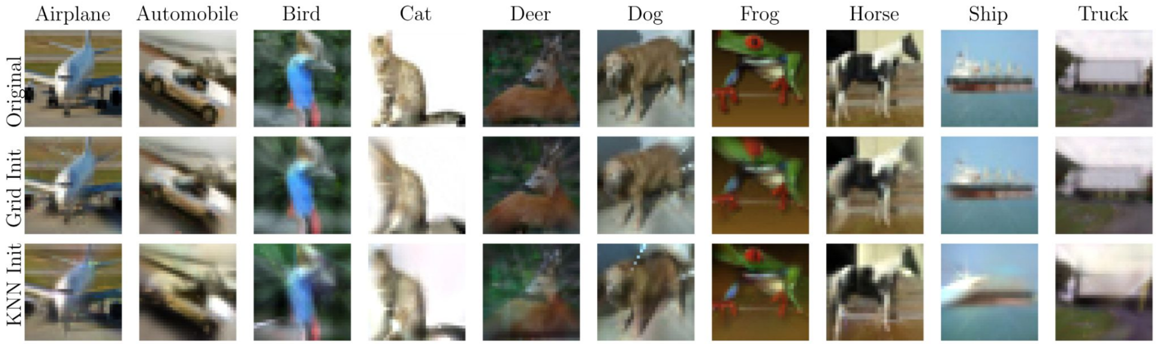


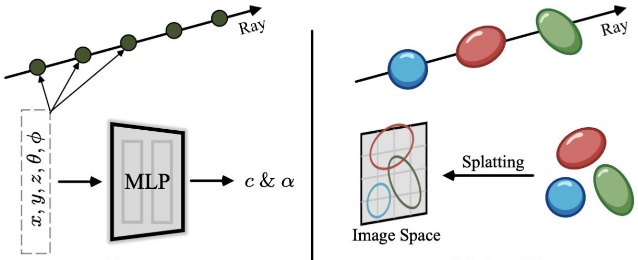
Autoencoders for 2D Gaussian Splats

Rok Mokotar, Federico Harjes Ruiloba

Background

Gaussian Splats

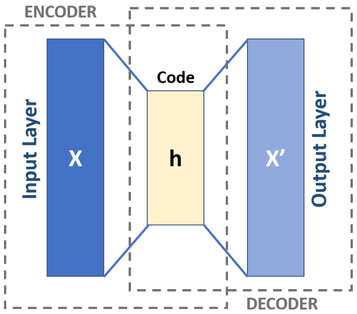
- ML-based scene representation and novel view synthesis -> **NeRF:** high-quality, continuous representation
- Slow optimization due to the cost of training a NN
- **3D Gaussian Splats:** much faster and similar quality, discrete representation



(a) NeRF (b) 3D GS

Autoencoders

- Encoder learns a compact representation of the input in the *latent space*
- Decoder then reconstructs the input from the latent space
- Useful for feature extraction and dimensionality reduction, among others



Methodology: Training a Splat Dataset

- **Dataset:** CIFAR-10
- **Splat generation:** gsplat
- **Hyperparameter Tuning:** Optuna
- **Added features:**
  - Selective learning of splat parameters
  - Support for different rasterization techniques
  - Bilateral guided radiance support
- **Initialization strategies:**
  - Random 3D initialization
  - Grid-based initialization
  - KNN-based initialization

Methodology: Autoencoding Gaussian Splats

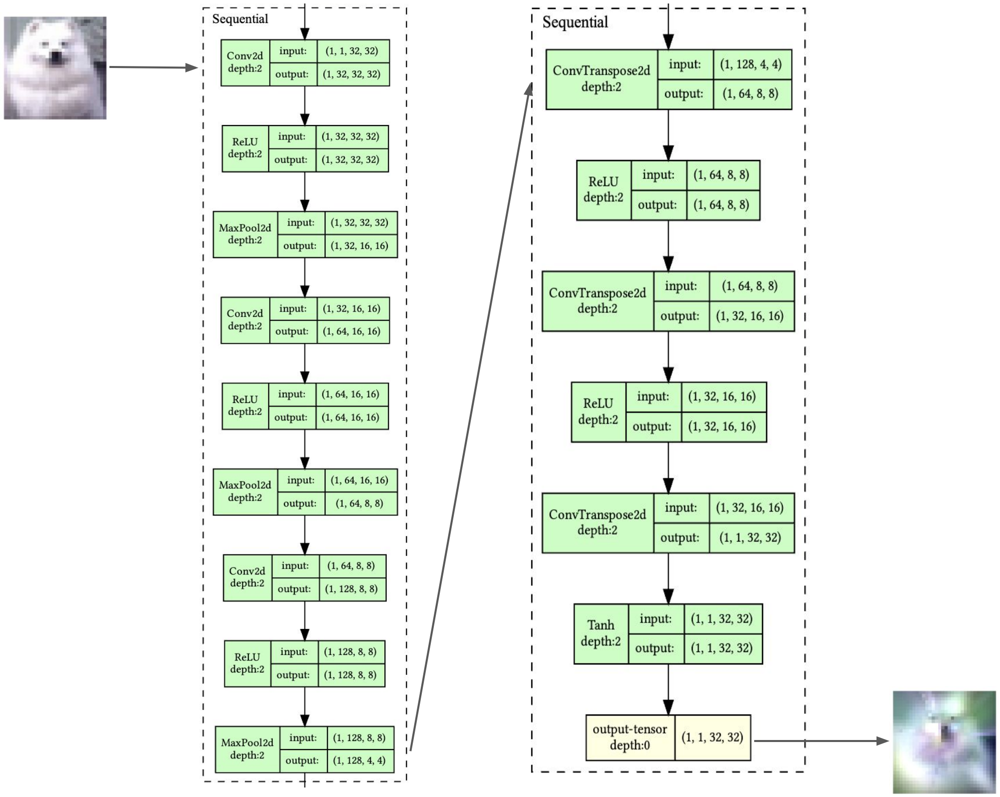
Model Architectures

- Simple (Deep) AE
- Convolutional AE
- ResNet-18 AE
- (Abstract AE)

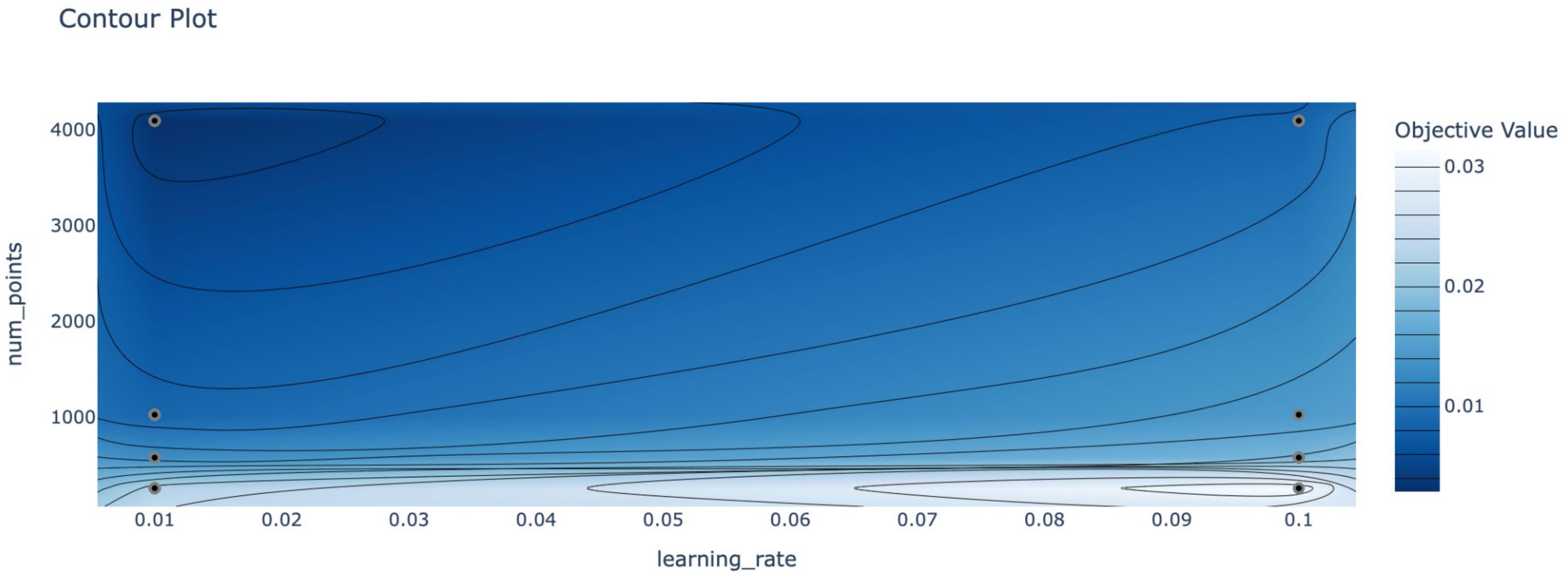
Splat Representations

- Vector-based encoding
- Full-image encoding
- Single-channel encoding
- Independent parameter models

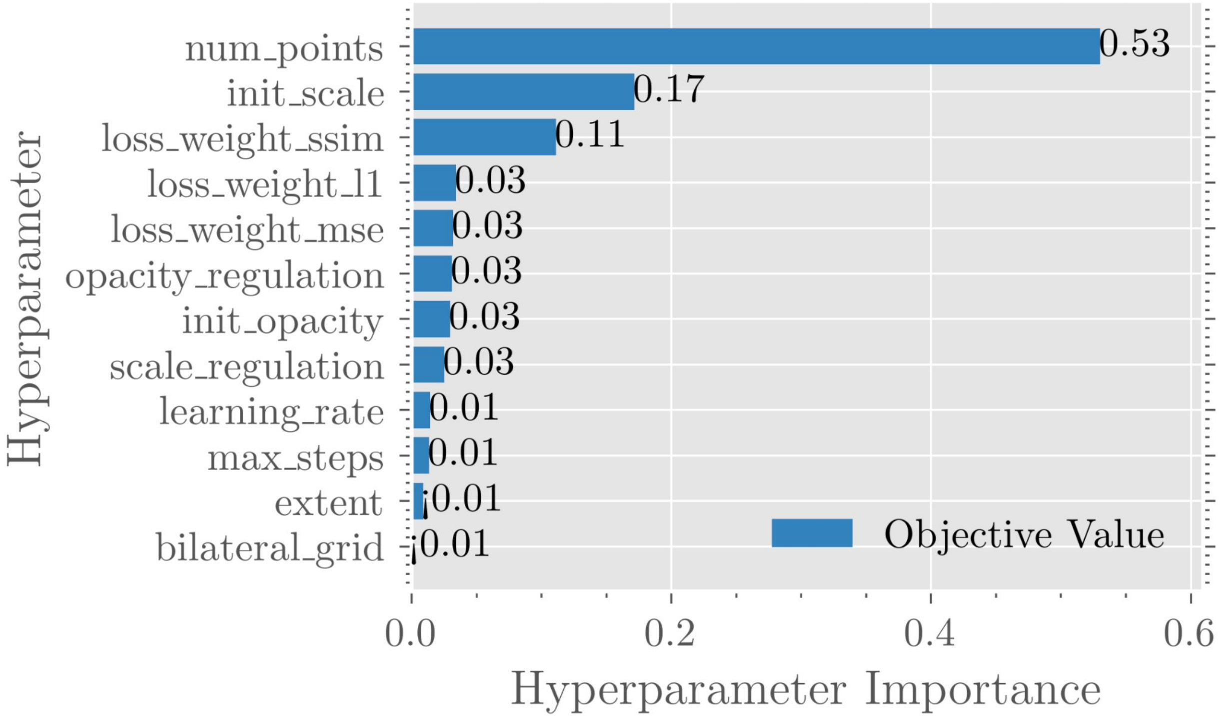
Hyperparameter Tuning (Optuna)
Slurm: Distributed Training over 10 machines with either 2060 RTX or CPU



Results: Gaussian Splats - Hyperparameter Tuning

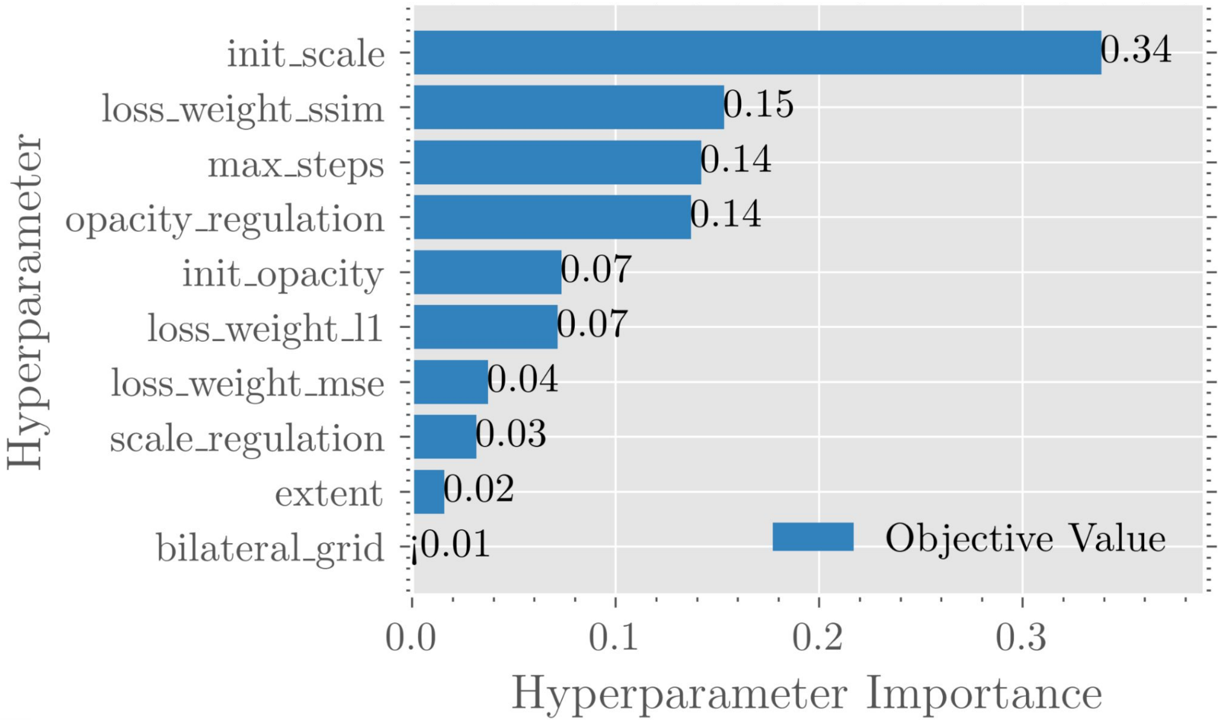


Results: Gaussian Splats - Hyperparameter Tuning

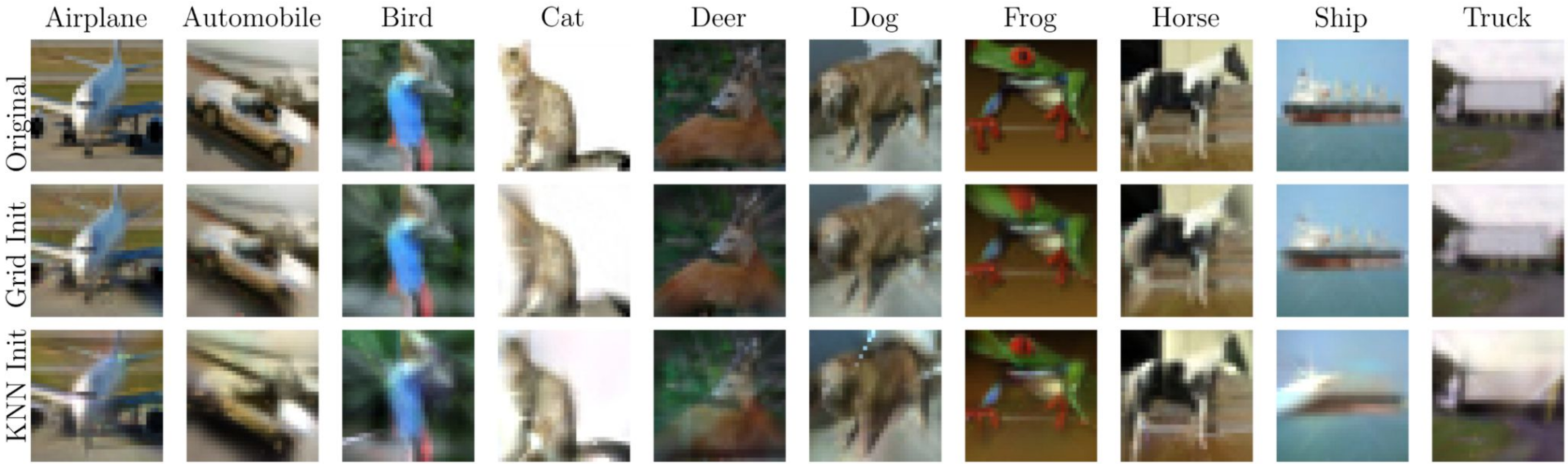


*KNN-init yielded slightly lower losses

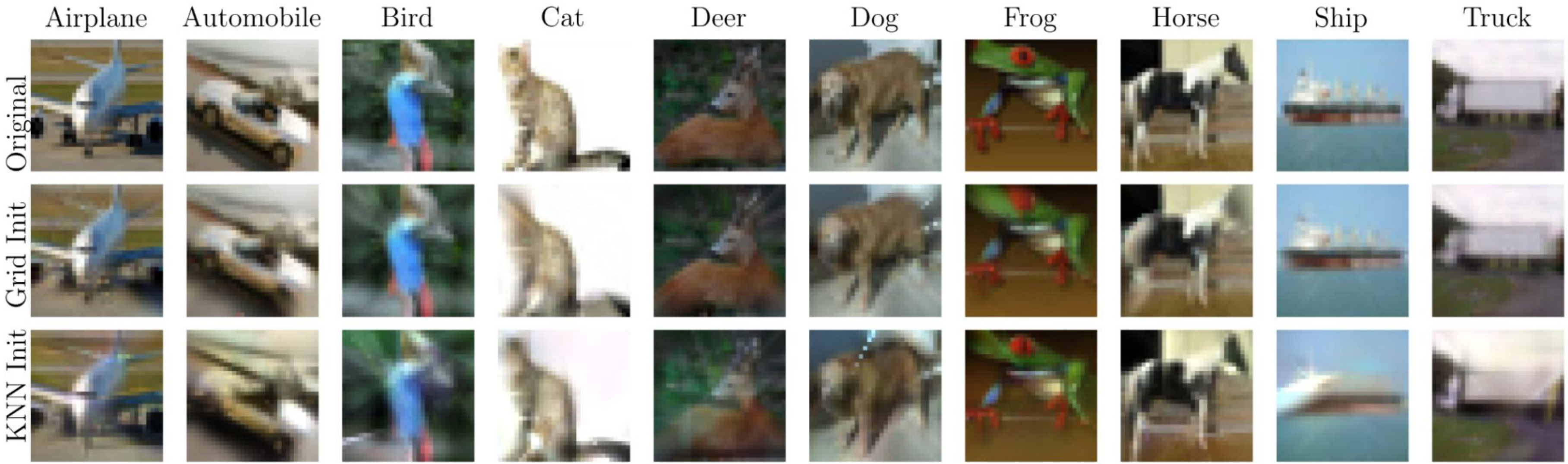
Results: Gaussian Splats - Hyperparameter Tuning



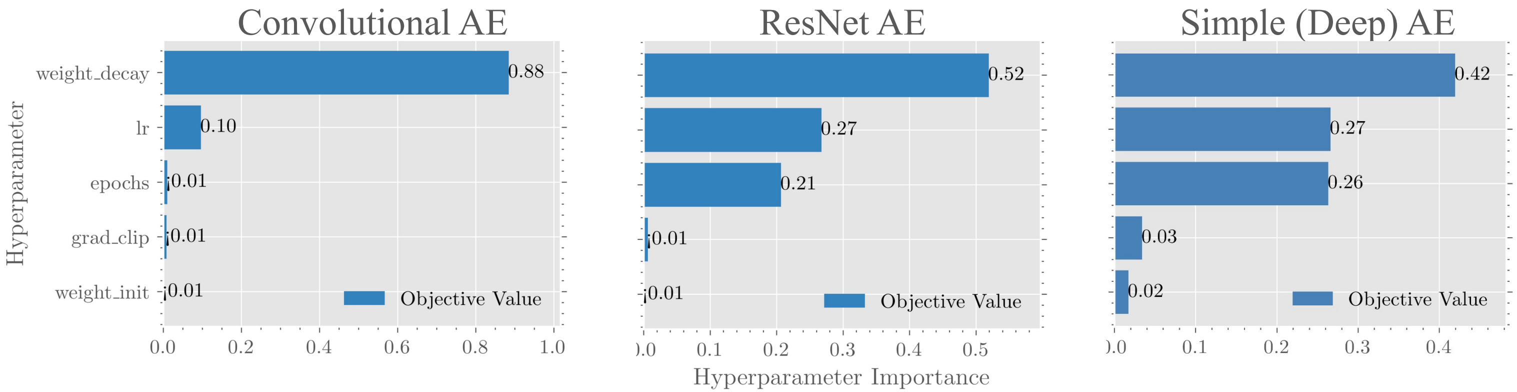
Results: Gaussian Splats - Full Dataset

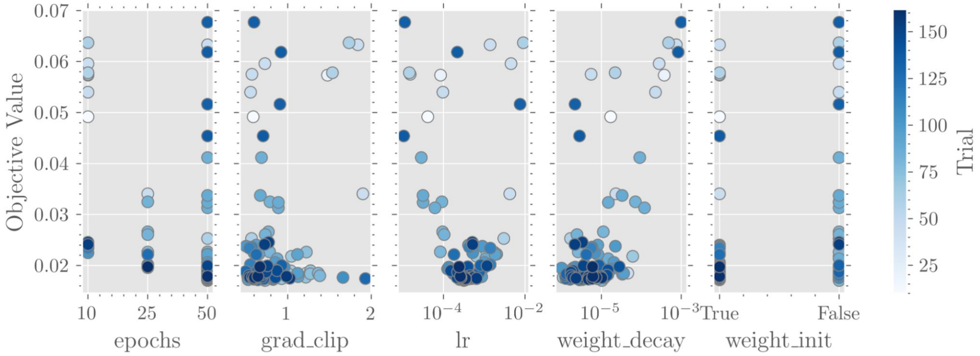


Results: Gaussian Splats - Full Dataset

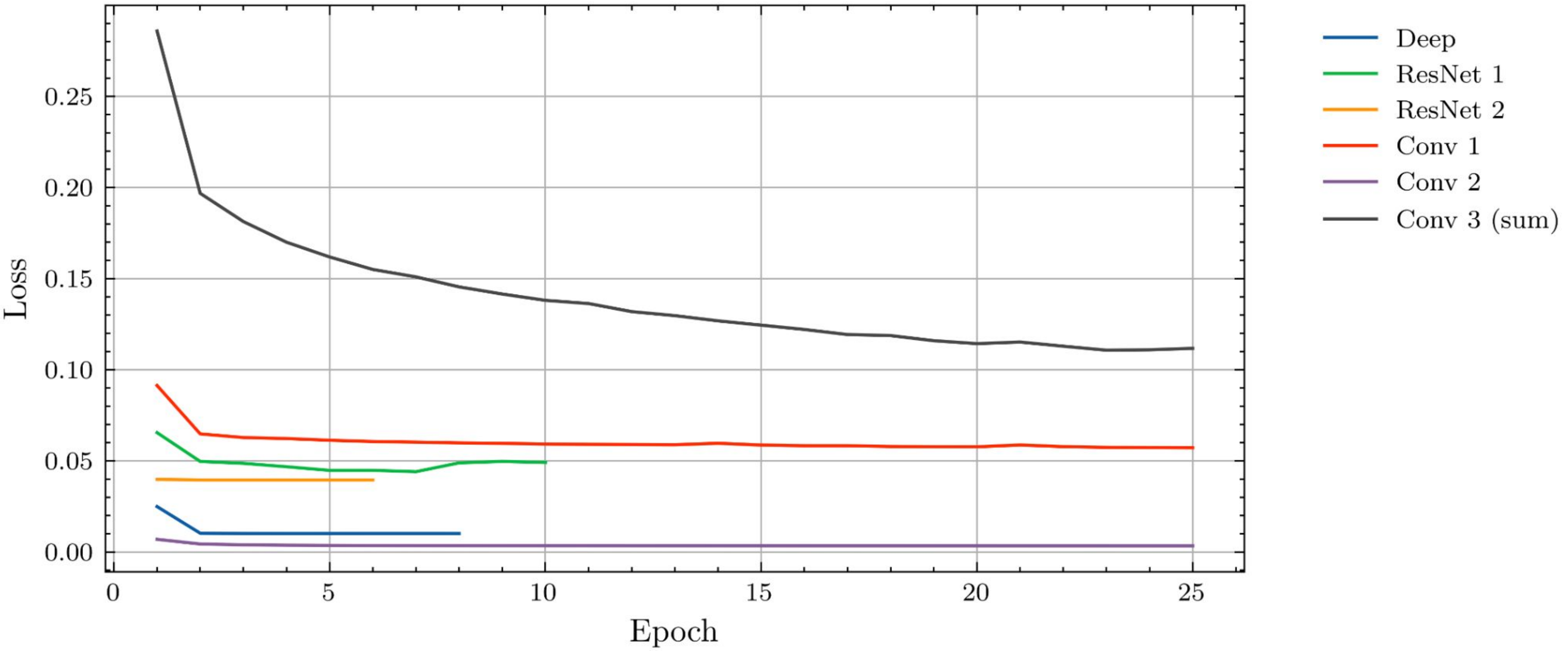


Results: Autoencoding GS - Hyperparameter Tuning

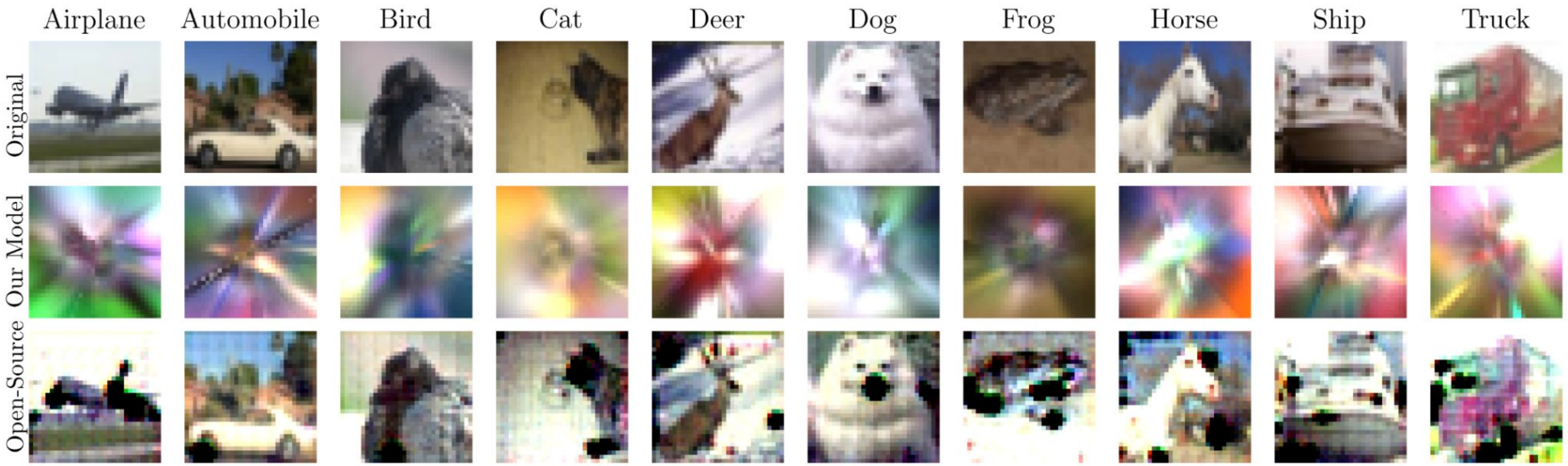




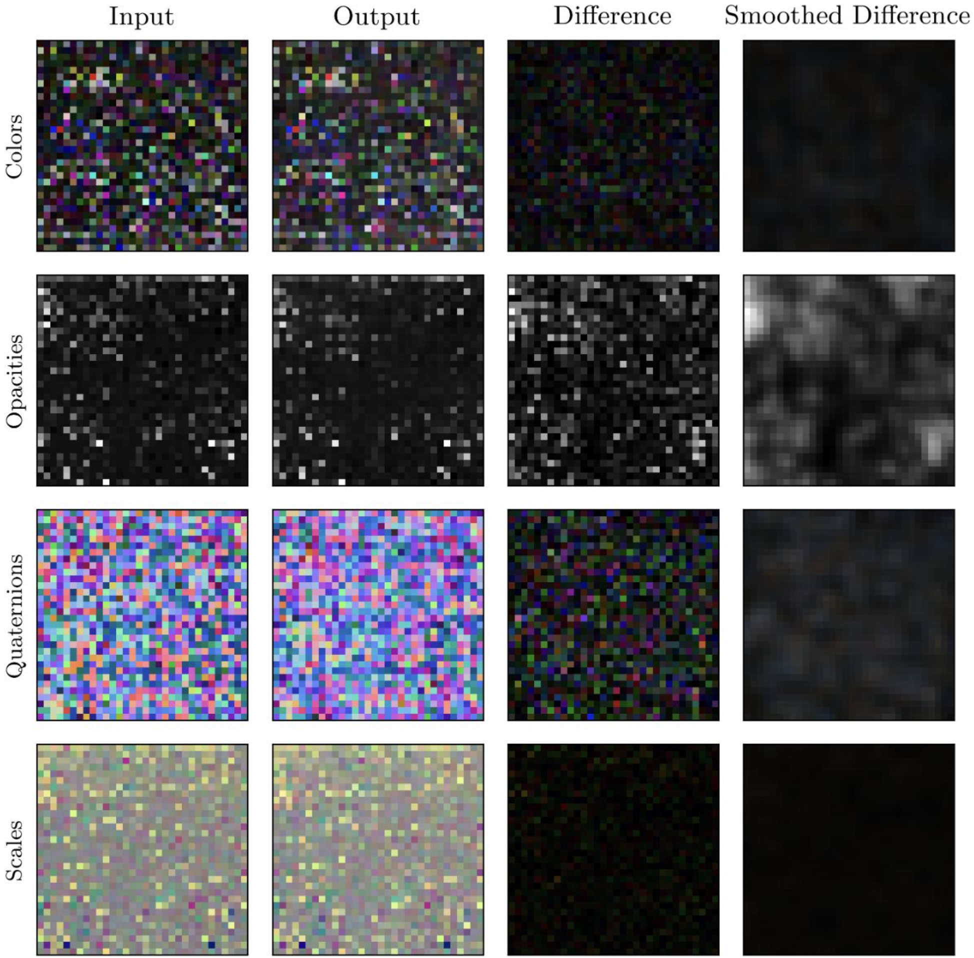
Results: Autoencoding GS - Loss Evolution



Results: Autoencoding GS - Reconstructed Images

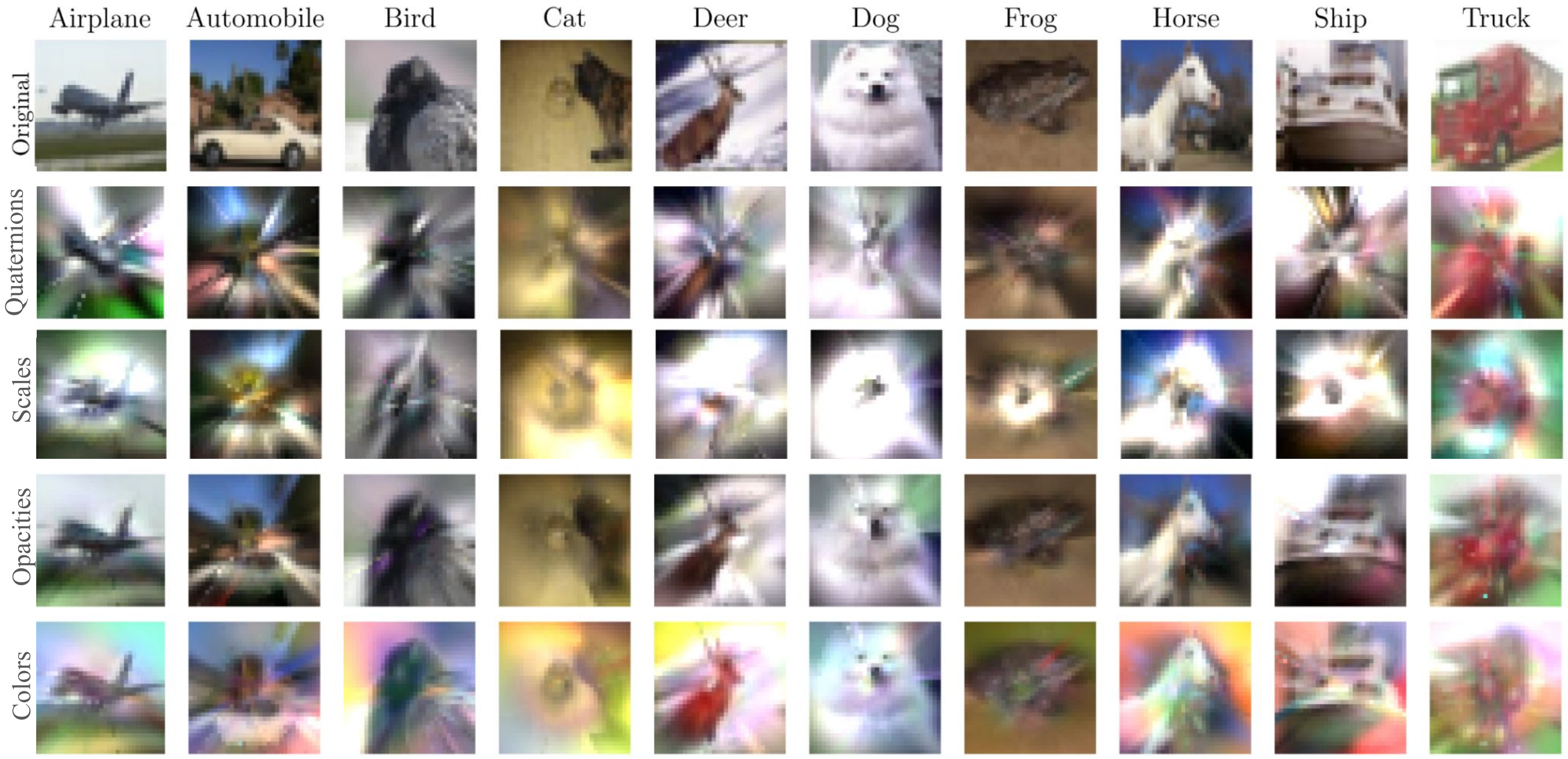


Individual Parameters



* **σ** = 1 for the smoothed difference
** quaternions were converted into a rotation matrix

Results: Autoencoding GS - Individual Parameters



Discussion: Interpreting the Results - Insights and Shortcomings

- **Gaussian Splatting:** highly accurate reconstruction of the CIFAR-10 dataset
- **Autoencoding:** small losses + similar visualizations for all parameters
  - ↯ deficient output quality across all AE variants
- **Possible causes:**
  - Small variations in splat representation might cause large variation in reconstructed image
  - Issues with reconstruction
  - Latent space might be too small (size = 16 for the best model)
- Unlikely that the model is incapable of

Discussion: Future Work

- Improved loss function
- Exploration of the latent space (size, classification)
- Implementation of more complex architectures (HiP)
- Latent space classification
- Generative modelling on the latent space (GAN, Stable Diffusion)
- Gaussian splat generation (VAE)

Methodology: Training Process

Model Architectures

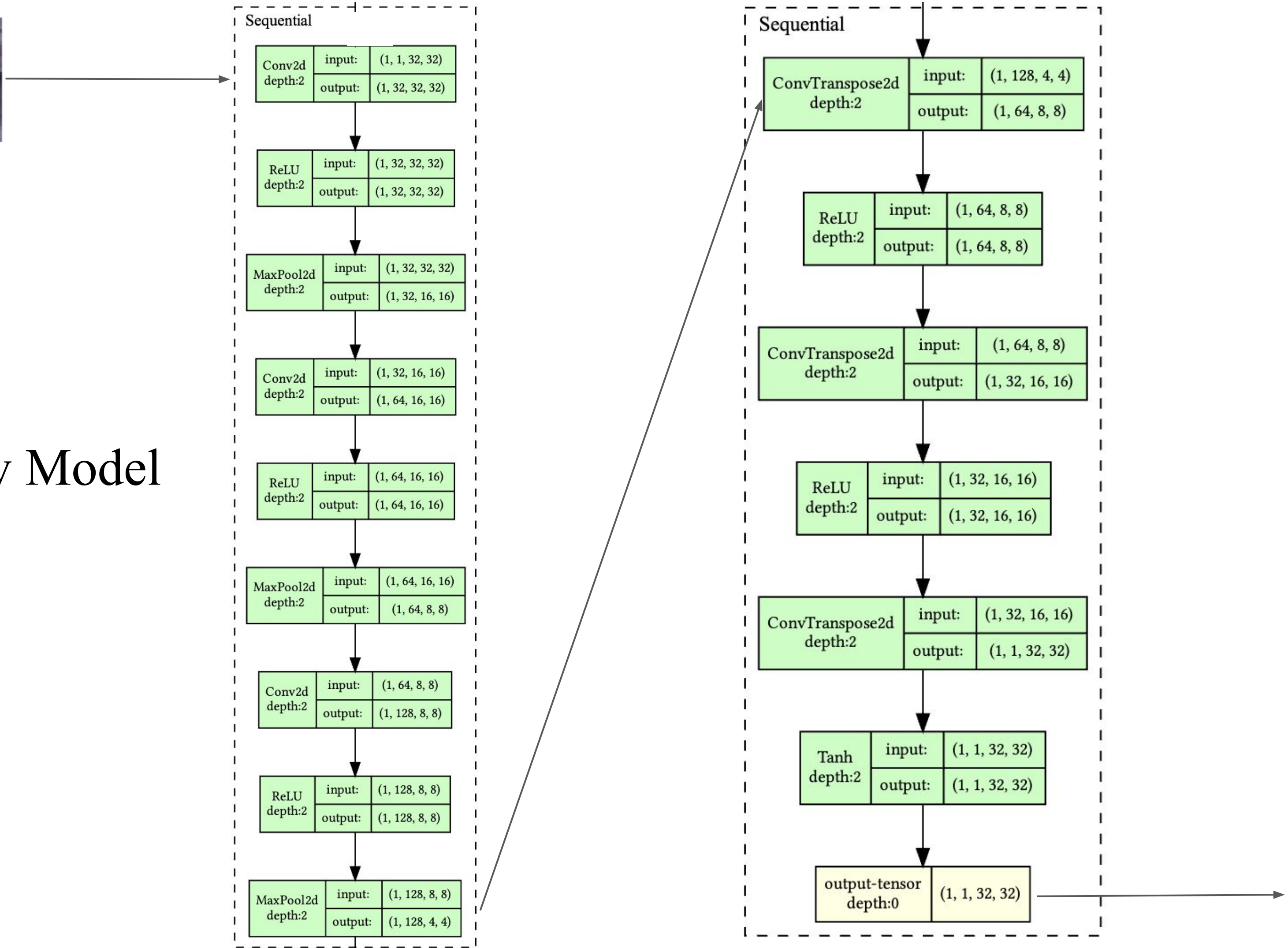
- Simple (Deep) AE
- Convolutional AE
- ResNet-18 AE
- (Abstract AE)

Splat Representations

- Vector-based encoding
- Full-image encoding
- Single-channel encoding
- Independent parameter models

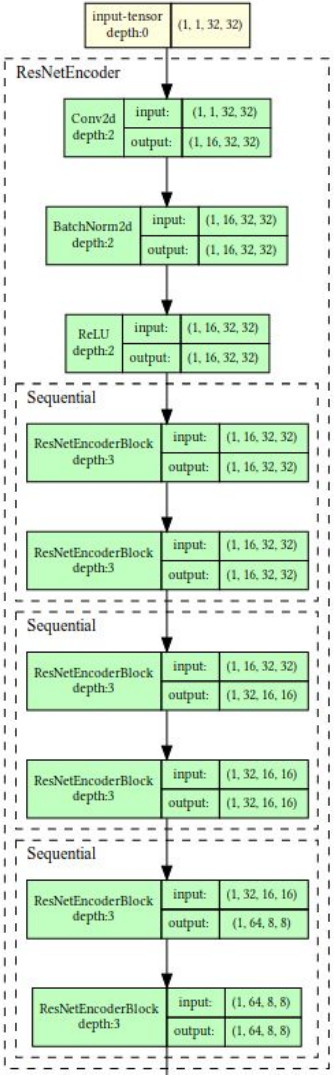
Hyperparameter Tuning (Optuna)

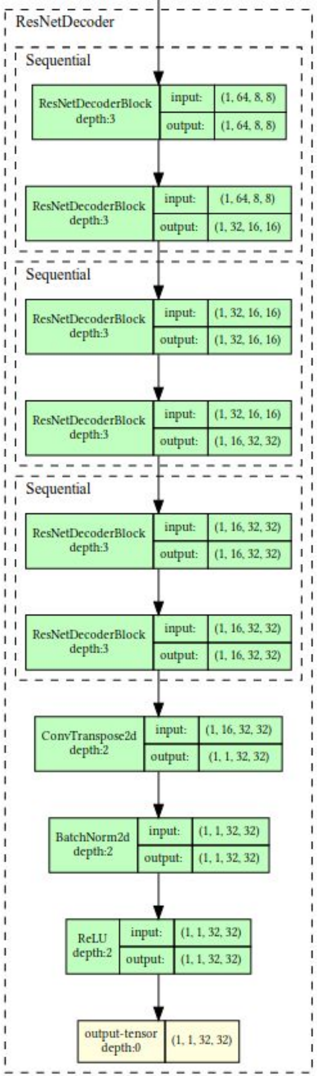
Conv Model



Sequential
Conv2d depth:2 | input: (1, 1, 32, 32) | output: (1, 32, 32, 32)
ReLU depth:2 | input: (1, 32, 32, 32) | output: (1, 32, 32, 32)
MaxPool2d depth:2 | input: (1, 32, 32, 32) | output: (1, 32, 16, 16)
Conv2d depth:2 | input: (1, 32, 16, 16) | output: (1, 64, 16, 16)
ReLU depth:2 | input: (1, 64, 16, 16) | output: (1, 64, 16, 16)
MaxPool2d depth:2 | input: (1, 64, 16, 16) | output: (1, 64, 8, 8)
Conv2d depth:2 | input: (1, 64, 8, 8) | output: (1, 128, 8, 8)
ReLU depth:2 | input: (1, 128, 8, 8) | output: (1, 128, 8, 8)
MaxPool2d depth:2 | input: (1, 128, 8, 8) | output: (1, 128, 4, 4)
Sequential
ConvTranspose2d depth:2 | input: (1, 128, 4, 4) | output: (1, 64, 8, 8)
ReLU depth:2 | input: (1, 64, 8, 8) | output: (1, 64, 8, 8)
ConvTranspose2d depth:2 | input: (1, 64, 8, 8) | output: (1, 32, 16, 16)
ReLU depth:2 | input: (1, 32, 16, 16) | output: (1, 32, 16, 16)
ConvTranspose2d depth:2 | input: (1, 32, 16, 16) | output: (1, 1, 32, 32)
Tanh depth:2 | input: (1, 1, 32, 32) | output: (1, 1, 32, 32)
output-tensor depth:0 | (1, 1, 32, 32)

ResNet Model





Results: Effect of the Number of Points on the Splat Quality

Results: ResNet-18 AE (Method 2)