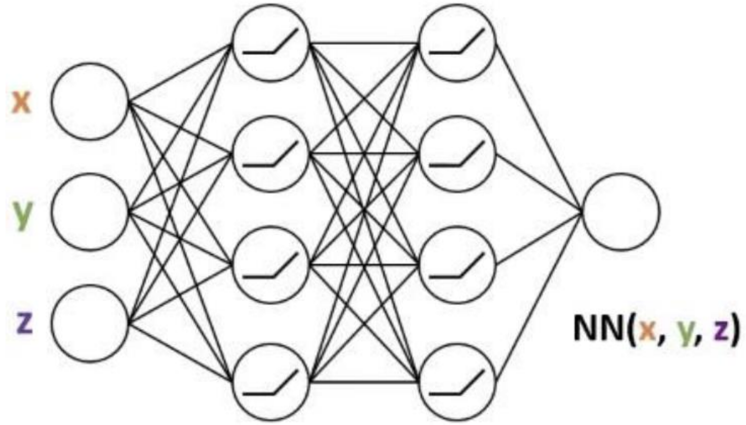
GaussianImage: 1000 FPS Image Representation and Compression by 2D Gaussian Splatting

Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang,
Hongwei Qin, Guo Lu, Jing Geng, Jun Zhang

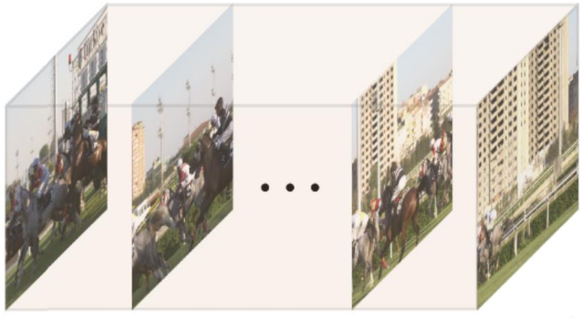
香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Background: Implicit Neural Representations

- Parametrize a signal as a continuous function
  - Input: coordinate
  - Function: neural network
  - Output: RGB values, density
- Advantages:
  - Arbitrary Resolution → signal super-resolution
  - Memory efficient → signal compression
  - Capture, retain and infer signal details → signal inpainting, deblurring, denoising, ….



image



video

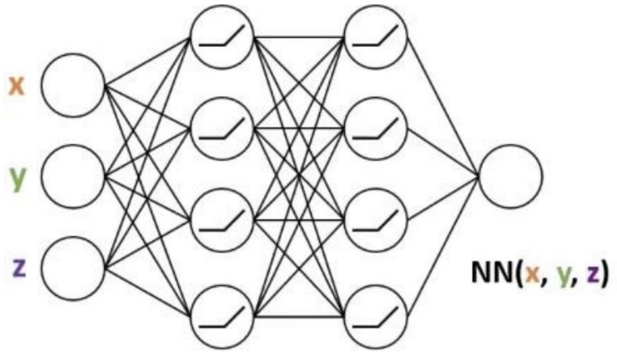
3D object

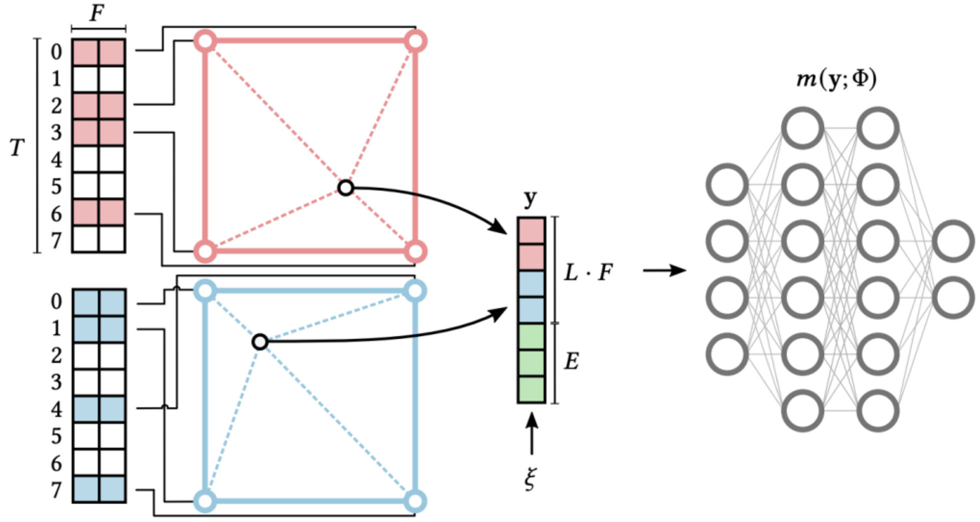
[1] Vincent Sitzmann, et al. Implicit Neural Representations with Periodic Activation Functions, NeurIPS 2020.
[2] Hao Chen, et al, NeRV: Neural Representations for Videos, NeurIPS 2021.

Background: Implicit Neural Representations

- Two types in image INRs:
  - MLP-based INR:
    - □ Long training times
    - □ Slow decoding speed
    - □ High GPU memory consumption
  - Feature grid-based INR:
    - Fast training and inference
    - Higher GPU memory consumption

Low-end devices with limited memory: unfriendly!

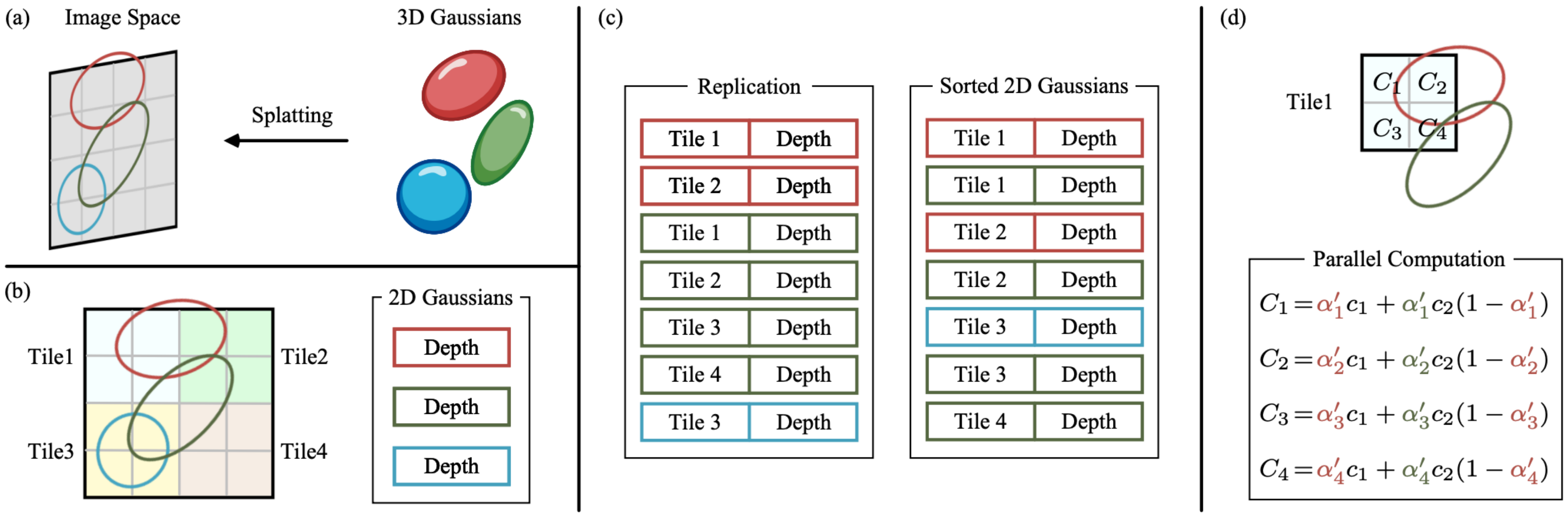




[3] Thomas Müller et al. “Instant Neural Graphics Primitives with a Multiresolution Hash Encoding”, SIGGRAPH 2022.

Motivation: Gaussian Splatting

- The characteristics of advanced neural image representation:
  - Efficient training
  - Fast decoding
  - Friendly GPU memory usage
- Gaussian Splatting in 3D scene reconstruction:
  - Explicit 3D Gaussian representations and differentiable tile-based rasterization,
  - High visual quality with competitive training times,
  - Real-time rendering capabilities



Replication

Tile 1	Depth
Tile 2	Depth
Tile 1	Depth
Tile 2	Depth
Tile 3	Depth
Tile 4	Depth
Tile 3	Depth

Sorted 2D Gaussians

Tile 1	Depth
Tile 1	Depth
Tile 2	Depth
Tile 2	Depth
Tile 3	Depth
Tile 3	Depth
Tile 4	Depth

Parallel Computation

$$C_1 = \alpha'_1 c_1 + \alpha'_1 c_2 (1 - \alpha'_1)$$

$$C_2 = \alpha'_2 c_1 + \alpha'_2 c_2 (1 - \alpha'_2)$$

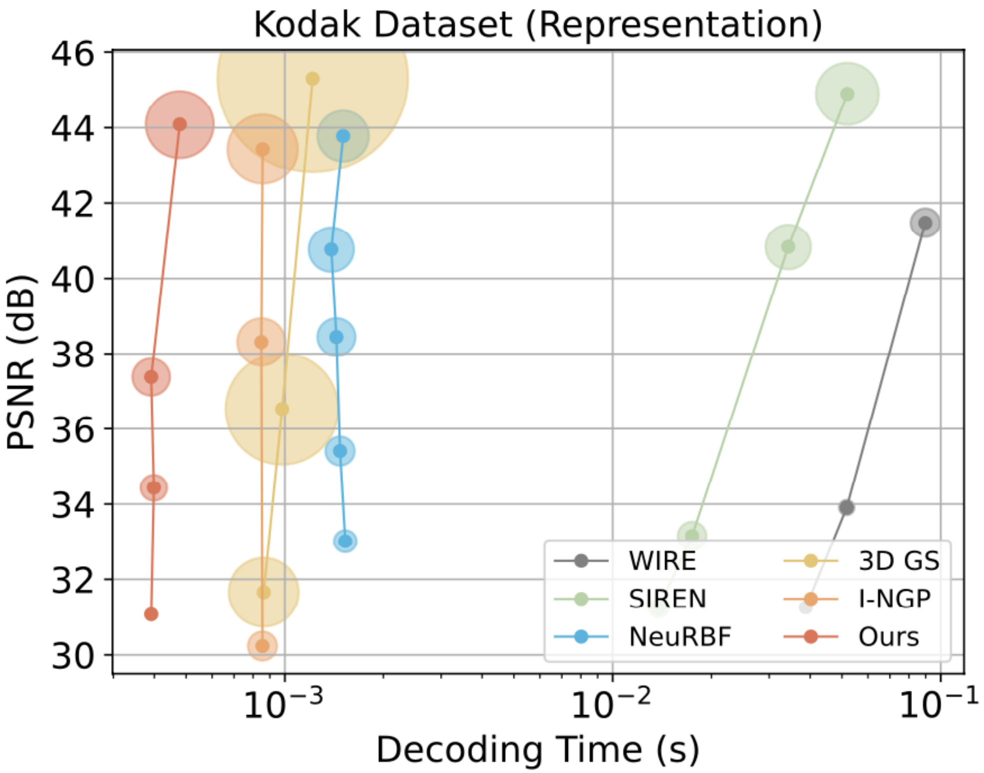
$$C_3 = \alpha'_3 c_1 + \alpha'_3 c_2 (1 - \alpha'_3)$$

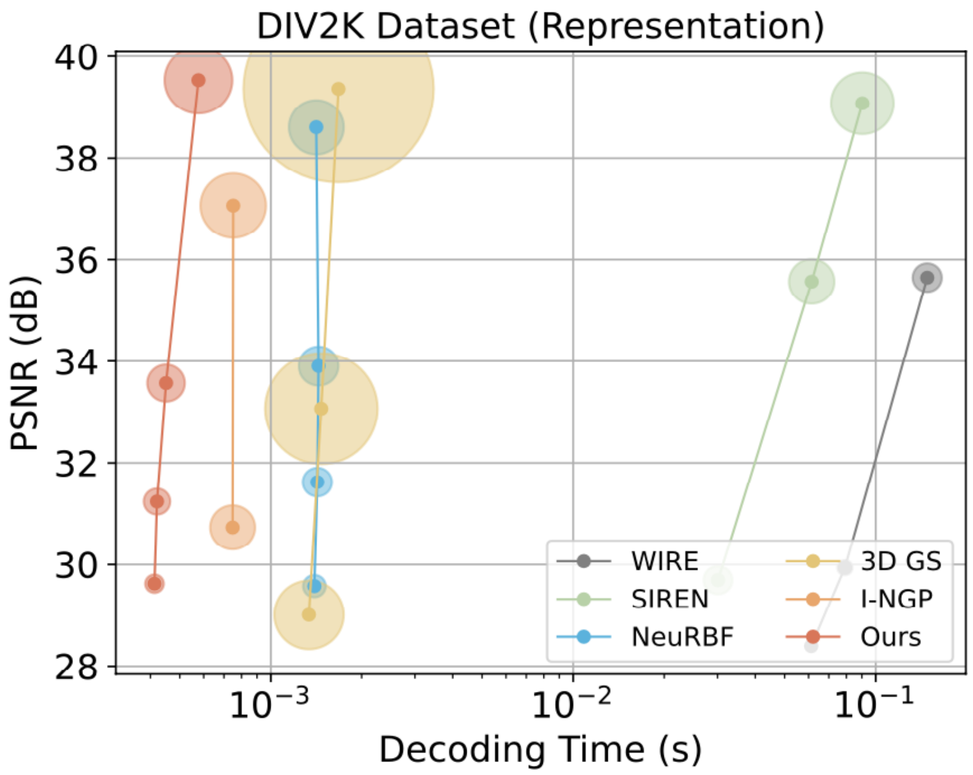
$$C_4 = \alpha'_4 c_1 + \alpha'_4 c_2 (1 - \alpha'_4)$$

[4] Bernhard Kerbl et al. "3D Gaussian Splatting for Real-Time Radiance Field Rendering", ACM Transactions on Graphics 2023.

Challenges

- Non-trivial to directly adapt 3D GS for efficient single image representation
  - ➢ 3D Gaussian Representation:
    - □ Each 3D Gaussian has 59 parameters
    - □ Thousands of 3D Gaussians are required for representing a single image
    - □ Increases the storage and communication demands


Kodak Dataset (Representation)


DIV2K Dataset (Representation)

Challenges

- Non-trivial to directly adapt 3D GS for efficient single image representation
  - Alpha Blending-based Rasterization:
    - Requires pre-sorted Gaussians based on depth information
      - Single natural images: detailed camera parameters are often not known
      - Non-natural images: they are not captured by cameras
      - w/o depth information → Gaussian sorting is impaired
    - Skips remaining Gaussians once the accumulated opacity surpasses the threshold
      - Underutilization of Gaussians
      - Require more Gaussians for high-quality rendering

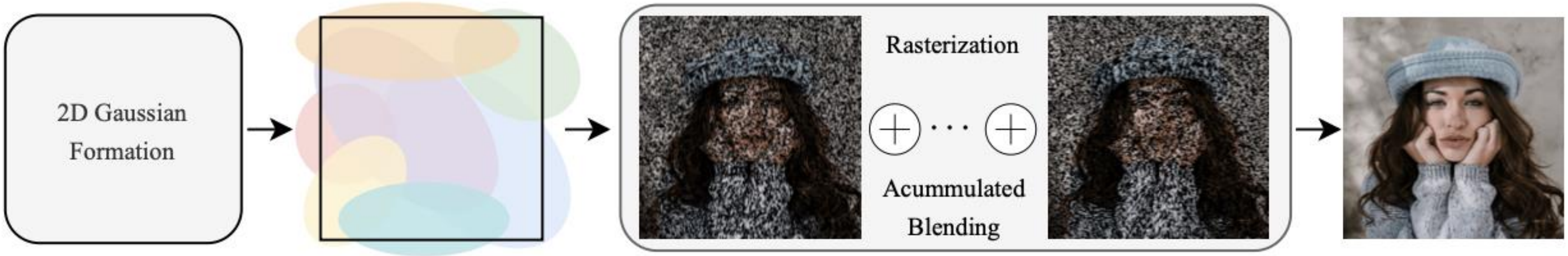
$$\boldsymbol{C}_i = \sum_{n \in \mathcal{N}} \boldsymbol{c}_n \alpha_n T_n \, , \qquad T_n = \prod_{m=1}^{n-1} (1 - \alpha_m) \, , \qquad \alpha_n = o_n \exp(-\sigma_n) \, , \qquad \sigma_n = \frac{1}{2} \boldsymbol{d}_n^T \boldsymbol{\Sigma}^{-1} \boldsymbol{d}_n$$

GaussianImage: 2D Gaussian Formation

- GaussianImage: groundbreaking image representation paradigm
  - ➢ 2D Gaussian Formation:
    - □ Each 2D Gaussian has 4 attributes (9 parameters in total):
      - ◆ Position: $\mu \in \mathbb{R}^2$
      - ◆ Anisotropic covariance: $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ or $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{R}^T$
      - ◆ Color coefficients : $c \in \mathbb{R}^3$
      - ◆ Opacity : $\mathrm{o} \in \mathbb{R}$
    - □ A 6.5× compression over 3D Gaussians

$$\boldsymbol{L} = \begin{bmatrix} l_1 & 0 \\ l_2 & l_3 \end{bmatrix}$$

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$$



GaussianImage: Rasterization

- GaussianImage: groundbreaking image representation paradigm
  - ➢ Accumulated Blending-based Rasterization:
    - □ No viewpoint influence → Deterministic order → Merge T_n into o_n
    - □ Benefits:
      - ◆ Fully utilize the information of all Gaussian points covering the current pixel
      - ◆ Avoid the tedious calculation of accumulated transparency to accelerate training and inference
      - ◆ Allow us to combine color coefficients and opacity into a singular set of weighted color coefficients → 8 parameters and a 7.375× compression over 3D Gaussians

$$\boldsymbol{C}_i = \sum_{n \in \mathcal{N}} \boldsymbol{c}_n \alpha_n T_n , \qquad T_n = \prod_{m=1}^{n-1} (1 - \alpha_m) , \qquad \alpha_n = o_n \exp(-\sigma_n) , \qquad \sigma_n = \frac{1}{2} \boldsymbol{d}_n^T \boldsymbol{\Sigma}^{-1} \boldsymbol{d}_n$$

⬇

$$\boldsymbol{C}_i = \sum_{n \in \mathcal{N}} \boldsymbol{c}_n \alpha_n = \sum_{n \in \mathcal{N}} \boldsymbol{c}_n o_n \exp(-\sigma_n) \dashrightarrow \boldsymbol{C}_i = \sum_{n \in \mathcal{N}} \boldsymbol{c}'_n \exp(-\sigma_n)$$

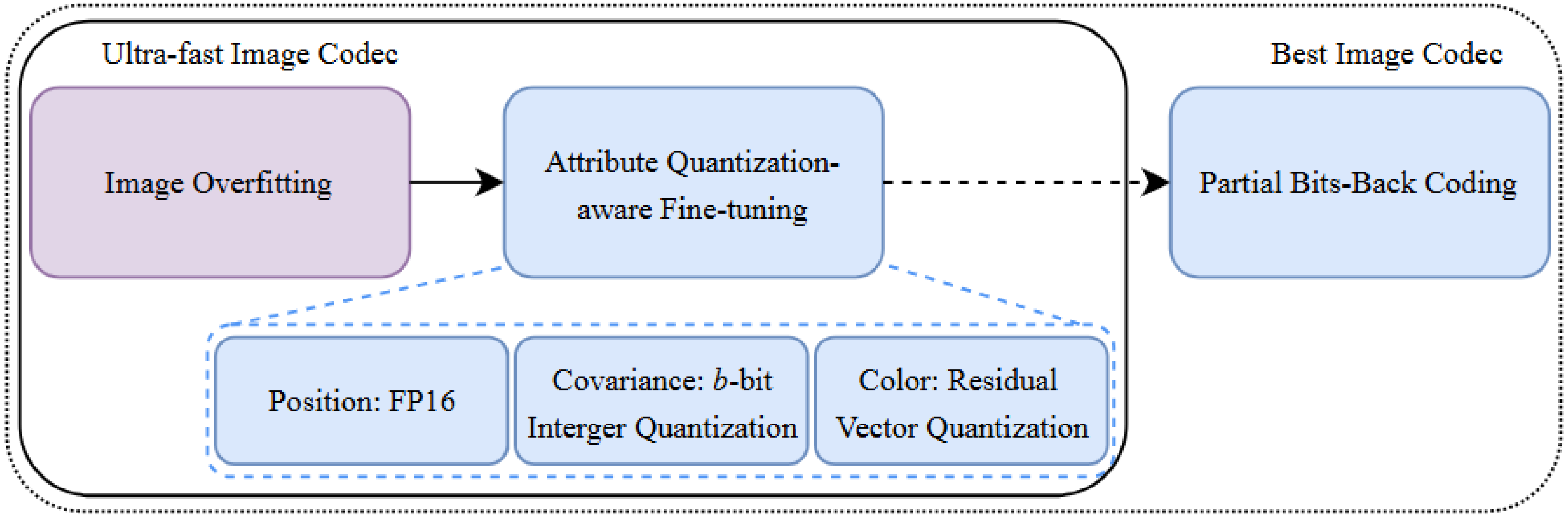
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Application: Image Compression

- Ultra-fast Image Codec: Attribute Quantization-aware Fine-tuning
  - ➢ Position: FP16
  - ➢ Covariance: 6-bit quantization

$$\hat{l}_i^n = \left\lfloor clamp(\frac{l_i^n - \beta_i}{\gamma_i}, 0, 2^b - 1) \right\rfloor, \bar{l}_i^n = \hat{l}_i^n \times \gamma_i + \beta_i$$

  - ➢ Color: residual vector quantization
    - □ Codebook size *B*: 8
    - □ Number of quantization stages *M*: 2


Ultra-fast Image Codec
Best Image Codec
Image Overfitting
Attribute Quantization-aware Fine-tuning
Partial Bits-Back Coding
Position: FP16
Covariance: b-bit Interger Quantization
Color: Residual Vector Quantization

Application: Image Compression

- Best GaussianImage-based Codec: Partial Bits-back Coding [5]
  - ➢ Encoding ordered data brings additional storage overhead
  - ➢ An unordered set with N elements has $N!$ equivariant
  - ➢ Bits-back coding can save a bitrate of $logN! - logN$
  - ➢ Practical operation:
    - □ Encode the initial K Gaussians by vanilla entropy coding
    - □ Encode the subsequent $N - K$ Gaussians by bits-back coding
    - □ Find the optimal K: Let R_k denotes the bitrate of k-th Gaussian, the final bitrate saving can be formalized as:

$$\log(N - K^*)\,! - \log(N - K^*)\,,$$

$$where\ K^* = \inf K\,, s.t. \sum_{k=1}^{K} R_k - \log(N - K^*)\,! \geq 0\,.$$

[5] Julius Kunze et al. “Entropy coding of unordered data structures”, ICLR 2024.

Comprehensive Evaluation: Image Representation

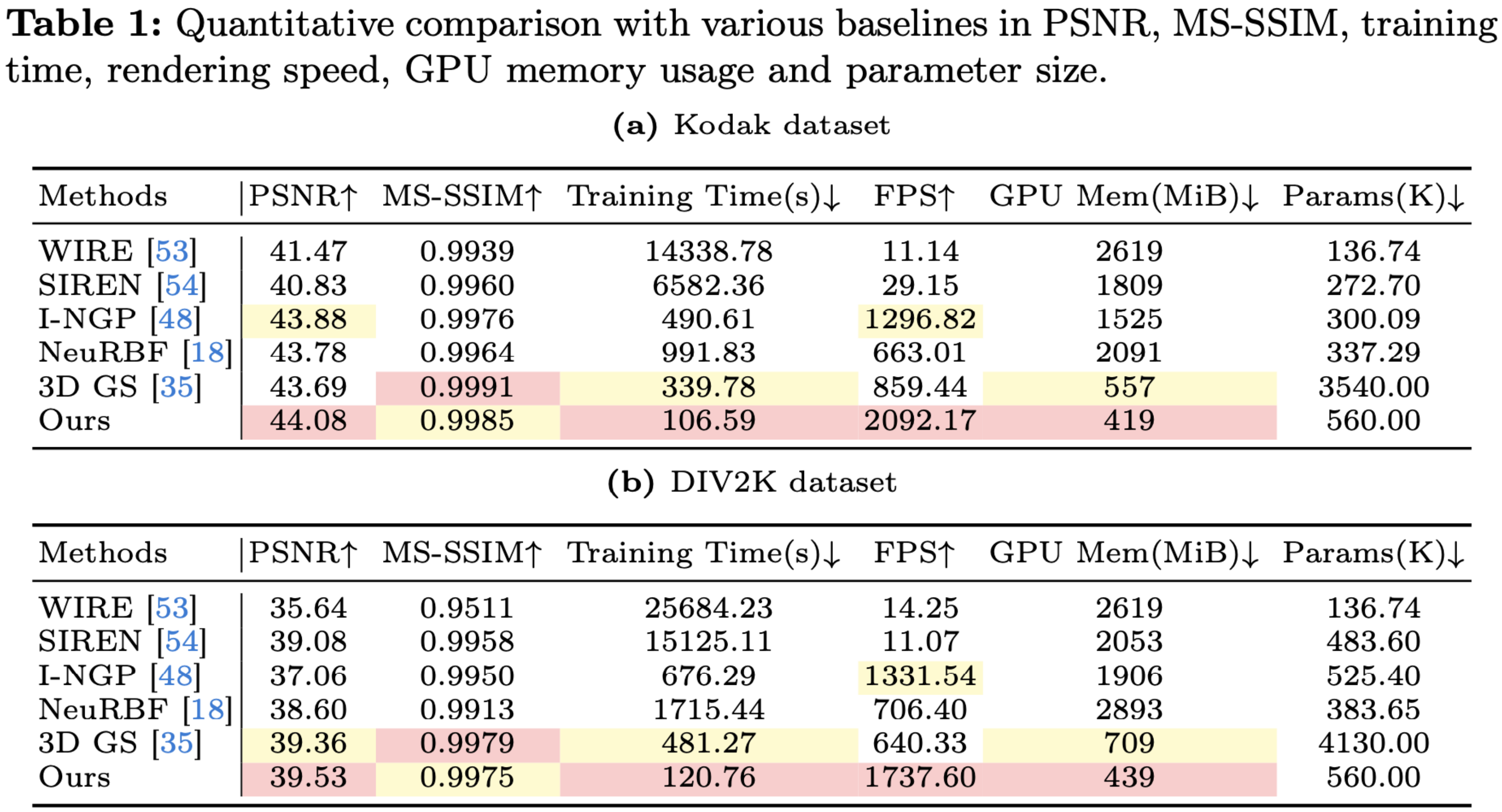


Table 1: Quantitative comparison with various baselines in PSNR, MS-SSIM, training time, rendering speed, GPU memory usage and parameter size.

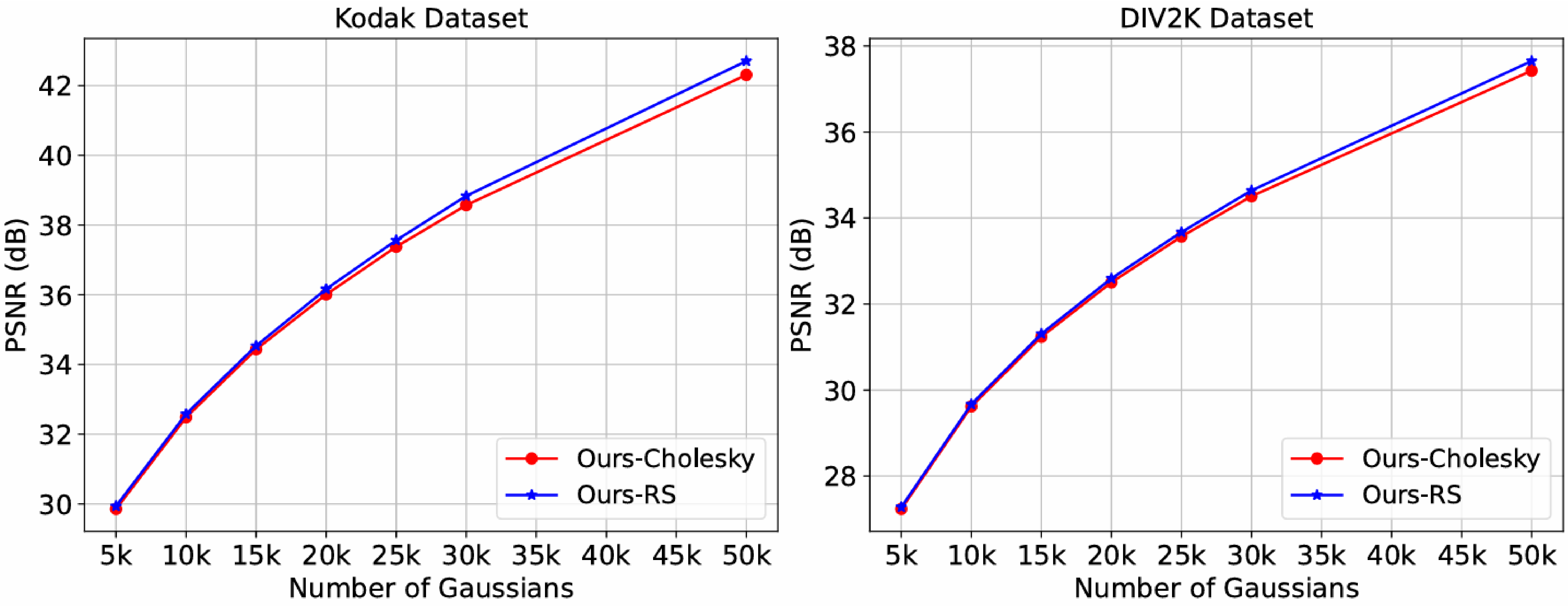
(a) Kodak dataset

Methods	PSNR↑	MS-SSIM↑	Training Time(s)↓	FPS↑	GPU Mem(MiB)↓	Params(K)↓
WIRE [53]	41.47	0.9939	14338.78	11.14	2619	136.74
SIREN [54]	40.83	0.9960	6582.36	29.15	1809	272.70
I-NGP [48]	43.88	0.9976	490.61	1296.82	1525	300.09
NeuRBF [18]	43.78	0.9964	991.83	663.01	2091	337.29
3D GS [35]	43.69	0.9991	339.78	859.44	557	3540.00
Ours	44.08	0.9985	106.59	2092.17	419	560.00

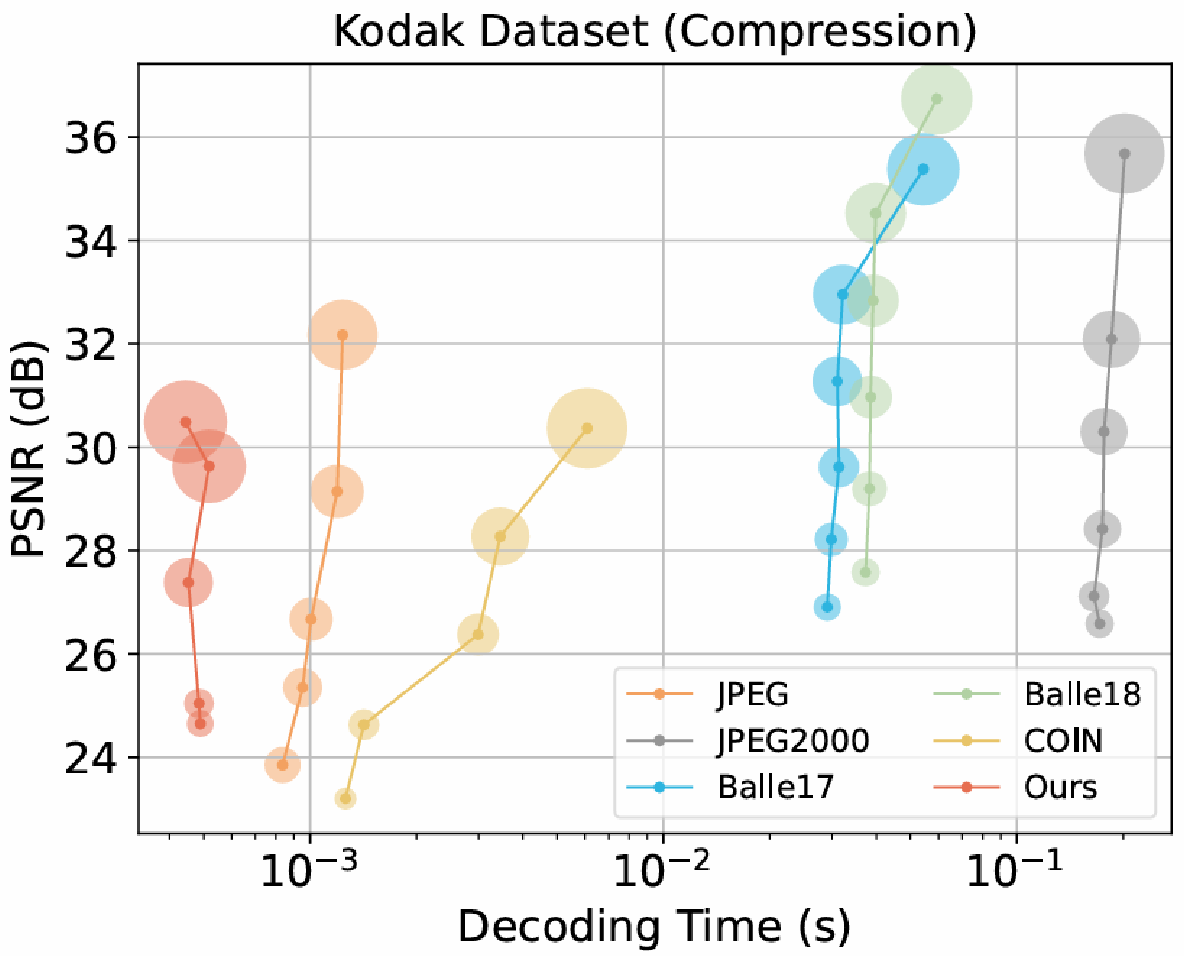
(b) DIV2K dataset

Methods	PSNR↑	MS-SSIM↑	Training Time(s)↓	FPS↑	GPU Mem(MiB)↓	Params(K)↓
WIRE [53]	35.64	0.9511	25684.23	14.25	2619	136.74
SIREN [54]	39.08	0.9958	15125.11	11.07	2053	483.60
I-NGP [48]	37.06	0.9950	676.29	1331.54	1906	525.40
NeuRBF [18]	38.60	0.9913	1715.44	706.40	2893	383.65
3D GS [35]	39.36	0.9979	481.27	640.33	709	4130.00
Ours	39.53	0.9975	120.76	1737.60	439	560.00

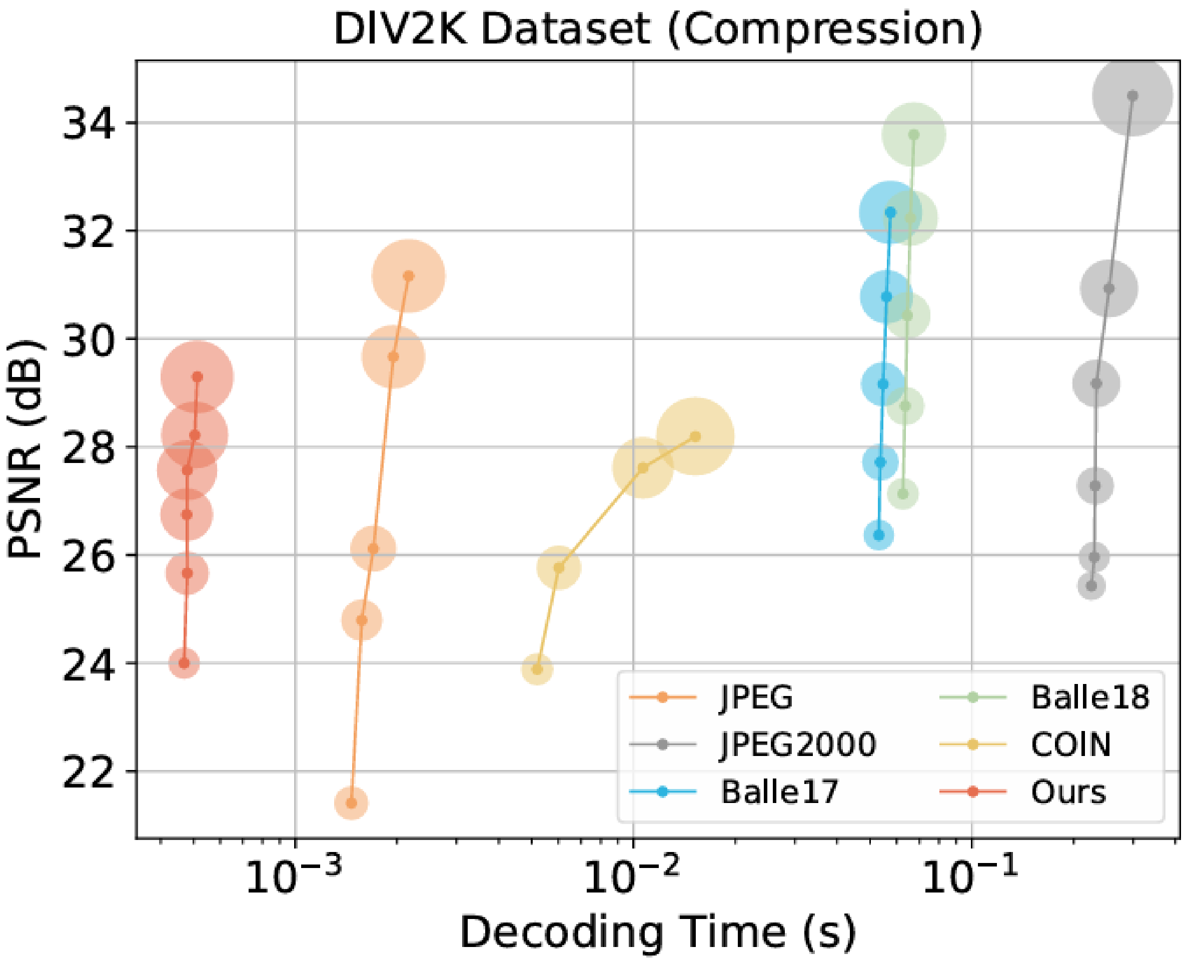
Comprehensive Evaluation: Image Representation



Comprehensive Evaluation: Image Compression



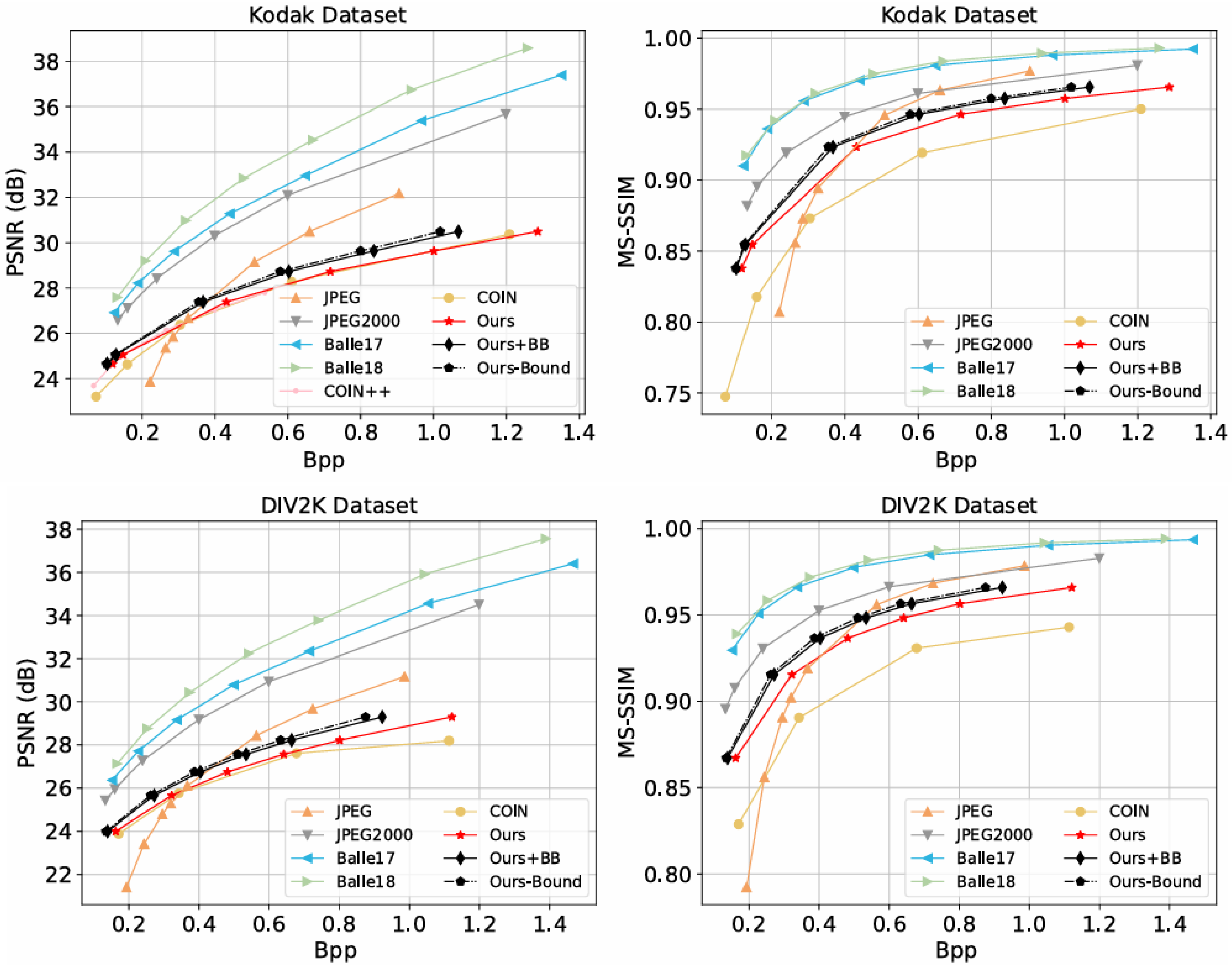
Kodak Dataset (Compression)



DIV2K Dataset (Compression)

THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Comprehensive Evaluation: Image Compression



Comprehensive Evaluation: Image Compression

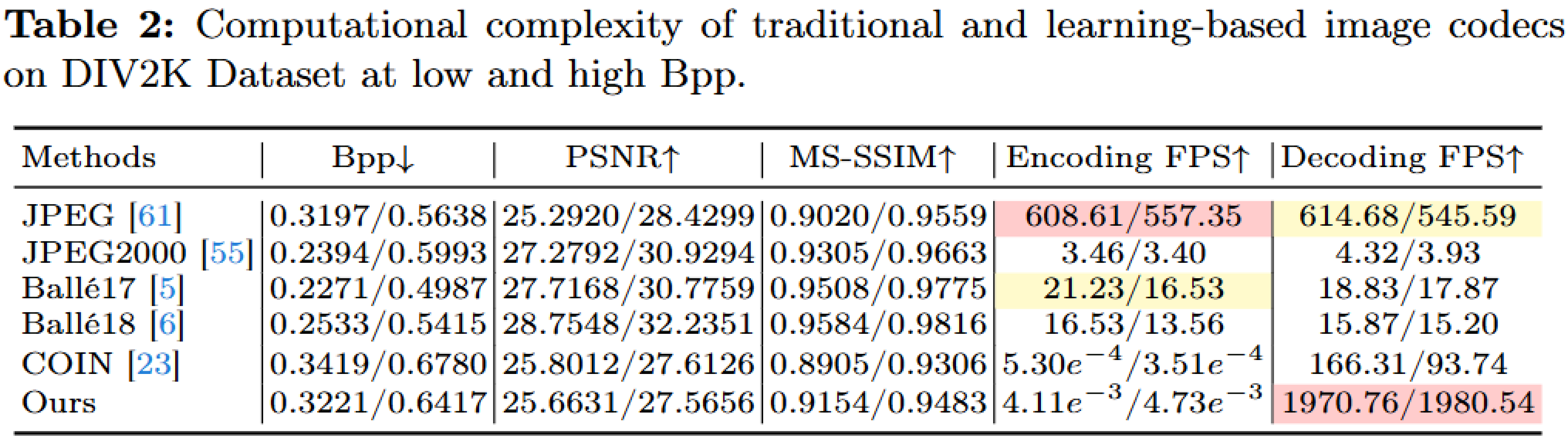
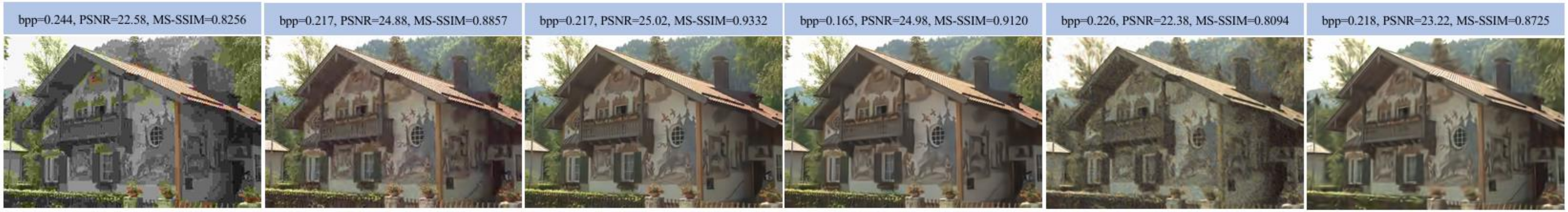


Table 2: Computational complexity of traditional and learning-based image codecs on DIV2K Dataset at low and high Bpp.

Methods	Bpp↓	PSNR↑	MS-SSIM↑	Encoding FPS↑	Decoding FPS↑
JPEG [61]	0.3197/0.5638	25.2920/28.4299	0.9020/0.9559	608.61/557.35	614.68/545.59
JPEG2000 [55]	0.2394/0.5993	27.2792/30.9294	0.9305/0.9663	3.46/3.40	4.32/3.93
Ballé17 [5]	0.2271/0.4987	27.7168/30.7759	0.9508/0.9775	21.23/16.53	18.83/17.87
Ballé18 [6]	0.2533/0.5415	28.7548/32.2351	0.9584/0.9816	16.53/13.56	15.87/15.20
COIN [23]	0.3419/0.6780	25.8012/27.6126	0.8905/0.9306	$5.30e^{-4}/3.51e^{-4}$	166.31/93.74
Ours	0.3221/0.6417	25.6631/27.5656	0.9154/0.9483	$4.11e^{-3}/4.73e^{-3}$	1970.76/1980.54


bpp=0.244, PSNR=22.58, MS-SSIM=0.8256
bpp=0.217, PSNR=24.88, MS-SSIM=0.8857
bpp=0.217, PSNR=25.02, MS-SSIM=0.9332
bpp=0.165, PSNR=24.98, MS-SSIM=0.9120
bpp=0.226, PSNR=22.38, MS-SSIM=0.8094
bpp=0.218, PSNR=23.22, MS-SSIM=0.8725

(a) JPEG (b) JPEG2000 (c) Ballé17 (d) Ballé18 (e) COIN (f) Proposed

Image Representation: Ablation Study

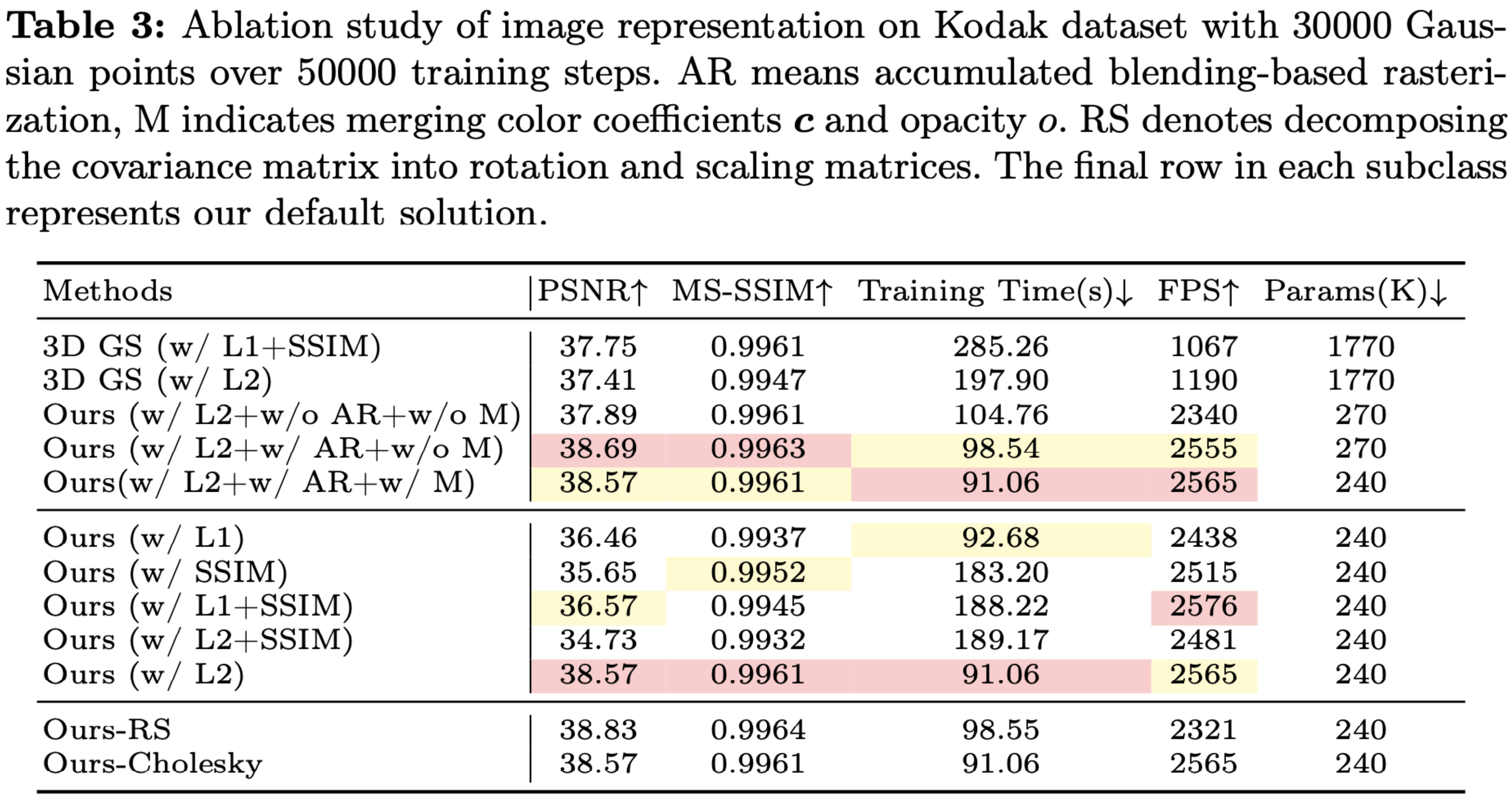


Table 3: Ablation study of image representation on Kodak dataset with 30000 Gaussian points over 50000 training steps. AR means accumulated blending-based rasterization, M indicates merging color coefficients ***c*** and opacity o. RS denotes decomposing the covariance matrix into rotation and scaling matrices. The final row in each subclass represents our default solution.

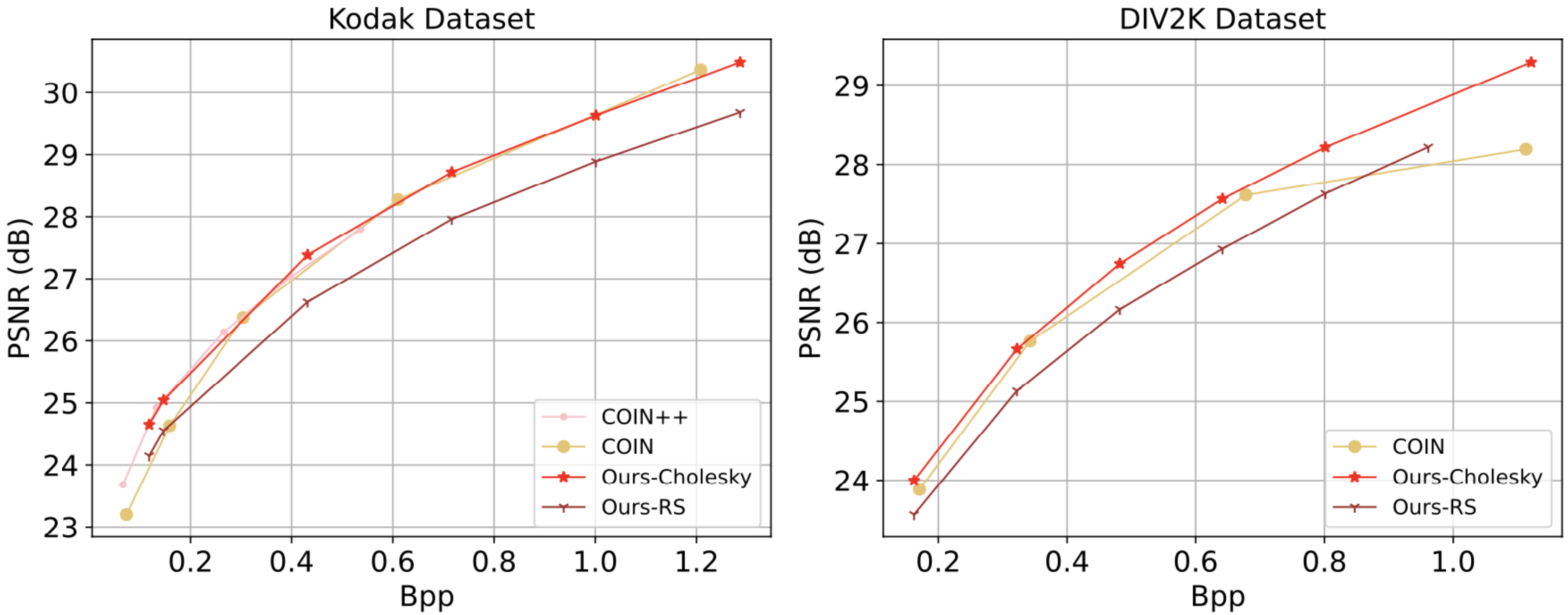
Methods	PSNR↑	MS-SSIM↑	Training Time(s)↓	FPS↑	Params(K)↓
3D GS (w/ L1+SSIM)	37.75	0.9961	285.26	1067	1770
3D GS (w/ L2)	37.41	0.9947	197.90	1190	1770
Ours (w/ L2+w/o AR+w/o M)	37.89	0.9961	104.76	2340	270
Ours (w/ L2+w/ AR+w/o M)	38.69	0.9963	98.54	2555	270
Ours(w/ L2+w/ AR+w/ M)	38.57	0.9961	91.06	2565	240
Ours (w/ L1)	36.46	0.9937	92.68	2438	240
Ours (w/ SSIM)	35.65	0.9952	183.20	2515	240
Ours (w/ L1+SSIM)	36.57	0.9945	188.22	2576	240
Ours (w/ L2+SSIM)	34.73	0.9932	189.17	2481	240
Ours (w/ L2)	38.57	0.9961	91.06	2565	240
Ours-RS	38.83	0.9964	98.55	2321	240
Ours-Cholesky	38.57	0.9961	91.06	2565	240

THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Image Compression: Ablation Study

Table 4: Ablation study of quantization schemes on Kodak dataset. The first row denotes our final solution and is set as the anchor.

Variants	BD-PSNR (dB) ↑	BD-rate (%) ↓	BD-MS-SSIM ↑	BD-rate (%) ↓
Ours	0	0	0	0
(V1) w/o $\mathcal{L}_c$+w/ RVQ + 6bit	-3.145	333.16	-0.0824	337.84
(V2) w/o $\mathcal{L}_c$+w/o RVQ + 6bit	-0.159	7.02	-0.0030	6.14
(V3) w/o $\mathcal{L}_c$+w/o RVQ + 8bit	-0.195	11.69	-0.0127	62.77



Conclusion

- We present a pioneering paradigm of image representation and compression by 2D Gaussian Splatting. With compact 2D Gaussian representation and a novel accumulated blending-based rasterization method, our approach achieves high representation performance with short training duration, minimal GPU memory overhead and remarkably, 2000 FPS rendering speed.

- We develop a ultra-fast neural image codec using vector quantization. It achieves competitive compression performance with COIN and COIN++, while providing around 2000 FPS decoding speed. Furthermore, a partial bits-back coding technique is optionally used to reduce the bitrate.

Source Code

Project Page

Future Direction

- Various Exciting Potential Research Directions:
  - ➢ High-level vision tasks: Adopt the 2D Gaussian as a new tokenizer (Varying size, unlimited by image resolution, carry position information)
    - ☐ How to extract semantic Gaussian?
  - ➢ Basic Generative model: Build a brand-new asymmetric generative paradigm
    - ☐ GM generates a set of Gaussian parameters to render an image: High encoding complexity but very low decoding complexity
  - ➢ Low-level vision tasks: super-resolution, deblurring, …
  - ➢ Text-guided 2D Gaussian Editing

Thank you!

香港科技大學
THE HONG KO G
UNIVERSITY OF SCIENCE
AND TECHNOLOGY