BinaryDM: Accurate Weight Binarization for Efficient Diffusion Models

Xingyu Zheng[1], **Xianglong Liu**[✉1], **Haotong Qin**[2], **Xudong Ma**[1], **Mingyuan Zhang**[3], **Haojie Hao**[1], **Jiakai Wang**[4], **Zixiang Zhao**[5], **Jinyang Guo**[1], **Michele Magno**[2]
[1]Beihang University [2]ETH Zürich [3]Nanyang Technological University
[4]Zhongguancun Laboratory [5]Xi'an Jiaotong University
{zhengxingyu,xlliu,macaronlin,haojiehao,jinyangguo}@buaa.edu.cn
{haotong.qin,michele.magno}@pbl.ee.ethz.ch mingyuan001@e.ntu.edu.sg
wangjk@zgclab.edu.cn zixiangzhao@stu.xjtu.edu.cn

Abstract

With the advancement of diffusion models (DMs) and the substantially increased computational requirements, quantization emerges as a practical solution to obtain compact and efficient low-bit DMs. However, the highly discrete representation leads to severe accuracy degradation, hindering the quantization of diffusion models to ultra-low bit-widths. This paper proposes a novel weight binarization approach for DMs, namely **BinaryDM**, pushing binarized DMs to be accurate and efficient by improving the representation and optimization. From the representation perspective, we present an *Evolvable-Basis Binarizer* (EBB) to enable a smooth evolution of DMs from full-precision to accurately binarized. EBB enhances information representation in the initial stage through the flexible combination of multiple binary bases and applies regularization to evolve into efficient single-basis binarization. The evolution only occurs in the head and tail of the DM architecture to retain the stability of training. From the optimization perspective, a *Low-rank Representation Mimicking* (LRM) is applied to assist the optimization of binarized DMs. The LRM mimics the representations of full-precision DMs in low-rank space, alleviating the direction ambiguity of the optimization process caused by fine-grained alignment. Comprehensive experiments demonstrate that BinaryDM achieves significant accuracy and efficiency gains compared to SOTA quantization methods of DMs under ultra-low bit-widths. With 1-bit weight and 4-bit activation (W1A4), BinaryDM achieves as low as 7.74 FID and saves the performance from collapse (baseline FID 10.87). As the first binarization method for diffusion models, W1A4 BinaryDM achieves impressive 15.2$\times$ OPs and 29.2$\times$ model size savings, showcasing its substantial potential for edge deployment.

1 Introduction

Diffusion models (DMs) [11, 31] have shown excellent capabilities in generation tasks in various fields, such as image [11, 31, 32], vision [20, 10], and speech [22, 24, 14]. DMs have become one of the most popular generative model paradigms with significant quality and diversity advantages. DMs generate data through the iterative noise estimates, while up to 1000 iterative steps slow the inference process and rely on expensive hardware resources. Although some proposed methods can effectively reduce the number of iterations to dozens of times [30, 28, 23, 1], the complex neural network of DMs also results in a large number of floating point calculations and memory usage in each step, which hinders the efficient deployment and inference on edge. Therefore, the compression of DMs has been widely studied as a practical technology to accelerate the iterative process and reduce the inference cost, including quantization [16, 29], distillation [27, 19, 21], pruning [5], *etc*.

Machine Learning and Compression Workshop @ NeurIPS 2024.

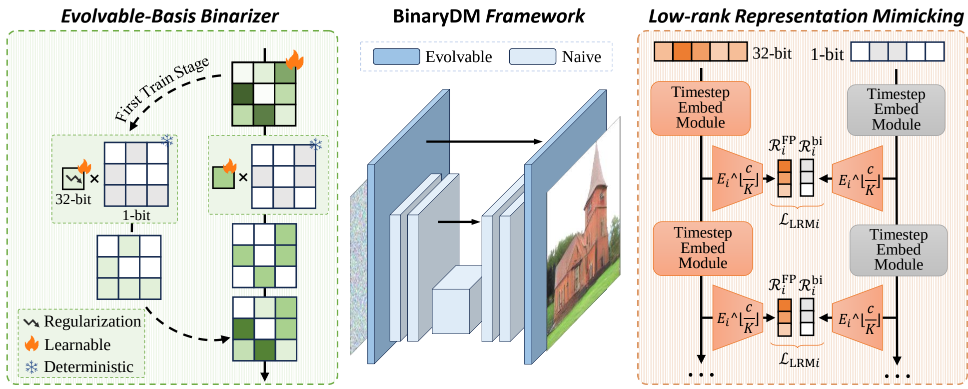


Figure 1: Overview of BinaryDM, consisting of Learnable Multi-basis Binarizer to enhance information representation and Low-rank Representation Mimicking to improve optimization direction.

Low-bit quantization emerges as a practical approach to compress deep learning models by reducing the bit-width of parameters [35, 7]. Thus, with quantization, diffusion models can enjoy the compression and acceleration brought by fixed-point parameters and computation in inference [16, 15, 8, 29]. The 1-bit quantization, namely binarization, allows the binarized model to enjoy compact 1-bit parameters and efficient computation [18, 34, 33]. With the most aggressive bit-width, 1-bit weights can lead to up to 32× size reduction and replace expensive floating-point multiplications with addition constructions during inference, thus saving resources significantly [25, 6].

However, binarized DMs suffer significant performance degradation compared to their full-precision counterparts. The performance decline primarily arises from two aspects: **First**, weight binarization severely restricts the feature extraction capability of DM, causing significant damage to information in critical representations of generative models. **Second**, introducing discrete binarization functions in DMs poses a significant hurdle to stable convergence.

In this paper, we propose **BinaryDM** to push the weights of diffusion models toward binarization. The proposed method pushes the weights of DMs toward accurate and efficient binarization, considering the representation and computation properties. BinaryDM is composed of two novel techniques: *From the representation perspective*, we present an Evolvable-Basis Binarizer (EBB) to recover the representations generated by the binarized DM. EBB first applies dual sets of binary bases with learnable scalars to significantly enhance the feature extraction capability of the initial binarized weights, then evolves the high-order bases to the single-basis form guided by regularization loss. It is selectively applied only to key parameter locations of the DM architecture to reduce unnecessary evolution processes, thereby easing the training burden and making the evolution smoother. *From the optimization perspective*, a Low-rank Representation Mimicking (LRM) is incorporated to enhance the binarization-aware optimization of DMs. LRM projects binarized and full-precision representations to low-rank, enabling the optimization of binarized DM to focus on the principal direction and mitigate direction ambiguity caused by the representation complexity of generation.

2 BinaryDM

2.1 Preliminaries

In the forward process of diffusion models, Gaussian noise is added to data $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$ in T times via a schedule β_t controlling noise strength, the process can be expressed as

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right)=\mathcal{N}\left(\boldsymbol{x}_t ; \sqrt{1-\beta_t} \boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\right), \tag{1}$$

where $\boldsymbol{x}_t \in \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ denote the noisy samples at t-th step. The reverse process aims to generate samples by removing noise, approximating the unavailable conditional distribution $q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$ with learned distributions $p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$, which can be expressed as

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)=\mathcal{N}\left(\boldsymbol{x}_{t-1} ; \tilde{\boldsymbol{\mu}}_\theta\left(\boldsymbol{x}_t, t\right), \tilde{\beta}_t \boldsymbol{I}\right). \tag{2}$$

The mean $\tilde{\boldsymbol{\mu}}_\theta\left(\boldsymbol{x}_t, t\right)$ and variance $\tilde{\beta}_t$ could be derived using the reparameterization [11]:

$$\tilde{\boldsymbol{\mu}}_\theta\left(\boldsymbol{x}_t, t\right)=\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t, t\right)\right), \qquad \tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t, \tag{3}$$

where $\alpha_t=1-\beta_t, \bar{\alpha}_t=\prod_{i=1}^t \alpha_i$, and $\boldsymbol{\epsilon}_\theta$ denotes a function approximation with the learnable parameter θ, which predicts $\boldsymbol{\epsilon}$ from $\boldsymbol{x}_t$. The U-Net with spatial transformer layers is applied as the architecture of the noise estimation network in common practices. For the training of DMs, a simplified variant of the variational lower bound is usually applied as the loss function to achieve high sample quality, which can be expressed as

$$\mathcal{L}_{\text{simple}}=\mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}_t, t\right)\right\|^2\right]. \tag{4}$$

The binarization and quantization compress and accelerate the noise estimation model by discretizing weights and activations to low bit-width. In the baseline of the binarized diffusion model, the weight $\boldsymbol{w} \in \theta$ is binarized to 1-bit by $\boldsymbol{w}^{\text{bi}}=\sigma \operatorname{sign}(\boldsymbol{w})$ [25, 3], where sign function confine $\boldsymbol{w}$ to +1 or -1 with 0 thresholds, $\boldsymbol{w}^{\text{bi}} \in \theta^{\text{bi}}$ denotes the binarized weight, and θ^{bi} denotes the binarized noise estimation network. σ is the floating-point scalar, which is initialized as $\frac{\|\boldsymbol{w}\|}{n}$ (n denotes the number of weight elements) and learnable during training process following [25, 18]. The activation is quantized by the LSQ quantizer [4]. With the $32\times$ compressed weight, the computation of noise estimation can also be replaced with integer additions, achieving significant compression and acceleration.

2.2 Evolvable-Basis Binarizer

In the current baseline, weights are quantized to 1-bit values to economize on storage and computation during inference, and activations can be quantized to integers. However, the extensive discretization of weights to binary in DMs results in a notable deterioration of the generated representations. Previous works present a straightforward approach that enhances binarized parameters via higher-order residual bases [17, 12, 2] have achieved significant success in terms of accuracy, but the introduced additional bases result in substantial additional hardware overhead, making them unsuitable for practical deployment on existing hardware architectures.

To utilize the representation capability of high-order bases while avoiding redundant costs during inference, we sought to use residual binarized structures as transitional structures and evolve during training. This would allow fully binarized DMs to start from a more favorable initial state, resulting in a smoother optimization process and better final outcomes.

We propose the Evolvable-Basis Binarizer (EBB) to address the adaptation challenges faced by binarized DMs during the early stages of optimization due to structural limitations. EBB is implemented in two stages during training. The first stage uses higher-order residual multi-basis with regularization penalties, which then transitions into the second stage with simple single-basis binary weights.

Learnable Multi-Basis. In the forward propagation of the first stage, EBB is defined as

$$\boldsymbol{w}_{\text{EBB}}^{\text{bi}}=\sigma_{\text{I}} \operatorname{sign}(\boldsymbol{w})+\sigma_{\text{II}} \operatorname{sign}\left(\boldsymbol{w}-\sigma_1 \operatorname{sign}(\boldsymbol{w})\right), \tag{5}$$

where the σ_{I} and σ_{II} are learnable scalars which are initialized as $\sigma_{\text{I}}^0=\frac{\|\boldsymbol{w}\|}{n}$ and $\sigma_{\text{II}}^0=\frac{\|\boldsymbol{w}-\sigma_1 \operatorname{sign}(\boldsymbol{w})\|}{n}$, respectively, $\|\cdot\|$ denotes the $\ell 2$-normalization. The inference of layer binarized by EBB involves the computation of multiple bases. For instance, the convolution in binarized DM is

$$o=\boldsymbol{a} \times \boldsymbol{w}_{\text{EBB}}^{\text{bi}}=\sigma_{\text{I}}(\boldsymbol{a} \otimes \operatorname{sign}(\boldsymbol{w}))+\sigma_{\text{II}}\left(\boldsymbol{a} \otimes \operatorname{sign}\left(\boldsymbol{w}-\sigma_1 \operatorname{sign}(\boldsymbol{w})\right)\right), \tag{6}$$

where $\boldsymbol{a}$ denotes the activation, and $\times$ and $\otimes$ denote the convolution consisting of multiplication and addition instructions [25, 13], respectively.

In the backward propagation of EBB, the gradient of the learnable scalars is calculated as follows:

$$\frac{\partial \boldsymbol{w}_{\text{EBB}}^{\text{bi}}}{\partial \sigma_{\text{I}}}= \begin{cases}\operatorname{sign}(\boldsymbol{w})\left(1-\sigma_{\text{II}} \operatorname{sign}(\boldsymbol{w})\right), & \text{if } \sigma_{\text{I}} \operatorname{sign}(\boldsymbol{w}) \in(\boldsymbol{w}-1, \boldsymbol{w}+1), \\ \operatorname{sign}(\boldsymbol{w}), & \text{otherwise,}\end{cases} \tag{7}$$

$$\frac{\partial \boldsymbol{w}_{\text{EBB}}^{\text{bi}}}{\partial \sigma_{\text{II}}}=\operatorname{sign}\left(\boldsymbol{w}-\sigma_1 \operatorname{sign}(\boldsymbol{w})\right), \tag{8}$$

where the Straight Through Estimator (STE) is applied to approximate the sign function during backwards. With the binary basis with different learnable scalars, the representation capability of quantized weights can be significantly enhanced. The residual initialization makes the optimization of binarized DM start from an error-minimizing state. With EBB, the representation of weight is significantly diversified compared to the binarized DM baseline, where the statistic about the EBB is presented in Fig **??**.

Surrender Strategy. We adopted a two-stage training process with a regularization strategy, allowing the DM to transition from an initial multi-basis structure to full binarization. In the first stage, regularization loss is applied to the higher-order learnable scaling factors, encouraging them to approach zero:

$$\mathcal{L}_{\mathrm{EBB}} = \mu \frac{1}{N} \sum_{i=1}^{N} \sigma_{\mathrm{II}}^{i}. \tag{9}$$

Where N denote the number of basic layers (e.g., convolutional, linear) in the noise estimation network of DMs, and μ are hyperparameter coefficients used to balance the loss terms.

In the second stage, all higher-order terms are removed, and the forward propagation is simplified to:

$$\boldsymbol{w}^{\mathrm{bi}} = \sigma_{\mathrm{I}} \operatorname{sign}(\boldsymbol{w}). \tag{10}$$

Location Selection. In our BinaryDM, the proposed EBB is partially applied to crucial and parameter-sparse locations of the diffusion models while retaining concise vanilla binarization at other locations to reduce unnecessary evolution processes and the associated training overhead. Specifically, we apply EBB where the feature scale is greater or equal to $\frac{1}{2}$ input scale, *i.e.*, the first and last six layers with only the 15% of whole parameters in the noise estimation network of BinaryDM. In contrast, other layers keep consistent with the binarized DM baseline with vanilla binarizers. On the one hand, applying EBB to these key parameter locations within DM architectures significantly enhances the information processing capacity of binarized DMs in the early stages of optimization, leading to a better overall learning process. On the other hand, using a vanilla binarizer for intermediate layers, which contain the most parameters but are less sensitive to quantization loss, reduces the instability caused by switching between stages for unimportant components and lowers the training overhead.

2.3 Low-rank Representation Mimicking

In the quantization-aware training of DMs, the discretization of parameter space caused by weight binarization and activation quantization function and the inaccurate gradient approximation involved in the derivation process bring difficulties to the stable convergence of binarized DM. Since having almost the same architecture, the original full-precision DM can be regarded as an oracle of the binarized one. Therefore, an intuitive approach is to assist the training of binarized DMs by mimicking the representation of full-precision replicas. During training, aligning outputs and/or intermediate representations of binarized DMs with full-precision counterparts can provide additional supervision, accelerating the convergence of quantized DMs significantly.

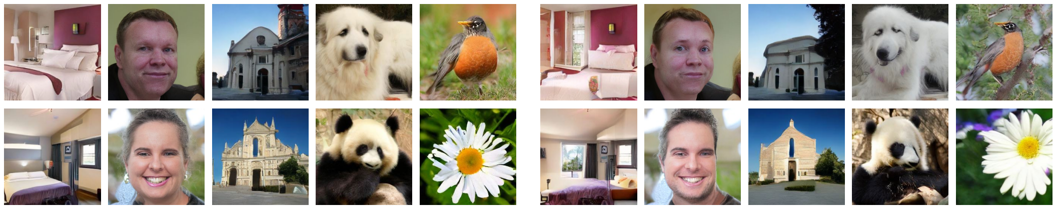
However, there are issues directly aligning the intermediate representations of binarized and full-precision DMs during optimization. Firstly, fine-grained alignment of high-dimensional representation leads to a blurry optimization direction for DMs, especially when mimicking the intermediate features is introduced. Secondly, compared to the full-precision DM, the intermediate features in the binarized one are derived from a discrete latent space since the discretization of parameters makes it difficult to mimic the full-precision DM directly.

Therefore, we propose Low-rank Representation Mimicking (LRM) to efficiently optimize the BinaryDM by mimicking full-precision representations in a low-rank space. We group the full-precision DM θ^{FP} based on the timestep embedding modules composed of residual convolution and transformer blocks. The intermediate representation can be denoted as $\hat{\boldsymbol{\epsilon}}^{\mathrm{FP}}_{\theta_i}(\boldsymbol{x}_t, t) \in \mathbb{R}^{h\times w\times c}$. We use principal component analysis (PCA) to project representations to low-rank space. The covariance matrix for representations of the full-precision DM is

$$C_i = \frac{1}{(h\times w)^2} \hat{\boldsymbol{\epsilon}}^{\mathrm{FP}}_{\theta_i}(\boldsymbol{x}_t, t)\, \hat{\boldsymbol{\epsilon}}^{\mathrm{FP}T}_{\theta_i}(\boldsymbol{x}_t, t), \tag{11}$$

where θ_i represents the composition of the top i modules. The eigenvector matrix $E_i \in \mathbb{R}^{c\times c}$ is

$${E_i}^T C_i E_i = \Lambda_i, \tag{12}$$



BinaryDM Baseline

Figure 2: Visualization of samples generated by the binarized DM baseline and W1A4 BinaryDM.

where Λ_i is the diagonal matrix of eigenvalues of C_i, arranged in descending order. We take the matrix composed of the first $\lceil \frac{c}{K} \rfloor$ column eigenvectors of E_i as the transformation matrix, denoted as $E_i^{\lceil \frac{c}{K} \rfloor}$, where $\lceil \cdot \rfloor$ denotes the round function and K denotes to the reduction times of dimension. We use $E_i^{\lceil \frac{c}{K} \rfloor}$ to project the intermediate representation of both full-precision and binarized:

$$\boldsymbol{\mathcal{R}}_i^{\text{FP}}\left(\boldsymbol{x}_t, t\right) = \hat{\boldsymbol{\epsilon}}_{\theta_i}^{\text{FP}}\left(\boldsymbol{x}_t, t\right) E_i^{\lceil \frac{c}{K} \rfloor}, \quad \boldsymbol{\mathcal{R}}_i^{\text{bi}}\left(\boldsymbol{x}_t, t\right) = \hat{\boldsymbol{\epsilon}}_{\theta_i^{\text{bi}}}^{\text{bi}}\left(\boldsymbol{x}_t, t\right) E_i^{\lceil \frac{c}{K} \rfloor}, \tag{13}$$

where $\hat{\boldsymbol{\epsilon}}_{\theta_i}^{\text{bi}}\left(\boldsymbol{x}_t, t\right)$ denotes the intermediate representation of the i-th layer in the DM with binarized parameters θ^{bi}, and $\boldsymbol{\mathcal{R}}_i^{\text{FP}}\left(\boldsymbol{x}_t, t\right)$ and $\boldsymbol{\mathcal{R}}_i^{\text{bi}}\left(\boldsymbol{x}_t, t\right)$ denote the low-rank representations of full-precision and binarized DMs, respectively, with the same shape $h \times w \times \lceil \frac{c}{K} \rfloor$. The K empirically defaults as 4 and is detailed ablated in Appendix **??**.

We then leverage the obtained low-rank representation to drive the binarized DM to learn the full-precision counterpart. We construct a mean squared error (MSE) loss between the i-th module of low-rank representations between full-precision and binarized DMs:

$$\mathcal{L}_{\text{LRM}i} = \left\| \boldsymbol{\mathcal{R}}_i^{\text{FP}} - \boldsymbol{\mathcal{R}}_i^{\text{bi}} \right\|. \tag{14}$$

The total loss function is composed of Eq. (4), Eq. (9) and Eq. (14):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{simple}} + \mathcal{L}_{\text{EBB}} + \lambda \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{\text{LRM}i}, \tag{15}$$

where M denotes the number of timestep embedding modules in the noise estimation network of DMs, and λ is a hyperparameter coefficient to balance the loss terms.

Since the computation cost of obtaining the transformation matrix $E_i^{\lceil \frac{c}{K} \rfloor}$ in LRM is significantly expensive, we compute the matrix by the first batch of input and keep it fixed during the training process. The fixed mapping between representations is also beneficial to the optimization of binarized DM from a steady perspective.

LRM enables binarized DMs to mimic the representation of full-precision counterparts, improving the optimization process by introducing additional supervision. As shown in Fig **??**, LRM effectively brings the local block closer to the full-precision block. Furthermore, by applying low-rank projections based on the principal components from full-precision representations before representation mimicking, the binarized DM can be optimized along clear and stable directions, accelerating the convergence of the model. Furthermore, binarized and full-precision DMs have completely consistent architectures, making representation mimicking between them natural.

3 Experiment

Settings. We conduct experiments on LSUN-Bedrooms 256×256 [36] for unconditional image generation tasks over LDM-4. The evaluation metrics used in our study encompass Fréchet Inception Distance (FID) [9], Sliding Fréchet Inception Distance (sFID) [26], and Precision-and-Recall. We implement and evaluate the DMs binarized by our BinaryDM and the baseline presented in Section 2.1, where LSQ [4] is employed uniformly as activations quantizers. Several SOTA quantization methods for DMs with 2~8 bits weights are also considered [8, 15].

Main Results. Our LDM experiments encompass the evaluation of LDM-4 on LSUN-Bedrooms. We showcase results across various activation bit widths in the context of weight binarization, comparing

Table 1: Results for LDM on multiple datasets in unconditional generation by DDIM with 100 steps.

Model	Dataset	Method	#Bits	Size$_{\text{(MB)}}$	FID↓	sFID↓	Precision↑	Recall↑
LDM-4	LSUN-Bedrooms 256 × 256	FP	32/32	1045.4	3.09	7.08	65.82	45.36
		LSQ	2/32	69.8	7.49	12.79	64.02	**37.60**
		Baseline	1/32	35.8	8.43	13.11	65.45	29.88
		BinaryDM	1/32	35.8	**6.99**	**12.15**	**67.51**	36.80
		Q-Diffusion	2/8	69.8	62.01	33.56	16.48	14.12
		Baseline	1/8	35.8	9.37	12.10	64.36	30.76
		BinaryDM	1/8	35.8	**6.51**	**11.67**	**65.80**	**35.28**
		Q-Diffusion	4/4	134.9	427.46	277.22	0.00	0.00
		EfficientDM	4/4	134.9	10.60	-	-	-
		LSQ	2/4	69.8	12.95	12.79	55.97	**34.30**
		Q-DM	1/4	35.8	9.99	11.96	57.62	29.30
		TDQ	1/4	35.8	11.28	12.80	55.14	27.32
		Baseline	1/4	35.8	10.87	15.46	64.05	26.50
		BinaryDM	1/4	35.8	**7.74**	**10.80**	**64.71**	32.98

Table 2: Ablation results on LSUN-Bedrooms 256 × 256.

Method	#Bits	FID↓	sFID↓	Prec.↑	Recall↑
FP	32/32	3.09	7.08	65.82	45.36
Vanilla	1/32	8.43	13.11	65.45	29.88
+EBB	1/32	7.39	12.34	65.98	35.84
+LRM	1/32	**6.99**	**12.15**	**67.51**	**36.80**

them with the outcomes of some quantization methods at higher bit settings. The conventional binary baseline method exhibits subpar performance in the LDM context and experiences a further decline in the W1A4 experimental setup. In contrast, BinaryDM significantly enhances the generation quality, exhibiting consistent performance across different activation bit settings. Notably, when compressing from W1A32 to W1A4, the FID increased by a mere 0.75 for BinaryDM, showcasing its robustness.

Ablation Study. We evaluate the effectiveness of our proposed EBB and LRM, and the results are presented in Table 2. The performance has shown significant recovery when applying our EBB only to binarized DM. With the application of LRM on this basis, the generative capability of the resulting binarized diffusion models is further enhanced, with the FID decreasing to 6.99.

Efficiency Analysis. The results in Table 3 indicate that our DM can achieve up to 29.2× space savings while obtaining up to 15.2× acceleration during inference.

Table 3: Inference efficiency of our proposed BinaryDM of LDM-4 on LSUN-Bedrooms 256 × 256.

Model	Method	#Bits	Size$_{\text{(MB)}}$	OPs$_{(\times 10^9)}$	FID↓
LDM-4	Full-Precision	4/4	1045.4	96.0	3.09
	Q-Diffusion	4/4	134.9	24.3	427.46
	EfficientDM	4/4	134.9	24.3	10.60
	LSQ	2/4	69.8	12.3	12.95
	BinaryDM	1/4	**35.8**	**6.3**	**7.74**

Limitations. BinaryDM directly uses layerwise LSQ [4] for activations instead of specific designs, we thus believe the potential for improving BinaryDM from activation quantization perspective.

4 Conclusion

In this paper, we propose BinaryDM, a novel accurate quantization-aware training approach to push the weights of diffusion models towards the limit of binary. Firstly, we present an Evolvable-Basis Binarizer (EBB) to enable the QAT of binarized DMs to start from a more favorable initial state, leading to a smoother optimization process and better final results. Secondly, a Low-rank Representation Mimicking (LRM) is applied to enhance the binarization-aware optimization of the DM, alleviating the optimization direction ambiguity caused by fine-grained alignment. Comprehensive experiments demonstrate that BinaryDM achieves significant accuracy and efficiency gains compared to SOTA quantization methods of DMs under ultra-low bit-widths. As the first binarization method for diffusion models, W1A4 BinaryDM achieves impressive 15.2× OPs and 29.2× storage savings, showcasing substantial advantages and potential for deploying DMs on edge.

References

[1] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

[2] Hong Chen, Chengtao Lv, Liang Ding, Haotong Qin, Xiabin Zhou, Yifu Ding, Xuebo Liu, Min Zhang, Jinyang Guo, Xianglong Liu, et al. Db-llm: Accurate dual-binarization for efficient llms. *arXiv preprint arXiv:2402.11960*, 2024.

[3] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, pages 1–11, 2016.

[4] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *International Conference on Learning Representations*, pages 1–12, 2019.

[5] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. *arXiv preprint arXiv:2305.10924*, 2023.

[6] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[8] Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Efficientdm: Efficient quantization-aware fine-tuning of low-bit diffusion models. *arXiv preprint arXiv:2310.03270*, 2023.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[12] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*, 2024.

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in Neural Information Processing Systems*, 29:1–9, 2016.

[14] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. *arXiv preprint arXiv:2104.01409*, 2021.

[15] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023.

[16] Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[17] Zefan Li, Bingbing Ni, Wenjun Zhang, Xiaokang Yang, and Wen Gao. Performance guaranteed network acceleration via high-order residual quantization. In *Proceedings of the IEEE international conference on computer vision*, pages 2584–2592, 2017.

[18] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *Proceedings of the European Conference on Computer Vision*, pages 143–159. Springer, 2020.

[19] Weijian Luo. A comprehensive survey on knowledge distillation of diffusion models. *arXiv preprint arXiv:2304.04262*, 2023.

[20] Kangfu Mei and Vishal Patel. Vidm: Video implicit diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9117–9125, 2023.

[21] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

[22] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021.

[23] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[24] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021.

[25] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[27] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[28] Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021.

[29] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1972–1981, 2023.

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[33] Yixing Xu, Kai Han, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Learning frequency domain approximation for binary neural networks. *Advances in Neural Information Processing Systems*, 34:25553–25565, 2021.

[34] Zihan Xu, Mingbao Lin, Jianzhuang Liu, Jie Chen, Ling Shao, Yue Gao, Yonghong Tian, and Rongrong Ji. Recu: Reviving the dead weights in binary neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5198–5208, 2021.

[35] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7316, 2019.

[36] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We make the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work in the section 3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data used is open source, and the code will be included in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the experimental setting in section 3 and section **??**.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We fixed random seeds for all experiments to ensure the reproducibility of the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources in the section **??** and the section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper introduces the original owners of assets in the section 3 and the section **??**.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will include the anonymous code in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.