Compression of 3D Gaussian Splatting with Optimized Feature Planes and Standard Video Codecs

Supplementary Material

Detailed Architecture. Each attribute is decoded using an MLP g, a three-layer, fully connected network. The intermediate layers contain 128 units each and use ReLU activation, except for the output layer. The four decoder MLPs have a total size of 0.16 MB, which is included in the final size results. We applied a sigmoid function to ensure the opacity values are within the range $[0, 1]$. For scaling, we first used the sigmoid function, then scaled the values to the range $[-10, -0.1]$, and followed with an exponential function. Finally, the predicted attributes are used as inputs to the original 3DGS rasterizer.

The feature plane learning rate is set to 0.005 using the Adam optimizer. Additionally, we apply learning rate schedulers to gradually decrease the learning rates during training. All other settings, including the learning rate for point positions, follow the original 3DGS.

Q_{step}	Feature Plane Size (MB)	PSNR
2^0	16.6	31.55
2^2	14.2	31.66
2^4	11.5	31.54
2^6	9.07	31.75
2^8	**8.45**	**31.72**
2^{10}	12.1	31.77
2^{12}	17.8	31.32

Table 1. Effect of Q_{step} on the proposed method for the 'bonsai' scene, with fixed λ_{ent} across all experiments.

Analysis for Quantization Step Size. To effectively optimize the standard video codec process, the quantization step size Q_{step} is crucial. During training, the uniform quantizer with a Q_{step} operates on floating-point transformed coefficients. However, it is important to note that the plane is scaled to 16-bit integers for compression. The difference between the pixel domain and the frequency domain makes it difficult to determine the proper quantization step size. Tab. 1 shows the impact of different quantization step sizes Q_{step} on our proposed method's performance. Throughout all experiments, we maintained a constant λ_{ent} to ensure consistent comparison. Since video codecs use predefined quantization matrix for compression, if the Q_{step} is too small or too large, entropy modeling may not work correctly. Our experiments demonstrate that a Q_{step} of 2^8 produces better results.

Configurations for Video Coding. The configuration parameters of traditional codecs influence the performance of the model. We provide the detailed settings used to obtain our experimental results, ensuring reproducibility of our findings.

The FFmpeg x265 (up to 12bit) command lines for lossless encoding of point positions used in our paper are:

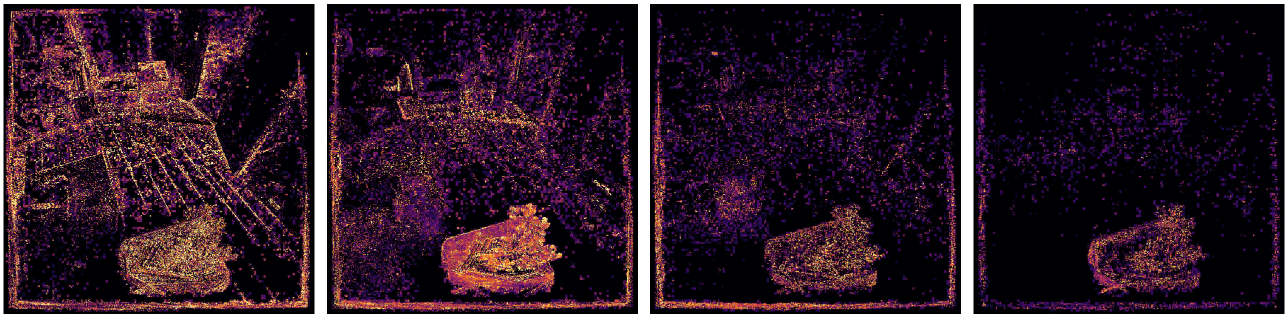
```
ffmpeg
-y
-pix_fmt gray16be
-s {width}x{height}
-framerate {framerate}
-i {input file name}
-c:v libx265
-x265-params
lossless=1
{bitstream file name}
```

The command lines for HM 16.0 (RExt) compression are:

```
TAppEncoder
-c encoder_randomaccess_main_rext.cfg
--InputFile={input file name}
--SourceWidth={width}
--SourceHeight={height}
--InputBitDepth=16
--InternalBitDepth=16
--OutputBitDepth=16
--InputChromaFormat=400
--FrameRate={framerate}
--FramesToBeEncoded=32
--QP={qp}
--BitstreamFile={bitstream file name}
```

As mentioned in the manuscript, the HM configuration used in the experiments has all QP offsets between frames set to 0. All other settings followed the default configuration. The modified configuration is as follows:

```
encoder_randomaccess_main_rext.cfg
#         Type POC QPoffset (...)
Frame1:  B   16    0
Frame2:  B    8    0
Frame3:  B    4    0
Frame4:  B    2    0
Frame5:  B    1    0
Frame6:  B    3    0
Frame7:  B    6    0
Frame8:  B    5    0
(...)
```



(a) $\mathcal{P}_1^{XZ}$ (b) $\mathcal{P}_3^{XZ}$ (c) $\mathcal{P}_5^{XZ}$ (d) $\mathcal{P}_7^{XZ}$

Figure 1. **Visualization of channel levels in XZ plane for the *'Bonsai'* scene.** With dynamic w_c, the lower-level channels preserve more information, whereas the higher-level channels are largely minimized due to the higher λ_{ent} weight assignment.

Analysis of Bit Allocation. Fig. 1 illustrates the results of each channel learned through our proposed bit allocation method. The signals in higher-level channels exhibit increased sparsity, making them more challenging for video coding. However, since these channels have lower impact on visual quality, most regions are minimized through allocating higher entropy λ_{ent} for each level. This strategic allocation results in improved rate-distortion performance compared to conventional approaches.

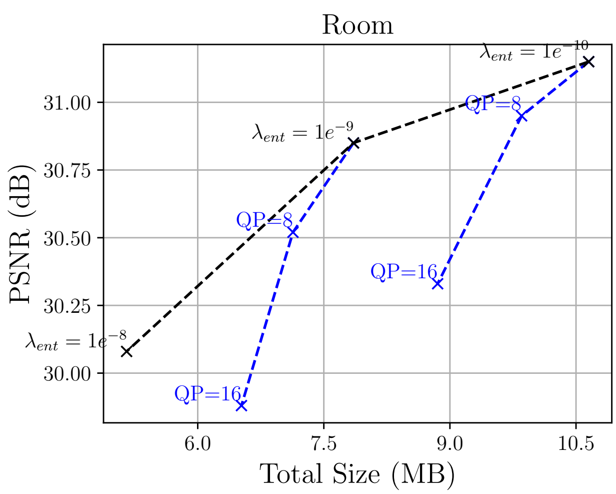
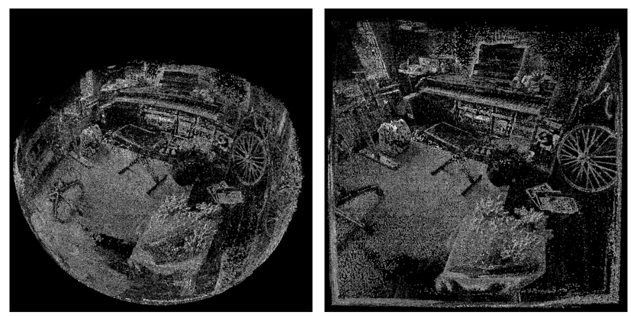


Figure 2. **RD curves with video QP adjustment.** Controlling rate-distortion with video QP is worse than using λ_{ent} with QP=1.

Performance analysis of QP adjustment. Beyond adjusting the parameter λ_{ent}, the rate-distortion trade-off can also be controlled by modifying the quantization parameter (QP) in video codecs such as HM or FFmpeg. However, our experimental results demonstrate that modifying the video codec QP yields inferior performance compared to adjusting λ_{ent}, as shown in Fig. 2. Our results suggest this may be because video codecs focus exclusively on feature plane restoration rather than the rendering view quality.



(a) sphere contraction (b) piecewise contraction

Figure 3. **Visualization of each contraction method for the *'Bonsai'* scene.** (a) sphere contraction and (b) piecewise-projective contraction.

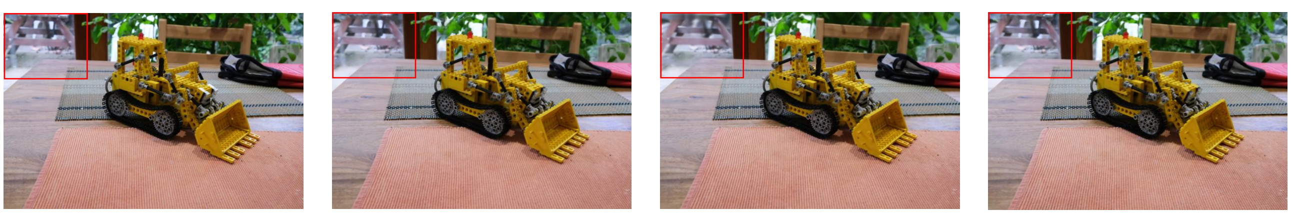
Contraction methods. Fig 4 presents the feature plane results for each contraction strategy. Although there is no significant difference in reconstruction quality between the two methods, piecewise contraction is expected to perform better in block-wise DCT due to the spatial correlation of the features. Experimental results demonstrate a slight improvement in size reduction when using piecewise contraction.

Per-scene Quantitative Results. We evaluated the performance on various datasets for novel view synthesis. Our analysis includes per-scene results for the Mip-NeRF 360, Deep Blending, and Tank&Temples datasets.

Scene	bicycle	flowers	garden	stump	tree hill	room	counter	kitchen	bonsai	Avg.
PSNR	25.19	21.37	27.61	26.59	23.07	30.99	28.47	30.73	31.71	27.30
SSIM	0.746	0.593	0.853	0.781	0.648	0.922	0.898	0.916	0.934	0.810
LPIPS	0.265	0.380	0.133	0.236	0.346	0.217	0.208	0.137	0.202	0.236
Storage (MB)	9.82	10.25	15.22	14.30	9.92	7.91	6.25	7.68	6.71	9.78

Table 2. Per-scene results evaluated on Mip-NeRF 360 dataset.

GT	Compact3DGS+PP	CompGS	Ours
	43.2MB	30.7MB	**15.2MB**
	26.72dB/0.831/0.158	27.47dB/0.855/0.131	27.59dB/0.853/0.132



GT	Compact3DGS+PP	CompGS	Ours
	24.6MB	10.4MB	**7.7MB**
	30.48dB/0.919/0.130	30.67dB/0.912/0.146	30.73dB/0.916/0.137

Figure 4. **Qualitative results for visual comparison for Mip-NeRF360 dataset.** Each subfigure displays the storage size along with the PSNR, SSIM, and LPIPS metrics. Detailed observation is encouraged by zooming in.

Dataset	Tanks&Temples			Deep Blending		
Scene	train	truck	Avg.	drjohnson	playroom	Avg.
PSNR	22.08	25.19	23.63	29.22	30.42	29.82
SSIM	0.801	0.882	0.842	0.904	0.909	0.907
LPIPS	0.226	0.158	0.192	0.250	0.252	0.251
Storage (MB)	6.21	8.72	7.46	9.39	7.86	8.62

Table 3. Per-scene results evaluated on Tank&Temples and Deep Blending.

GT

Compact3DGS+PP
19.4MB
30.30dB/0.900/0.259

CompGS
8.9MB
30.12dB/0.901/0.267

Ours
7.9MB
30.42dB/0.909/0.252

GT

Compact3DGS+PP
22.6MB
25.02dB/0.870/0.163

CompGS
10.1MB
25.14dB/0.866/0.176

Ours
8.8MB
25.09dB/0.878/0.158

Figure 5. **Qualitative results for visual comparison for DeepBlending and T&T dataset.** Each subfigure displays the storage size along with the PSNR, SSIM, and LPIPS metrics. Detailed observation is encouraged by zooming in.