, applies the pretrained FCGS feature encoder, rounds the latent, and passes it through the feature decoder, retaining the first 128 of its 256 output channels. No learned adapters, LoRA updates, semantic retraining, or original FCGS masking/skip routing are used. **FCGS-Semantic-RD** instead uses learned linear input and output adapters ($128 \to 56$ and $256 \to 128$) and fine-tunes the pretrained compression transforms with a hyperprior entropy model under a semantic rate–distortion objective. It does not include our nonlinear projection networks, LoRA, or geometry-guided module. We additionally include DF-3DGS (Dai et al., 2025) without scene-specific compression in the scene-storage comparison.

Metrics and bitrate accounting. We report mIoU for rendered semantic segmentation and per-Gaussian MSE and cosine similarity for feature reconstruction. PSNR and MS-SSIM compare rendered semantic feature maps from the decoded compressed representations with those from the uncompressed references on the same evaluation views. For our method and FCGS-Semantic-RD, semantic BPE is estimated from the main-latent and hyperlatent likelihoods as $(R_y + R_z)/(ND_s)$. For rank-k SVD, BPE is computed from the explicit FP32 payload of the retained factors:

$$\mathrm{BPE}_{\mathrm{SVD}} = \frac{32(Nk + k + kD_s)}{ND_s}. \tag{9}$$

These semantic rates exclude geometric storage and model overhead, which are reported separately in the storage analysis.

4.2 Quantitative Comparisons

Rate–semantic performance. Figure 3 compares segmentation performance across semantic bitrates on Replica room0 and office3. On room0, our method achieves 0.8124 mIoU at 3.67 BPE,

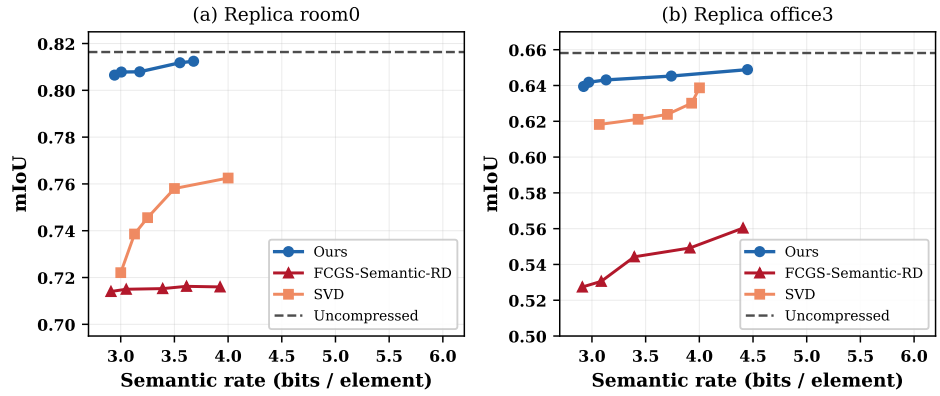

Figure 3: Rate–semantic performance on Replica room0 and office3. Each marker represents an evaluated operating point, and the dashed horizontal lines indicate the uncompressed mIoU references. Semantic rate is measured in bits per original semantic feature element: likelihood-based estimates for Ours and FCGS-Semantic-RD, and explicit FP32 factor storage for SVD. These rates exclude the Gaussian scene bitstream and model parameter storage. Our method maintains segmentation performance close to the uncompressed references across the evaluated rate range.

only 0.39 percentage points below the uncompressed reference of 0.8164. On office3, it achieves 0.6489 mIoU at 4.45 BPE, a decrease of 0.93 percentage points from the reference of 0.6582. At the lowest evaluated rates, our method retains 0.8065 and 0.6395 mIoU at 2.94 and 2.92 BPE, respectively. These results demonstrate effective preservation of downstream semantic performance across multiple operating points.

Comparison with semantic-adapted FCGS. On room0, our method achieves 0.812 mIoU at 3.67 BPE, compared with 0.716 at 3.61 BPE for FCGS-Semantic-RD: a gain of 9.62 percentage points with a 1.81% higher rate. On office3, the corresponding results are 0.649 at 4.45 BPE and 0.560 at 4.41 BPE, yielding an 8.85-point gain with a rate difference below 1%. The advantage therefore persists against a semantic-adapted FCGS baseline, beyond the naive frozen-transform diagnostic. These results assess the complete framework. Component ablations are presented in Sec. 4.3.

Comparison with SVD. Our method also achieves higher segmentation accuracy than SVD at comparable or lower reported semantic rates. On room0, it improves mIoU from 0.762 to 0.812 while reducing the rate from 4.00 to 3.67 BPE. On office3, our method obtains 0.645 mIoU at 3.74 BPE, exceeding SVD's 0.6387 at 4.00 BPE.

Reconstruction rate–distortion performance. We additionally evaluate PSNR and MS-SSIM between semantic feature maps rendered from the compressed and uncompressed representations. The evaluation uses the complete pipeline, including compressed Gaussian attributes and semantic features. Across the evaluated operating points, our method achieves PSNR values of 46.89–47.07 dB on room0 and 47.83–48.11 dB on office3. These reconstruction metrics complement the downstream semantic evaluation. Additional reconstruction rate–distortion curves for `room_0` are provided in Appendix C.

Storage accounting. Table 1 reports representative segmentation and storage results on room0. At 3.67 BPE on room0, our saved representation occupies 75.11 MB: 55.65 MB for semantic bitstreams, 18.35 MB for the Gaussian scene, and 1.11 MB for scene-specific model parameters. The 3.09 MB pretrained backbone is shared across scenes and counted separately. Storage is measured from saved files, whereas BPE is estimated from latent likelihoods.

4.3 Ablation Studies

Table 7 examines the contributions of semantic projection, joint adaptation, geometry-guided refinement, and pretrained initialization on room0. Rate–distortion variants use $\lambda_{\text{rd}} = 10^{-4}$, while the Stage 1 variant is evaluated without entropy coding.

Table 1: Quantitative comparison on Replica `room_0`. Ours uses $\lambda_{\text{rd}} = 10^{-4}$. Storage includes Gaussian and semantic bitstreams and scene-specific parameters; the shared backbone is excluded. Semantic rate definitions are given in Sec. 4.1.

Method	mIoU↑	cos↑	MSE↓	bpe↓	Storage↓	Generalizable
Feature 3DGS (Zhou et al., 2024)	0.816	1.000	0.000	—	718 MB	—
FCGS-Direct	0.014	—	—	—	—	—
FCGS-Semantic-RD	0.716	—	—	3.61	—	—
SVD (k=16)	0.762	—	—	4.00	84 MB	✗
DF-3DGS (w/o SC) (Dai et al., 2025)	0.774	—	—	—	57 MB‡	✗
Ours (Stage 1)	0.734	0.951	0.018	—	—	✓
Ours (full, λ=1×10^{-4})	**0.812**	**0.960**	**0.010**	**3.67**	**75.11 MB**	✓

‡ DF-3DGS uses approximately 38K semantic Gaussians in a decoupled field, whereas our representation contains approximately 990K Gaussians. The w/o SC variant omits the scene-specific autoencoder.

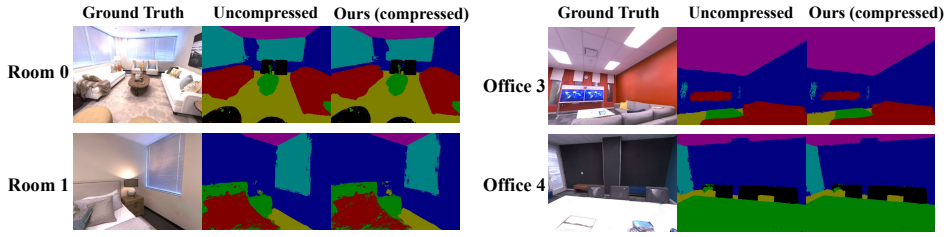

Figure 4: **Qualitative semantic segmentation results** on four Replica scenes. For each scene, we show the RGB image, predictions from the uncompressed Feature-3DGS representation, and predictions from our compressed representation.

Semantic projection. The projection warm-up variant achieves an mIoU of 0.734, compared with 0.014 for FCGS-Direct. This demonstrates the effectiveness of learning semantic input and output mappings around the pretrained feature transforms. Under LoRA adaptation and entropy modeling, the frozen-projection variant obtains 0.628 mIoU at 3.25 BPE, whereas the trainable-projection variant reaches 0.761 at 2.98 BPE. This comparison supports retaining trainable projection mappings during semantic-domain adaptation.

Joint adaptation and geometry guidance. Starting from the jointly optimized projection, LoRA, and entropy-model configuration, adding GeoGuide increases mIoU from 0.761 to 0.812 while increasing the rate from 2.98 to 3.67 BPE. Geometry-guided refinement therefore improves semantic accuracy at the evaluated setting, with an accompanying rate increase.

Pretrained initialization. Replacing the pretrained backbone with a randomly initialized variant reduces mIoU from 0.812 to 0.560 and increases the rate from 3.67 to 5.35 BPE. The pretrained variant thus achieves both higher semantic accuracy and a lower rate in this comparison. These results support the value of pretrained initialization for the proposed adaptation framework.

4.4 Cross-scene and Cross-dataset Transfer

Cross-scene transfer within Replica. We first evaluate transfer from a model trained on `room_0`. For each target scene, we freeze the learned LoRA, geometry-guided attention, and entropy-model parameters and optimize only the projection networks for 700 steps. Table 2 reports the resulting segmentation performance.

Cross-dataset transfer. A separate source model is trained on `room_0`, `room_1`, and `office_3`. We evaluate zero-shot transfer to unseen `office_4` and LERF `teatime`, and additionally adapt the projection networks on `teatime` for 700 steps while freezing the remaining components. Both datasets use 128-dimensional LSeg-derived features. As shown in Table 3, projection-only adaptation on teatime increases cosine similarity from 0.628 to 0.966 and reduces MSE from 0.073 to 0.013, while the estimated semantic rate increases from 2.90 to 3.54 BPE.

Table 2: Cross-scene segmentation transfer from `room_0`. Target scenes use 700 steps of projection-only adaptation.

Scene	Reference	Ours	Δ	Steps
`room_0` (source)	0.816	0.812	−0.004	2300
`room_1`	0.579	0.544	−0.035	700
`office_3`	0.658	0.648	−0.010	700
`office_4`	0.776	0.745	−0.031	700
Average	0.707	0.687	−0.020	—

Table 3: Transfer from the multi-scene Replica model. Proj. adapt. updates only the projection networks for 700 steps.

Target scene	Setting	Cos.↑	MSE↓	BPE↓
Replica office4	Zero-shot	0.824	0.075	2.92
LERF teatime	Zero-shot	0.628	0.073	2.90
LERF teatime	Proj. adapt.	**0.966**	**0.013**	3.54

These results demonstrate improved target-scene reconstruction without updating the source-trained compression components. Qualitative semantic predictions are provided in Appendix D.

4.5 Downstream Applications

We evaluate compressed semantic features on three downstream tasks of increasing complexity: *(i)* Open-vocabulary semantic segmentation tests feature discriminability and language grounding; *(ii)* Semantic retrieval probes the preservation of global feature directions for ranking; and *(iii)* Language-guided editing requires precise 3D spatial grounding for effective scene manipulation.

Semantic segmentation and open-vocabulary querying. Figure 4 compares semantic segmentation maps rendered from the uncompressed and compressed representations across 4 Replica scenes. The compressed representations retain the major semantic regions visible in the uncompressed predictions. We further visualize predictions with user-defined text labels in Figure 5 in Appendix, illustrating that reconstructed features remain usable for open-vocabulary scene querying.

Semantic retrieval. For each text query, we rank Gaussians by similarity in the language-aligned feature space and measure the overlap between the top-K retrieved sets from the compressed and uncompressed representations. This metric evaluates retrieval consistency with the uncompressed reference. The detailed results are reported in Table 6 in the appendix. Our full model improves over the variant without GeoGuide at all three evaluated values of K. At $K = 2000$, it achieves an overlap of 0.744, compared with 0.716 without GeoGuide and 0.750 for SVD. SVD obtains higher overlap scores in this evaluation, highlighting that retrieval consistency and segmentation accuracy capture different aspects of semantic preservation.

Language-guided scene editing. We also use text queries to select target Gaussians for attribute modification. Figure 6 in appendix presents sofa recoloring and table removal on room0, with removal implemented through opacity modification. The edited scenes are rendered from multiple viewpoints, illustrating how the reconstructed semantic features support object selection and subsequent manipulation in the 3D representation.

5 Conclusion

We revisit semantic feature compression in 3DGS through parameter-efficient adaptation of a pretrained compression model. Our end-to-end rate–distortion optimized framework combines semantic projection with LoRA to adapt pretrained FCGS transforms to language-aligned features while keeping the backbone weights frozen. Geometry-guided refinement incorporates Gaussian attribute information at the encoder without requiring a separate conditioning bitstream, while a hyperprior entropy model provides explicit rate control. Experiments on Replica demonstrate near-uncompressed segmentation performance at compact semantic rates and substantial gains over a semantic-adapted FCGS baseline. Transfer experiments further show that projection-only adaptation can recover feature reconstruction quality on target scenes, including real-world LERF scenes. These results support parameter-efficient adaptation as an effective approach to reusing pretrained 3DGS compression models for semantic representations.

AI USE STATEMENT

We used ChatGPT to assist with translation, organization, language editing, and grammar correction. AI tools were also used to edit formatting. The authors reviewed the AI-assisted material and take responsibility for the final text, claims, and figures.

REPRODUCIBILITY STATEMENT

Section 3 describes the compression pipeline and training objectives. Appendix A provides network architectures, training configurations, baseline implementations, evaluation protocols, and transfer settings. Section 4.1 and Appendix A.4 specify the distinction between likelihood-based semantic rates and saved-file storage.

REFERENCES

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.

Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pp. 422–438. Springer, 2024.

Yihang Chen, Qianyi Wu, Mengyao Li, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Fast feed-forward 3d gaussian splatting compression. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhenqi Dai, Ting Liu, and Yanning Zhang. Efficient decoupled feature 3d gaussian splatting via hierarchical compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11156–11166, 2025.

Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pp. 54–71. Springer, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Fatih Kamisli, Fabien Racapé, and Hyomin Choi. Variable-rate learned image compression with multi-objective optimization and quantization-reconstruction offsets. In *2024 Data Compression Conference (DCC)*, 2024.

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.

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19729–19739, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RriDjddCLN.

Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, et al. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4961–4972, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5333–5343, 2024.

Han Wang, Yongjie Ye, Bingru Li, Yuxiang Nie, Jinghui Lu, Jingqun Tang, Yanjie Wang, and Can Huang. Vision as lora. *arXiv preprint arXiv:2503.20680*, 2025.

Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024.

Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21676–21685, 2024.

A Implementation and Evaluation Details

A.1 Network Architecture

We describe the architecture used for the 128-dimensional LSeg-based semantic features. Each Gaussian carries a semantic vector $\boldsymbol{x}_{\text{sem}} \in \mathbb{R}^{128}$ and an attribute vector $\boldsymbol{x}_{\text{attr}} \in \mathbb{R}^{56}$. The latter contains 48 spherical harmonic coefficients, one opacity parameter, three scale parameters, and four rotation parameters. Gaussian positions are used separately for spatial ordering and are not included in this 56-dimensional conditioning vector. Table 4 summarizes the feature transforms and adaptation modules. The projection encoder maps semantic features to the native 56-dimensional input of the FCGS feature encoder. On the reconstruction side, the projection decoder takes the full 256-dimensional output of the FCGS feature decoder and maps it back to the semantic space. The original FCGS attribute-prediction heads are not used in the semantic path.

Low-rank adaptation. LoRA is applied to four linear layers in the feature encoder, four in the feature decoder, three in the hyperprior encoder, and three in the hyperprior decoder. The original attribute-prediction heads are outside this set. The resulting 14 adapted layers contain 50,624 LoRA parameters. This count refers only to the low-rank updates and excludes the projection networks, GeoGuide, and separately trainable entropy-model parameters. All pretrained base weights remain frozen during adaptation. For each update $\boldsymbol{BA}/r$, $\boldsymbol{A}$ is initialized with Kaiming uniform initialization and $\boldsymbol{B}$ is initialized to zero.

Attribute conditioning. The conditioning latent, denoted by $\boldsymbol{y}_{\text{geo}}$ in the main paper, is computed from the full Gaussian attribute vector using the frozen FCGS feature encoder. It is detached from the gradient graph. GeoGuide uses 64-dimensional queries, keys, and values to construct a gated residual in the 256-dimensional semantic latent space. The query, key, and value projection weights are

Table 4: Architecture of the semantic adaptation modules and the reused FCGS feature transforms. Dimensions describe successive linear layers.

Component	Architecture or setting
Semantic input dimension	128
Gaussian attribute input dimension	56
Projection encoder	$128 \to 128 \to 128 \to 56$
Projection decoder	$256 \to 256 \to 128 \to 128$
Projection hidden-layer operations	LayerNorm and LeakyReLU (0.1)
FCGS feature encoder	$56 \to 256 \to 256 \to 256 \to 256$
FCGS feature decoder	$256 \to 256 \to 256 \to 256 \to 256$
LoRA rank and scaling	$r = 8$, scaling $1/r$
LoRA-augmented linear layers	14
LoRA parameters	50,624
GeoGuide query, key, and value projections	$256 \to 64$, without bias
GeoGuide output projection	$64 \to 256$, without bias
Projection parameters	156,728
GeoGuide parameters	65,536

initialized with Xavier initialization, while the output projection $\boldsymbol{W}_O$ is initialized to zero. Refinement is performed before semantic quantization; the semantic decoder does not require a separately transmitted conditioning latent.

A.2 Training Configuration

We use two-stage optimization, with the settings summarized in Table 5. The projection networks are initialized with Xavier uniform initialization for source-model training.

Stage 1: projection warm-up. Only the projection encoder and decoder are optimized. The pretrained FCGS weights and LoRA parameters remain frozen. The forward pass does not receive the attribute-conditioning input, so GeoGuide is bypassed and the hyperprior entropy model is disabled. This stage optimizes the semantic reconstruction objective defined in the main paper, without a rate term.

Stage 2: joint rate–distortion optimization. We supply the Gaussian attribute input and jointly optimize the projection networks, LoRA adapters, GeoGuide, and trainable entropy-model parameters. The reconstruction objective is augmented with the estimated semantic rate, normalized by the number of original semantic elements. GeoGuide initially contributes a zero residual because its output projection remains at its zero initialization throughout Stage 1.

Spatial consistency. Morton ordering is constructed from the three-dimensional Gaussian positions. The smoothness penalty compares changes in reconstructed and original semantic features along this ordering, penalizing increases beyond the margin $\epsilon = 0.5$. Its weight is $\lambda_{\mathrm{sm}} = 0.05$. The coordinates used for Morton ordering are distinct from the 56-dimensional appearance and geometric attributes used by GeoGuide. Let the Gaussian indices follow the Morton ordering of their 3D positions. We use

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \max(0,\, \|\hat{\boldsymbol{x}}_{\mathrm{sem},i+1} - \hat{\boldsymbol{x}}_{\mathrm{sem},i}\|_2 - \|\boldsymbol{x}_{\mathrm{sem},i+1} - \boldsymbol{x}_{\mathrm{sem},i}\|_2 - \epsilon)\,, \tag{10}$$

with $\epsilon = 0.5$ and $\lambda_{\mathrm{sm}} = 0.05$.

A.3 Baseline Configurations

FCGS-Direct. This baseline diagnoses direct use of frozen pretrained transforms without semantic adaptation. Each 128-dimensional semantic vector is truncated to its first 56 channels, passed

Table 5: Training settings for the main LSeg-based experiments. Learning rates are initial values. The entropy auxiliary optimizer is separate from the main AdamW optimizer.

Setting	Stage 1	Stage 2
Optimization steps	800	1,500
Batch size (Gaussians)	50,000	50,000
Projection networks	Trainable	Trainable
FCGS base weights	Frozen	Frozen
LoRA parameters	Frozen	Trainable
GeoGuide	Bypassed	Trainable
Entropy model	Disabled	Enabled
Main optimizer	AdamW	AdamW
Weight decay	10^{-4}	10^{-5}
Learning-rate schedule	Cosine annealing	Cosine annealing
Projection learning rate	5×10^{-4}	10^{-4}
LoRA learning rate	—	10^{-3}
GeoGuide learning rate	—	2.5×10^{-4}
Entropy learning rate	—	10^{-4}
Entropy auxiliary optimizer	—	Adam
Entropy auxiliary learning rate	—	10^{-3}
λ_{cos}	10	10
λ_{sm}	0.05	0.05
Smoothness margin ϵ	0.5	0.5
λ_{rd}	—	$\{1, 2, 4, 8, 16\} \times 10^{-4}$

through the pretrained FCGS feature encoder, and quantized by rounding. The frozen feature decoder reconstructs a 256-dimensional vector, whose first 128 channels are retained as the semantic reconstruction. No learned input/output adapters, LoRA updates, or semantic retraining are used. The original FCGS masking and direct-quantization skip routing are not executed in this diagnostic path.

FCGS-Semantic-RD. This baseline introduces linear input and output adapters with dimensions $128 \to 56$ and $256 \to 128$. It initializes the FCGS compression transforms from pretrained weights and fine-tunes them using semantic reconstruction losses and a likelihood-based rate term. It uses full fine-tuning of the compression transforms rather than LoRA and does not include our nonlinear projection networks or geometry-guided refinement. The rate includes both main-latent and hyperlatent likelihoods and is normalized by ND_s, as for our method.

SVD. Given the semantic feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times D_s}$, we form a rank-$k$ approximation

$$\boldsymbol{X} \approx \boldsymbol{U}_k \operatorname{diag}(\boldsymbol{s}_k) \boldsymbol{V}_k^\top, \tag{11}$$

where $\boldsymbol{U}_k \in \mathbb{R}^{N \times k}$, $\boldsymbol{s}_k \in \mathbb{R}^k$, and $\boldsymbol{V}_k^\top \in \mathbb{R}^{k \times D_s}$. All retained factors are stored in FP32, giving

$$R_{\mathrm{SVD}} = 32(Nk + k + kD_s) \tag{12}$$

bits and a semantic rate of $R_{\mathrm{SVD}}/(ND_s)$. This is an explicit factor-payload rate, whereas our method and FCGS-Semantic-RD report likelihood-based rate estimates. These semantic rates exclude Gaussian scene storage and model parameter overhead.

A.4 Evaluation Protocol

Semantic prediction. In the LSeg-based Feature-3DGS representation, each Gaussian stores a 128-dimensional semantic feature. After decompression, reconstructed features are rasterized using the Feature-3DGS renderer. For semantic prediction, the rendered 128-dimensional feature map is mapped to the 512-dimensional LSeg feature space by the fixed 1×1 feature decoder. The compressed representation and its uncompressed reference use the same semantic prediction procedure. The main Replica evaluation uses 90 evaluation views for `room_0` and seven evaluation

views (00003–00009) for office_3. Within each scene, comparisons use the corresponding evaluation-view set and uncompressed reference.

Reconstruction and task metrics. We report mIoU for rendered semantic segmentation. Per-Gaussian MSE and cosine similarity measure feature reconstruction quality. PSNR and MS-SSIM compare rendered semantic feature maps from the decoded compressed representation with those from the uncompressed reference on the same evaluation views. For our method, this evaluation uses the complete compression pipeline, including compressed Gaussian attributes and semantic features.

Rate and storage. For the learned semantic codecs, BPE is computed as $(R_y + R_z)/(ND_s)$ using the main-latent and hyperlatent likelihoods, with $D_s = 128$ in the main experiments. We report saved-file storage separately from this estimated semantic rate. Storage accounting distinguishes semantic bitstreams, Gaussian scene bitstreams, scene-specific model parameters, and the shared pretrained backbone.

A.5 Transfer Settings

We distinguish the single-source Replica transfer experiment from the multi-scene source model used for cross-dataset evaluation.

Single-source transfer within Replica. The source model is trained on room_0. For target-scene adaptation, we load the source checkpoint, including its projection weights, and continue optimizing only the projection networks for 700 steps. The pretrained FCGS weights, LoRA adapters, GeoGuide, and entropy-model parameters remain fixed. The target projection networks are therefore initialized from the source model rather than randomly reinitialized.

Multi-scene transfer to LERF. A separate source model is trained on the Replica scenes room_0, room_1, and office_3. The evaluated target scenes include Replica office_4 and LERF teatime, neither of which is included in this source training set. These experiments use 128-dimensional LSeg-derived per-Gaussian semantic features. Zero-shot evaluation applies all source-trained components, including the projection networks, without target-scene parameter updates. For the adapted teatime setting, we load the multi-scene checkpoint and continue optimizing only the projection networks for 700 steps. All remaining compression components stay fixed. Reconstruction quality and estimated bitrate are reported separately before and after adaptation, since updating the projection networks changes the latents supplied to the frozen compression components.

B Additional Experimental Results

B.1 Open-Vocabulary Querying

Figure 5 presents semantic predictions obtained from reconstructed features using user-defined text labels on four Replica scenes. Complementing the segmentation comparisons in the main paper, these visualizations illustrate how the compressed representations support language-based scene querying without query-specific retraining.

B.2 Semantic Retrieval Results

We evaluate whether compressed semantic features preserve the retrieval results of the uncompressed representation. For a text query q, let $S_K^{\mathrm{comp}}(q)$ and $S_K^{\mathrm{ref}}(q)$ denote the sets of top-K Gaussian indices retrieved from the compressed and uncompressed representations, respectively. We define

$$\mathrm{Overlap@}\,K(q) = \frac{\left|S_K^{\mathrm{comp}}(q) \cap S_K^{\mathrm{ref}}(q)\right|}{K}. \tag{13}$$

This metric measures agreement with the uncompressed retrieval reference, rather than retrieval accuracy against ground-truth relevance labels.

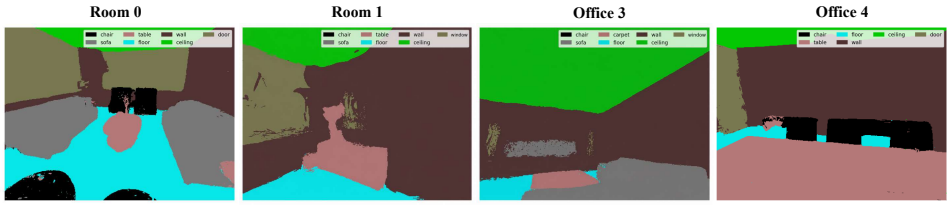

Figure 5: **Open-vocabulary querying** with reconstructed semantic features on four Replica scenes. Each scene is queried with a user-defined label set, such as "chair", "sofa", and "carpet", without additional training for the displayed queries.

Table 6: Semantic retrieval consistency on `room_0`, measured by top-K set overlap with the uncompressed reference. Bold indicates the highest score among compressed variants.

Method	Overlap@500 ↑	Overlap@1000 ↑	Overlap@2000 ↑
Uncompressed reference	1.000	1.000	1.000
SVD ($k = 16$)	**0.597**	**0.673**	**0.750**
Ours w/o GeoGuide	0.533	0.626	0.716
Ours (full)	0.551	0.654	0.744

Table 6 reports retrieval consistency at three retrieval scales. Our full model improves over the variant without GeoGuide across all evaluated values of K, increasing Overlap@2000 from 0.716 to 0.744. At this retrieval scale, its overlap is close to that of SVD (0.750). These results complement the segmentation experiments by evaluating the consistency of text-based retrieval after semantic feature compression.

B.3 Language-guided Scene Editing

We provide additional qualitative results for language-guided scene editing tasks. Given free-form language instructions, target Gaussians are identified via cosine similarity in the compressed semantic feature space, followed by attribute modification such as color editing or object removal.

As shown in Figure 6, our method preserves spatial grounding under compression and produces consistent editing results across multiple viewpoints, demonstrating that semantic structure remains intact even after aggressive rate–distortion optimization.

B.4 Ablation Studies

Table 7 reports component ablations on Replica `room_0`. Checkmarks indicate that a component is present; in the frozen-projection variant, the projection networks are retained but not updated during Stage 2. The Stage 1 variant bypasses entropy coding and therefore has no reported entropy-based rate.

C Additional Rate–Distortion Results

We provide additional reconstruction rate–distortion results on Replica `room_0`, complementing the rate–semantic comparisons in the main paper. Figure 7 reports PSNR and MS-SSIM between rendered semantic feature maps from the compressed representations and their uncompressed references. The evaluation protocol and bitrate accounting are described in Appendix A.4.

Reconstruction quality across rates. Our method maintains PSNR between 46.89 and 47.07 dB over the evaluated semantic rate range of approximately 2.94–3.67 BPE. FCGS-Semantic-RD exhibits a larger PSNR variation, from 41.66 to 46.28 dB over approximately 2.91–3.93 BPE. The slight non-monotonicity at its higher-rate operating points is retained as measured. The MS-SSIM curves provide an additional characterization of rendered feature reconstruction. Together with the

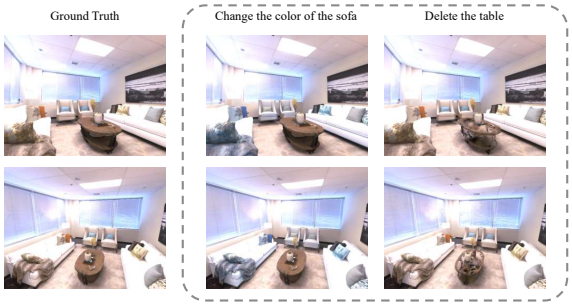

Figure 6: **Language-guided scene editing** on `room_0`. Left: original scene. Center: color modification ("change the color of the sofa"). Right: object removal ("delete the table"). Two novel viewpoints are shown per operation. Compressed semantic features accurately localize target objects and support editing operations consistent across viewpoints.

Table 7: Component ablations on Replica `room_0`. Rate–distortion variants use $\lambda_{\mathrm{rd}} = 10^{-4}$. BPE denotes the estimated semantic rate.

Configuration	Proj.	LoRA	Geo.	Entropy	mIoU↑	BPE↓
FCGS-Direct	✗	✗	✗	✗	0.014	—
+ Projection (Stage 1)	✓	✗	✗	✗	0.734	—
Frozen projection	✓	✓	✗	✓	0.628	3.25
+ LoRA + Entropy	✓	✓	✗	✓	0.761	2.98
+ GeoGuide (full)	✓	✓	✓	✓	**0.812**	3.67
Random-init backbone	✓	✓	✓	✓	0.560	5.35

mIoU results in the main paper, these metrics assess reconstruction fidelity and downstream semantic utility from complementary perspectives.

D Additional Transfer Results and LERF Visualizations

Transfer setting. We further examine cross-dataset transfer on the `teatime` scene from LERF. The source model is trained on three Replica scenes: `room_0`, `room_1`, and `office_3`. The target scene, `teatime`, is excluded from source-model training. Starting from the source checkpoint, we continue optimizing only the projection encoder and decoder on the target scene for 700 steps. The pretrained FCGS backbone, LoRA adapters, geometry-guided module, and entropy-model parameters remain fixed. This setting evaluates whether the source-trained compression components can be reused across datasets through lightweight projection adaptation.

Qualitative results. Figure 8 compares semantic predictions from the uncompressed and adapted compressed representations at the same viewpoint. The compressed representation retains the broad spatial layout of the uncompressed predictions, with visible differences in some local regions. This comparison complements the feature reconstruction results reported in the main paper by illustrating semantic prediction consistency after adaptation. The uncompressed prediction serves as a reference, rather than ground-truth annotation; this visualization does not constitute a quantitative evaluation of segmentation accuracy on LERF.

E Applicability to Other Semantic Representations

The main experiments use 128-dimensional LSeg features stored in Feature-3DGS. To examine applicability beyond this representation, we additionally apply our compression framework to 512-dimensional CLIP features from DF-3DGS on Replica `room_0`. This experiment changes both the source semantic representation and the feature dimensionality, while retaining the FCGS-based

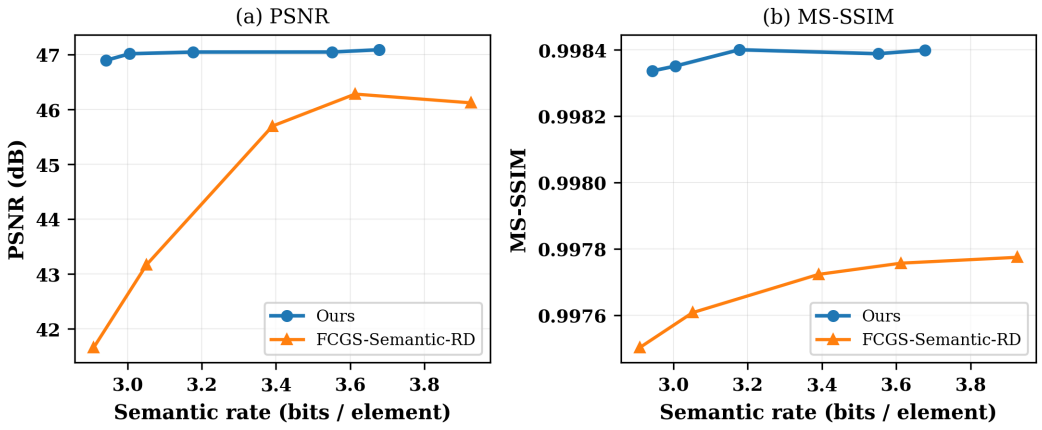

Figure 7: Reconstruction rate–distortion results on Replica `room_0`. Panels show (a) PSNR and (b) MS-SSIM of rendered semantic feature maps relative to the uncompressed reference. Each marker represents an evaluated operating point. Semantic rate is estimated from the main-latent and hyperlatent likelihoods and expressed in bits per original semantic feature element. It excludes the Gaussian scene bitstream and model parameter storage. Higher values indicate better reconstruction quality for both metrics.

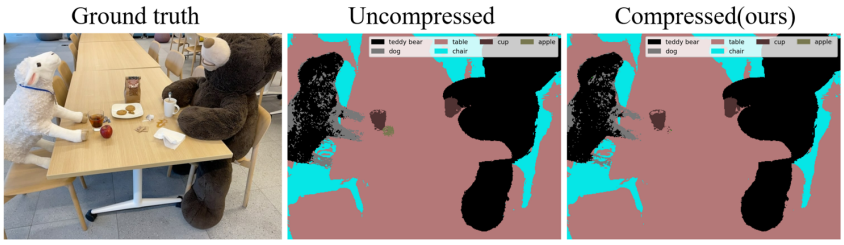

Figure 8: Qualitative semantic predictions on the LERF teatime scene. From left to right: the RGB image, predictions from the uncompressed representation, and predictions from our compressed representation after 700 steps of projection-only adaptation. The source-trained LoRA adapters, geometry-guided module, and entropy model remain fixed during adaptation. Colors indicate the semantic categories listed in the legends.

compression framework. The CLIP feature mapping uses a 64-dimensional bottleneck to reconstruct the original 512-dimensional features.

As shown in Table 8, the compressed representation achieves an mIoU of 0.764, compared with 0.774 for the reference, corresponding to a decrease of 1.0 percentage point. The reconstructed features retain a cosine similarity of 0.984 with the reference. These results provide evidence that the framework is applicable beyond the Feature-3DGS/LSeg configuration used in the main experiments, extending to the evaluated DF-3DGS/CLIP representation.

F LIMITATIONS

Our framework typically requires lightweight projection adaptation for new scenes, which introduces a per-scene optimization cost. Our experiments cover indoor scenes from Replica and selected real-world scenes from LERF; applicability to large-scale outdoor environments remains to be explored.

Table 8: Compression of DF-3DGS CLIP features on Replica `room_0`. The reference uses the original 512-dimensional features before applying our compression. Cosine similarity measures agreement between reconstructed and reference features.

Representation	Feature dim.	Bottleneck	mIoU ↑	Cosine ↑
DF-3DGS feature reference	512	–	0.774	–
Ours on DF-3DGS features	512	64	0.764	0.984