Unified Adaptive Quantization and Representation-Aware Caching for Diffusion Models

Deepanshi Sharma*, Rudra Pratap†, Shivam Jaiswal‡, Dr. Parimala Kancharla§
Indian Institute of Technology Mandi
Himachal Pradesh, India
{s24135, b22125, b22130}@students.iitmandi.ac.in, parimala@iitmandi.ac.in

***Abstract*—Diffusion models achieve strong performance in generative modeling but incur high memory and computational costs due to large model parameters and iterative denoising procedures, limiting deployment in resource-constrained environments. This paper presents a unified framework to improving inference latency and memory efficiency of diffusion models by jointly optimizing activation reuse and weight storage. We introduce a representation-aware caching strategy(ReCaT) which reduces redundant computation during inference by selectively reusing intermediate activations along the diffusion trajectory. ReCaT identifies effective caching points based on representational similarity measured using centered kernel alignment(CKA), enabling informed activation reuse without degrading generative quality. To complement activation reuse, we propose Adaptive Clustering-guided Quantization with Entropy Encoding (ACQ-E) for efficient weight storage. ACQ-E uses histogram-guided clustering to model weights, followed by adaptive mixed-precision quantization and entropy encoding to reduce storage and cost.The proposed framework integrates representation-aware caching and adaptive quantization into a unified optimization pipeline, leading to reduced effective bit-width, smaller model size, and lower computational cost while maintaining generative fidelity. Experiments on standard benchmarks, including LSUN Churches, LSUN Bedrooms, CelebA-HQ, MS-COCO, and PartiPrompt, demonstrate efficiency gains with minimal impact on image quality when compared to full-precision.**

***Index Terms*—Representational Similarity, Entropy modelling, Adaptive Quantization, Clustering, Bit-operation**

I. Introduction

Diffusion models [1] have become one of the most powerful generative modelling tools for high-quality image synthesis. However, their large size and multistep sampling process require high memory and resources, making deployment on edge devices challenging. Existing techniques like CacheQuant [2] and Q&C [3] jointly addresses both weight storage and activation reuse in diffusion models. Following this combined path, we introduce a unified compression framework designed specifically for diffusion models. Our approach combines two methods: ACQ-E and ReCaT. ACQ-E compresses model weights by first grouping them using histogram guided clustering and then applying mixed-precision quantization assigning higher bit-widths to more important clusters and lower precision to less critical ones. A learned entropy model further reduces storage by encoding these quantized and clustered weights efficiently. ReCaT focuses on improving runtime efficiency. It analyses representational similarity across diffusion steps using(CKA) [4] to decide where caching activation will yield most benefit with least impact on image quality. By jointly optimizing weight precision and activation reuse we achieve a balance trade-off between compactness and image quality. Our method reaches an average bit-width of 3.7 and a compression ratio of 7.9× from a full precision model, while maintaining FID [5] scores across datasets as shown in Table II. These results show that compression of diffusion models is possible without compromising generative quality.

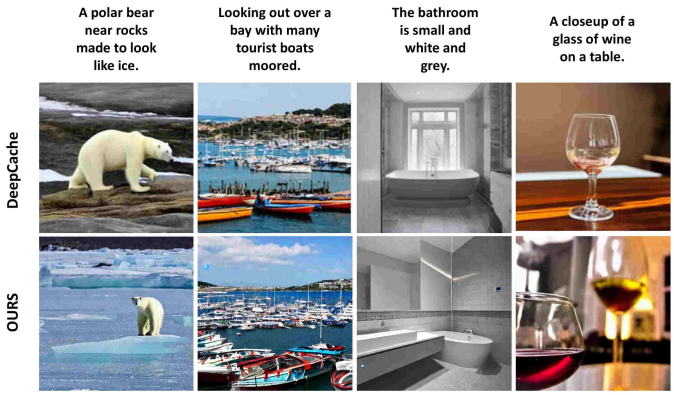

Fig. 1. Qualitative comparison of image generation results. Visual results for four prompts are shown, comparing DeepCache [6] (top row) with our method (bottom row).

In summary, our main contributions are as follows:

- We propose ACQ-E, a histogram-guided clustering and adaptive mixed-precision quantization method for efficient weight storage, complemented by learned entropy encoding for compact model representation.
- We introduce ReCaT, a representation-aware activation caching strategy that selects effective caching points along the diffusion trajectory based on representational similarity measured using centered kernel alignment [4].
- We propose a unified framework that improves runtime efficiency of diffusion models through representation-aware activation reuse and efficient weight storage.

II. Related Works

A. Diffusion Model Acceleration

Diffusion models achieve high-quality generative performance but require iterative denoising steps, making inference

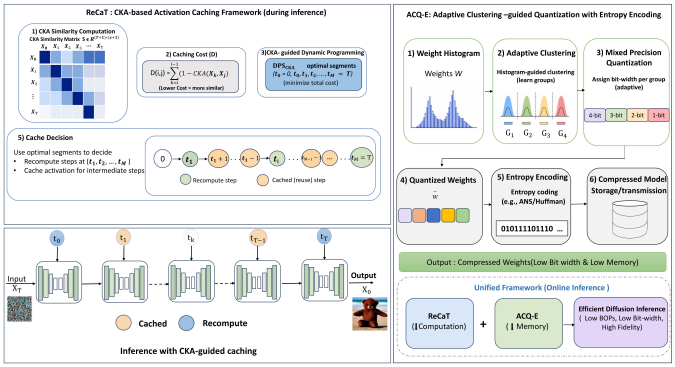

Fig. 2. Combined architecture of the proposed diffusion model compression framework. This integrates ReCaT, a runtime activation reuse mechanism driven by representational similarity (CKA) across diffusion timesteps with ACQ-E, a weight compression module based on histogram guided clustering, mixed precision quantization and learned entropy modeling.

slower and more resource intensive. Early acceleration strategies focused on reducing the number of sampling steps using improved solvers such as DDIM, DPM-Solver. More recent approaches explore architectural modifications or knowledge distillation to smaller student models; however, these methods require retraining and may not generalize across diverse diffusion backbones.

B. Weight Compression for Generative Models

Model compression techniques like pruning, quantization, and clustering have been widely studied. Applying these methods to generative models is challenging because generative quality is highly sensitive to precision loss. Post-training quantization (PTQ) methods adapt diffusion weights to lower precision demonstrating the feasibility of 4–8 bit reductions. More advanced techniques incorporate learned quantization parameters or layer-wise calibration.

C. Activation Reuse and Caching Techniques

Diffusion models process highly similar latent representations across timesteps suggesting opportunities for activation caching. Early caching approaches reused attention maps or intermediate feature blocks to reduce redundant compute. More recent works like CacheQuant [2] and Q&C [3] explicitely formalize activation reuse for diffusion models selecting specific blocks or timesteps where cached activations can be substituted with minimal error.

D. Joint Optimization of Compression and Speed

Recent research attempts to jointly compress weights while reducing runtime compute. CacheQuant [2] and Q&C [3] represent early steps in this direction by combining weight quantization with activation reuse. But their quantization is uniform and their caching selection strategies is rule based.

E. Entropy Models in Neural Compression

Entropy models have been widely used in image and neural compression to reduce storage by learning probability [9] of

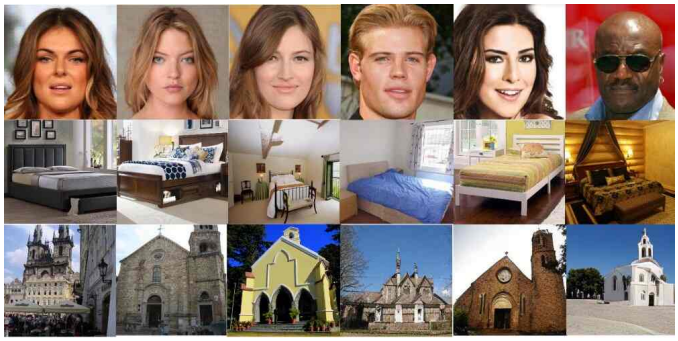

Fig. 3. Qualitative results on standard image generation benchmarks. Sample images generated on CelebA-HQ [7], LSUN Bedrooms [8] and LSUN Churches [8] datasets are shown.

quantized representations. In generative model compression learned entropy models have not been thoroughly explored. Existing approaches often rely on fixed Huffman [10] or arithmetic coding applied post-hoc.

III. Proposed Methodology

We propose a unified framework for accelerating inference in diffusion models by combining advanced weight compression with principled activation reuse. The framework consists of two core modules:

- **ACQ-E (Adaptive Clustering-guided Quantization with Entropy Encoding)**: This module compresses model weights via a pipeline that combines histogram-aware clustering and mixed-precision quantization to achieve runtime-efficient low-bit weights. Learned entropy modeling is additionally employed to enable effective lossless compression for storage and transmission.
- **ReCaT (Representational Similarity-driven Caching Technique)**: This module enables safe activation reuse by evaluating representational similarity using Centered Kernel Alignment (CKA) [4], ensuring minimal degradation in generation quality.

By integrating these modules, our framework produces a compact and efficient diffusion model suitable for resource-constrained deployment. The following sections provide detailed descriptions of each component and their interaction within the system.

A. Adaptive Clustering-guided Quantization with Entropy Encoding(ACQ-E)

It aims to reduce the memory footprint of diffusion models by compressing their weights in a more principled way than uniform quantization. Instead of assigning the same bit width across all layer we analyze model's weight distribution and adapt precision accordingly.

We begin by extracting weights from the pre-trained diffusion model and constructing a histogram of their distribution. Using this information we perform clustering on model weights and partition these clusters into four groups:$[G = G_1, G_2, G_3, G_4]$. Once weights are clustered and divided into seperate groups each group is assigned a fixed bit-width based on the density of clusters as demonstrated in Figure 4.

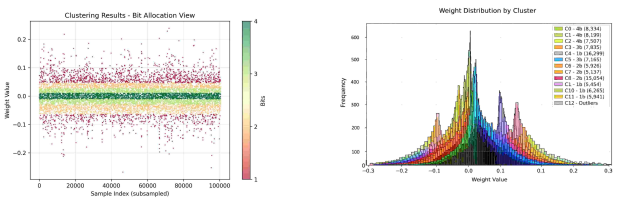

Fig. 4. Group wise bit allocation results for different groups of clusters (left) and the weight distributions across clusters (right) demonstrating how clustering and grouping captures distinct weight statistics.

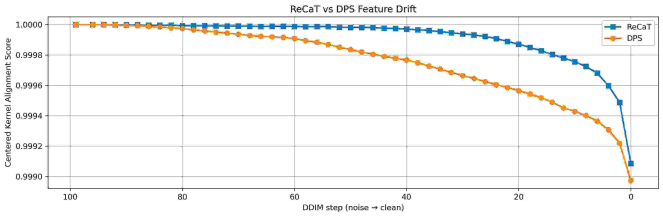

Fig. 5. Illustration of semantic alignment evolution across diffusion steps where higher CKA [4] indicates stronger alignment with base model.

Formally $b_1 = 4, b_2 = 3, b_3 = 2, b_4 = 1$. Within each group weights are uniformly quantized using:

$$\hat{w} = round\left(\frac{w - w_{\min}}{\Delta_i}\right)\Delta_i + w_{\min} \tag{1}$$

where step size is:

$$\Delta_i = \frac{w_{\max} - w_{\min}}{2^{b_i} - 1}$$

1) Learned Entropy Modelling:: Storage efficiency is further improved through a learned entropy modeling framework that captures the distribution of quantized weights. Instead of explicitly storing quantized values, the weights are encoded according to probabilities predicted by the entropy model. The expected coding cost is quantified by the entropy

$$\mathcal{H} = -\sum_{\hat{w}\in\mathcal{Q}} P(\hat{w}) \log_2 P(\hat{w}), \tag{2}$$

where $\mathcal{Q}$ denotes the set of quantized weight values and $P(\hat{w})$ represents the corresponding probability mass function. For each cluster group, the entropy model predicts a discrete probability mass function parameterized by learned logits, which are transformed into valid probability distributions using a softmax operation. Differentiable optimization in the presence of discrete quantization is enabled through a soft indexing mechanism that interpolates probabilities between neighboring quantization levels.

Under this formulation, the estimated bit cost associated with a quantized weight $\hat{w}$ is expressed as

$$\text{Bits}(\hat{w}) = -\log_2 P_{\text{interp}}(\hat{w}), \tag{3}$$

where $P_{\text{interp}}(\hat{w})$ denotes the interpolated probability obtained from the entropy model.

The combination of adaptive clustering-guided quantization and learned entropy coding allows ACQ-E to jointly reduce numerical precision and statistical redundancy.

B. *Representational Similarity-driven Caching Technique*

While ACQ-E focuses on reducing the storage cost of model weights, ReCaT targets inference-time efficiency by identifying timesteps along the reverse diffusion process where intermediate activations can be safely cached and reused. Caching-based acceleration methods exploit the observation that diffusion inference involves a sequence of iterative denoising steps, many of which exhibit redundant computation. By reusing cached activations at selected timesteps, redundant operations can be avoided, leading to a reduction in overall bit-operation (BOP) cost during inference.

ReCaT departs from heuristic scheduling by selecting caching points based on representational similarity within the model itself. Using the proposed kernel alignment score, ReCaT selects a subset of timesteps along the reverse diffusion process at which the model is fully executed and the resulting intermediate activations are stored in cache memory. For the subsequent timesteps, these cached activations are reused until the next selected full-execution step. This representation-aware scheduling enables safe reuse of cached features while preserving the fidelity of the diffusion trajectory. Diffusion activations tend to evolve smoothly across adjacent timesteps, resulting in highly similar intermediate representations. ReCaT exploits this property by explicitly measuring representational similarity and reusing cached activations only when similarity is sufficiently high as represented in Figure 5.

1) Representation-Aware Caching via CKA: CacheQuant [2] framework employs an L1-distance-based dynamic programming formulation to compute the caching cost:

$$D_k(i,j) = \sum_{t=i+1}^{j} \left\| X_{g_i} - X_{g_t} \right\|_1, \tag{4}$$

where X_{g_i} and X_{g_t} denote activations at timesteps i and t, respectively. While simple, this formulation primarily captures magnitude differences and is insensitive to structural alignment in the representation space.

In contrast, ReCaT replaces this distance metric with a similarity measure based on centered kernel alignment (CKA) [4], which captures higher-order structural similarity between representations. Given activations X and Y at timesteps i and t, CKA [4] is computed as

$$K = XX^\top, \quad L = YY^\top, \quad H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top,$$

$$K_c = HKH, \quad L_c = HLH, \quad \text{HSIC}_{\text{num}} = \langle K_c, L_c \rangle_F,$$

$$\text{Norm} = \langle K_c, K_c \rangle_F \langle L_c, L_c \rangle_F + \epsilon,$$

$$\text{CKA}(X,Y) = \frac{\text{HSIC}_{\text{num}}}{\text{Norm}}.$$

Using this similarity measure, caching cost is redefined as

$$D_k(i,j) = \sum_{t=i+1}^{j} \left(1 - \text{CKA}(i,t)\right). \tag{5}$$

TABLE I
COMPARISON ON MS-COCO [11] AND PARTIPROMPTS [12] FOR TEXT TO IMAGE GENERATION.

Method	Prec.(W/A)	BOPs↓	Size↓	Retrain	MS-COCO [11]		PartiPrompts [12]
					FID [5]↓	CLIP [13]↑	CLIP [13]↑
PLMS-50 steps	32/32	346.96T	4112.5	×	25.59	**26.89**	**27.23**
PLMS-20 steps	32/32	138.78T	4112.5	×	24.70	26.74	27.04
Deepcache [6]-N=10	32/32	133.58T	4112.5	×	23.45	26.71	26.86
Small SD [14]	16/16	57.28T	1158.6	✓	29.43	26.02	25.99
BK-SDM-base [14]	16/16	57.30T	1160.2	✓	28.47	26.21	26.53
BK-SDM-small [14]	16/16	55.75T	966.6	✓	30.10	25.22	25.76
BK-SDM-tiny [14]	16/16	52.50T	648.5	✓	31.82	25.10	25.51
CacheQuant [2]-N=5	8/8	8.44T	1029.7	×	23.74	26.87	27.13
CacheQuant [2]-N=5	4/8	4.27T	515.9	✓	23.23	26.77	27.15
Ours-N=5	4/8	**3.95T**	**407.64**	✓	**21.14**	21.76	22.79

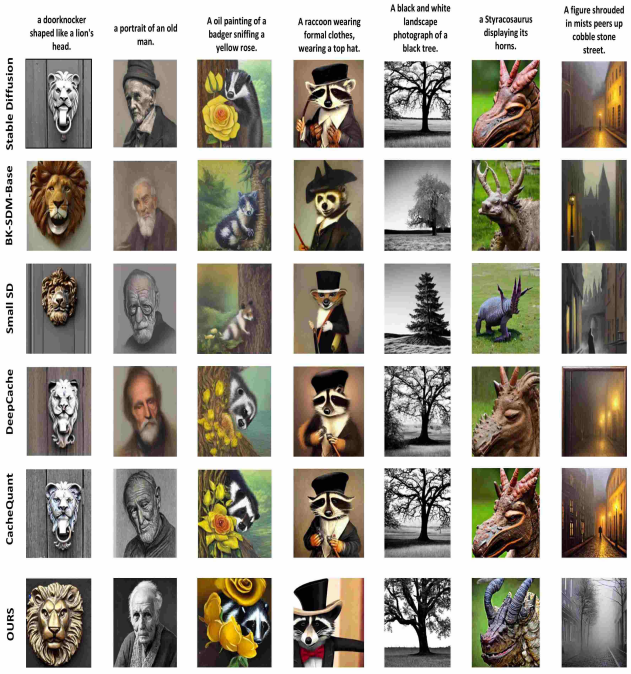

Fig. 6. Qualitative comparison across diffusion models and caching strategies. Images generated from multiple text prompts are shown for Stable Diffusion [1], BK-SDM Base [14], Small SD [14], DeepCache [6], CacheQuant [2] and our method.

Lower values of $D_k(i,j)$ indicate that representations across timesteps remain sufficiently aligned, allowing cached activations to be safely reused until the next full-execution point.

C. *Why Does the Unified Framework Work?*

The proposed framework benefits from two complementary properties of diffusion models. First, diffusion model weights exhibit significant redundancy and non-uniform importance, making them well suited for adaptive clustering and mixed-precision quantization. Second, latent representations evolve smoothly across diffusion timesteps. ACQ-E exploits weight redundancy to reduce storage cost while ReCaT exploits temporal redundancy to reduce inference computation. Since the two modules operate on different sources of redundancy, their gains leads to simultaneous reductions in model size and BOPs with minimal impact on generative quality.

TABLE II
COMPARISON OF BOPS AND FID [5] ACROSS DATASETS.

Dataset	Method	BOPs↓	FID [5]↓
CelebA-HQ [7] 256 × 256	DDPM [15]	-	4.19
	Deepcache [6]-N=3	-	4.70
	Deepcache [6]-N=5	-	5.28
	CacheQuant [2]-N=5	-	5.28
	CacheQuant [2]-N=10	-	8.19
	Ours- N=3	**0.96T**	**4.26**
	Ours- N=5	**0.56T**	**4.28**
LSUN-Church [8] 256 × 256	LDM-8 [1]	19.10T	3.99
	Deepcache [6]-N=2	10.07T	4.43
	CacheQuant [2]-N=2	0.63T	3.52
	Ours-N=2	**0.28T**	**3.26**
	Deepcache [6]-N=3	7.18T	5.10
	CacheQuant [2]-N=3	0.45T	3.66
	Ours-N=3	**0.19T**	**3.15**
	Deepcache [6]-N=5	4.65T	6.74
	CacheQuant [2]-N=5	0.29T	3.71
	Ours-N=5	**0.11T**	**2.98**
LSUN-Bedroom [8] 256 × 256	LDM-4 [1]	98.36T	10.49
	Deepcache [6]-N=2	52.23T	11.21
	CacheQuant [2]-N=2	3.26T	8.85
	Ours-N=2	**1.42T**	**8.06**
	Deepcache [6]-N=3	37.49T	11.86
	CacheQuant [2]-N=3	2.34T	9.27
	Ours-N=3	**0.96T**	**8.24**
	Deepcache [6]-N=5	24.59T	14.28
	CacheQuant [2]-N=5	1.54T	10.29
	Ours-N=5	**0.56T**	**9.61**

IV. EXPERIMENTS

This section elaborates the experiments done and evaluates the effectiveness of proposed framework on multiple benchmarks. We focus on measuring compression efficiency, cost reductions and generative quality of the framework.

A. *Experimental Setup*

1) Hardware Setup: All experiments were conducted on NVIDIA H200 GPUs with 141GB memory using CUDA 12.4

2) Models and Baselines: All experiments are conducted on pretrained diffusion models. Our primary baseline is CacheQuant [2] which combines uniform weight quantization with activation caching.

All image datasets follow standard evaluation protocols and for text to image generation we use official prompt sets.

TABLE III
THE EFFECT OF DIFFERENT COMPONENTS PROPOSED IN THE PAPER (BITS REPRESENT THE AVERAGE WEIGHT BITS REQUIRED).

Method	Prec.(W/A)	Retrain	FID [5]↓	Bits↓	Size↓
Deepcache [6]-N=20	32/32	×	8.08	32	412.67
Naive Caching + Standard Quantization	8/8	×	15.36	8	379.57
ReCaT	4/8	×	10.82	4	190.44
ReCaT + ACQ-E(without entropy)	4/8	×	**5.07**	4	190.44
ReCaT + ACQ-E(with entropy)	4/8	✓	5.21	**3.79**	**189.72**

Fig. 7. Effect of prompt specificity on generated scene composition. The left image, generated from a generic prompt like "figure shrouded in mist" depicts an ambiguous human like presence on the wall but the right image, produced using a more specific prompt "a shadow like figure of a man on a road shrouded in mist" results in a clearly defined person figure on the road.

3) Implementation Details: For ACQ-E, weights are clustered using histogram guided density estimation and quantized using mixed precision (4/3/2/1-bits) [9]. A learned entropy model is trained per cluster to estimate encoding cost. For ReCaT, caching schedules are computed using a dynamic programming strategy with a CKA [4]-based similarity objective. All experiments are performed using same inference settings.

B. Ablation Study

Table III analyzes the impact of individual components in the proposed framework. Starting from DeepCache [6], baseline introducing ReCaT leads to noticeable improvement in FID [5] while reducing the average weight precision. Adding ACQ-E without entropy guidance further improves generation quality, indicating the benefit of adaptive quantization [9]. Finally, incorporating entropy aware ACQ-E achieves good performance by reducing the average bit width and model size.

V. CONCLUSION

This work presented a unified compression framework for diffusion models that jointly addresses weight strage efficiency and inference time computation. ACQ-E exploits the inherent structure of diffusion model weights through histogram guided clustering, mixed precision quantization and learned entropy modelling which enables compact representations with low average bit-width. ReCaT uses representations similarity across diffusion timesteps to identify safe caching points which reduces redundant computation during iterative denoising. Together these components provide good compression along with efficiency and generative quality as it can be seen in Figure 1, Figure 3 and Figure 6. Experimental results on CelebA-HQ [7], LSUN Churches [8], LSUN Bedrooms [8] and large scale text to image benchmarks shows consistent reductions in model size and BOPs.

ACKNOWLEDGEMENTS

The authors acknowledge financial support from ANRF under the PMECRG scheme(Grant No. ANRF/ECRG/2024/004417/ENS; Project No. IITM/ANRF/KP/529/2025).

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] X. Liu, Z. Li, and Q. Gu, "Cachequant: Comprehensively accelerated diffusion models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 23 269–23 280.

[3] X. Ding, X. Li, H. Qin, and Z. Chen, "Q&c: When quantization meets cache in efficient image generation," *arXiv preprint arXiv:2503.02508*, 2025.

[4] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 179–192, 2015.

[5] S. Jayasumana, S. Ramalingam, A. Veit, D. Glasner, A. Chakrabarti, and S. Kumar, "Rethinking fid: Towards a better evaluation metric for image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9307–9315.

[6] X. Ma, G. Fang, and X. Wang, "Deepcache: Accelerating diffusion models for free," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 762–15 772.

[7] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[8] F. Yu, Y. Zhang, and S. Song, "Lsun: Construction of a large-scale image dataset using deep learning," *arXiv preprint arXiv:1506.03365*, 2015.

[9] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[10] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4394–4402.

[11] T.-Y. Lin, M. Maire, and S. Belongie, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[12] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *arXiv preprint arXiv:2206.10789*, vol. 2, no. 3, p. 5, 2022.

[13] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," in *Proceedings of the 2021 conference on empirical methods in natural language processing*, 2021, pp. 7514–7528.

[14] B.-K. Kim, H.-K. Song, T. Castells, and S. Choi, "Bk-sdm: A lightweight, fast, and cheap version of stable diffusion," in *European Conference on Computer Vision*. Springer, 2024, pp. 381–399.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.