Autoencoding Gaussian Splats in 2D

Rok Mokotar*
LMU Munich
Rok.Mokotar@campus.lmu.de

Federico Bernardo Harjes Ruiloba*
LMU Munich
f.harjes@campus.lmu.de

Abstract

TODO - at the end :D

1. Introduction

Recent advances in neural rendering and generative modeling have demonstrated the effectiveness of Gaussian splatting for representing images and 3D scenes. In this work, we explore the novel task of autoencoding for Gaussian splats, where we leverage autoencoders (AEs) to learn compact representations of images modeled as 2D Gaussian splats. Our primary objectives are twofold: *(1)* to construct and train an autoencoder for Gaussian splat representations of images and *(2)* to compare its performance against a standard autoencoder trained directly on raw image data.

The importance of this problem arises from the growing need for efficient and flexible representations of visual data. Gaussian splats offer a structured and parametric alternative to pixel-based representations, encapsulating key image features such as position, scale, rotation, opacity, and color in a compact form. This representation aligns well with modern neural rendering techniques and has the potential to enhance interpretability, adaptability, and compression efficiency in learned visual representations.

We adopt Gaussian splats and autoencoders for this task due to their complementary strengths. Gaussian splatting has proven to be a powerful representation technique for reconstructing visual data with high fidelity while being inherently adaptable to multi-resolution settings. Autoencoders, on the other hand, provide a well-established framework for extracting meaningful latent representations and reducing data dimensionality in an unsupervised manner. By combining these approaches, we aim to investigate whether autoencoding in the Gaussian splat space can yield benefits in terms of compression efficiency, reconstruction quality, and feature disentanglement compared to standard pixel-based autoencoding.

A key focus of this work is on the compression capabilities of Gaussian splats. Unlike traditional pixel-based approaches, where compression often relies on feature extraction in a high-dimensional space, Gaussian splats inherently provide a more structured, lower-dimensional representation of images. This opens up opportunities for novel encoding strategies that take advantage of the underlying parametric nature of the splat representation.

Our contributions in this study include:

- The creation of a dataset consisting of trained Gaussian splats derived from CIFAR-10 images, enabling further research on learned splat representations.
- The design and implementation of an autoencoder specifically tailored for Gaussian splats, exploring different architectural choices and their impact on reconstruction quality.
- A comparative analysis of Gaussian splat-based autoencoding against conventional pixel-based autoencoding to evaluate reconstruction performance and compression efficacy.

By addressing these objectives, we aim to provide insights into the potential of Gaussian splats as a learned visual representation and contribute to the broader research on neural compression and generative modeling.

The remainder of this report is structured as follows: Section 2 provides essential background on Gaussian splatting and autoencoders. Section 3 details our dataset, splatting process, autoencoder design, and experimental setup. Section 4 presents our findings on different splatting techniques, autoencoder architectures, and various evaluation approaches. Section 5 interprets the results, highlighting key insights, limitations, and future directions. Finally, Section 6 summarizes our contributions.

2. Background

2.1. Gaussian Splats

Recently, scene representation and novel-view synthesis techniques making use of machine learning methods have

*Both authors are students at Ludwig Maximilian University of Munich, Geschwister-Scholl-Platz 1, 80539 Munich, Germany. They contributed equally to this work in all aspects, including conceptualization, research, and writing. The order of authorship was determined randomly and does not reflect any difference in their contributions.

gained attention. One of the most popular frameworks in this category are Neural Radiance Fields (NeRFs) [], able to produce high quality implicit representations of scenes. Typically, NeRFs do this by optimizing a deep neural network using a volumetric continuous representation of the scene, using thechniques such as volumetric ray marching. In spite of various improvements to increase the efficiency of this framework [], achieving high visual quality through NeRFs remains computationally expensive due to the training cost of the neural network, in addition to a high rendering cost. To address these issues, the Gaussian Splatting (GS) framework was proposed.

Through Gaussian Splatting, instead of learning a continuous implicit representation of a scene, scenes are explicitly represented through a set of points using a large number of Gaussian primitives. The Gaussian ellipsoids constituting a scene have a set of learnable parameters controlling properties of the reconstruction, such as position, opacity, anisotropic covariance, and spherical harmonic (SH) coefficients. In addition to providing a high reconstruction quality, Gaussian Splatting provides much faster rendering, being able to produce novel-views in real time. GS achieves this by using rasterization-based rendering, which, in contrast to the rendering used by NeRFs, does not require sampling points.

Although GS was originally formulated for the reconstruction of 3D-scenes, and is often studied in that domain, this work considers 2D GS. Considering 2D GS alleviates some of the challenges involved in the process of auto-encoding Gaussian Splats.

2.2. Auto-Encoders

3. Methodology

3.1. Dataset Preparation

Our approach utilizes the CIFAR-10 dataset, a widely used benchmark for image processing tasks. The dataset consists of 60,000 color images of size 32×32 pixels, categorized into 10 classes. Each image is represented using a set of Gaussian splats, where each Gaussian encapsulates local image features.

To construct the Gaussian representation, we preprocess the dataset by mapping each image to a set of 23 Gaussians, each defined by a 1024-dimensional feature vector. These feature vectors capture spatial, color, and intensity distributions across the image. The dataset is partitioned into training, validation, and test sets using an 80-10-10 split.

3.2. Gaussian Splatting and Representation

Gaussian splatting is employed to encode images in a continuous and differentiable manner. Each image is decomposed into multiple Gaussians, where each Gaussian is parameterized by its position, scale, rotation, and opacity. This representation enables efficient image compression and reconstruction.

Each Gaussian G_i in an image is characterized by:

- **Mean Position** (x_i, y_i): Determines the spatial location of the Gaussian.
- **Covariance Matrix** Σ_i: Controls the spread and orientation.
- **Color Components** (r_i, g_i, b_i): Defines the color distribution.
- **Opacity** α_i: Regulates transparency in blending operations.

3.3. Configurations and Experimental Setups

We implemented various configurations to analyze the impact of different hyperparameters on Gaussian-based autoencoding. The key configurations include:

Baseline Configuration This setup uses a fixed number of 23 Gaussians per image, each represented using 1024-dimensional feature vectors. A simple MLP-based autoencoder is trained to reconstruct the Gaussian representation from a compressed latent space.

Variable Gaussian Count To study the effect of the number of Gaussians on reconstruction accuracy, we experimented with configurations ranging from 10 to 50 Gaussians per image. The autoencoder is adapted to dynamically encode and decode varying numbers of Gaussians.

Gaussian Parameter Augmentation We extended the feature vector to include additional properties such as:

- Higher-order moments for improved shape modeling.
- Adaptive opacity modulation to enhance reconstruction fidelity.
- Learned covariance scaling to optimize Gaussian spread.

Different Autoencoder Architectures We evaluated multiple autoencoder architectures, including:

- Standard MLP-based encoders and decoders.
- Transformer-based encoders for capturing long-range dependencies.
- Convolutional layers to improve spatial consistency in Gaussian embeddings.

Loss Function Variants We explored different loss functions to optimize reconstruction quality, including:

- Mean Squared Error (MSE) on Gaussian parameters.
- Structural Similarity Index (SSIM) for perceptual quality.
- KL-divergence regularization to enforce latent space structure.

The combination of these configurations provides insights into the optimal representation and reconstruction strategies for Gaussian-splatted images.

4. Results

5. Discussion

6. Conclusion

References