KOALA: Empirical Lessons Toward Memory-Efficient and Fast Diffusion Models for Text-to-Image Synthesis

Youngwan Lee[1,2] Kwanyong Park[1] Yoorhim Cho[3] Yong-Ju Lee[1] Sung Ju Hwang[2,4]

[1]Electronics and Telecommunications Research Institute (ETRI), South Korea
[2]Korea Advanced Institute of Science and Technology (KAIST), South Korea
[3]Sungkyunkwan University, South Korea
[4]DeepAuto.ai, South Korea
project page: https://youngwanlee.github.io/KOALA/

Abstract

As text-to-image (T2I) synthesis models increase in size, they demand higher inference costs due to the need for more expensive GPUs with larger memory, which makes it challenging to reproduce these models in addition to the restricted access to training datasets. Our study aims to reduce these inference costs and explores how far the generative capabilities of T2I models can be extended using only publicly available datasets and open-source models. To this end, by using the de facto standard text-to-image model, Stable Diffusion XL (SDXL), we present three key practices in building an efficient T2I model: (1) **Knowledge distillation**: we explore how to effectively distill the generation capability of SDXL into an efficient U-Net and find that self-attention is the most crucial part. (2) **Data**: despite fewer samples, high-resolution images with rich captions are more crucial than a larger number of low-resolution images with short captions. (3) **Teacher**: Step-distilled Teacher allows T2I models to reduce the noising steps. Based on these findings, we build two types of efficient text-to-image models, called KOALA-Turbo &-Lightning, with two compact U-Nets (1B & 700M), reducing the model size up to 54% and 69% of the SDXL U-Net. In particular, the KOALA-Lightning-700M is 4× faster than SDXL while still maintaining satisfactory generation quality. Moreover, unlike SDXL, our KOALA models can generate 1024px high-resolution images on consumer-grade GPUs with 8GB of VRAMs (3060Ti). We believe that our KOALA models will have a significant practical impact, serving as cost-effective alternatives to SDXL for academic researchers and general users in resource-constrained environments.

1 Introduction

Since Stable Diffusion XL [31] (SDXL) has become the de facto standard model for text-to-image (T2I) synthesis due to its ability to generate high-resolution images and its open-source nature, many models for specific downstream tasks [7; 47; 3; 58; 56] now leverage SDXL as their backbone. However, the model's massive computational demands and large size necessitate expensive hardware, thus incurring significant costs in terms of training and inference. Moreover, recent T2I works [5; 6; 39; 1] do not release their training datasets, making it challenging for the open-source community to reproduce their performance due to internal or commercial restrictions.

To alleviate the computation burden, previous works have resorted to quantization [45], hardware-aware optimization [8], denoising step reduction [38; 29; 39; 23], and architectural model optimization [21]. In particular, the denoising step reduction (*i.e.*, step-distillation) and architectural model compression methods adopt the knowledge distillation (KD) scheme [14; 11] by allowing the model

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

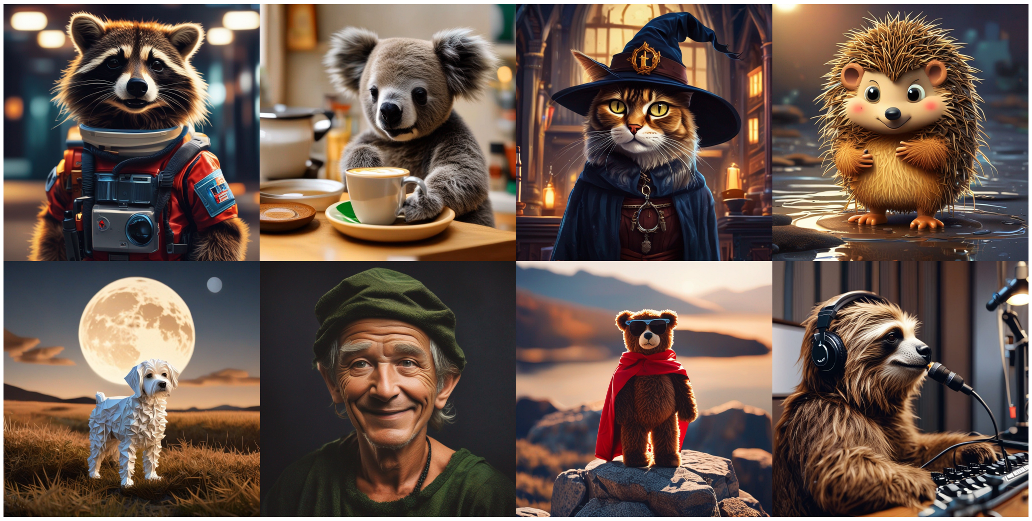


Figure 1: **Samples by KOALA-Lightning-700M** with $\mathbf{1024^2}$ resolution and 10 denoising steps, generated in 0.65 seconds on NVIDIA 4090 GPU. The prompts and more qualitative comparisons are illustrated in App. C.

to mimic the output of the SDMs as a teacher model. For the architectural model compression, BK-SDM [21] exploits KD when compressing the most heavy-weight part, U-Net, in SDM-v1.4 [35]. BK-SDM builds a compressed U-Net by simply removing some blocks, which allows the compressed U-Net to mimic the last features at each stage and the predicted noise from the teacher model during the pre-training phase. However, the compression method proposed by BK-SDM achieves a limited compression rate (33% in Tab. 2) when applied to the larger SDXL than SDM-v1.4, and the strategy for feature distillation in U-Net has *not yet been fully explored*.

In this work, our goal is to build an efficient T2I model based on **kno**wledge distill**a**tion in the **la**tent diffusion model (KOALA) by exploring how far we can push the performance using only open-source data and models alone. To this end, we first design two compressed U-Nets, KOALA-1B and KOALA-700M, using not only block removal but also *layer-wise removal* to reduce the model size of the SDXL U-Net by up to 54% and 69% (vs. BK's method: 33%) in Tab. 2. Then, we explore how to enhance the generative capability of the compact U-Net based on three empirical findings; (1) **self-attention based knowledge distillation**: we investigate how to effectively distill SDXL as a teacher model and find *essential factors* for feature-level KD. Specifically, we found that self-attention features are the most crucial for distillation since self-attention-based KD allows models to learn more discriminative representations between objects or attributes. (2) **Data**: When performing KD-training, high-resolution images and longer captions are more critical, even with fewer samples, than a larger number of low-resolution images with short captions. (3) **Teacher model**: The teacher model with higher performance improves the capability of the student model, and step-distilled teachers allow the student model to reduce the denoising steps, which results in further speed-up.

Based on these findings, we train efficient text-to-image synthesis models on *publicly available* LAION-POP [40] by using two types of distilled teacher models, SDXL-Turbo [39] and SDXL-Lightning [23], with 512px and 1024px resolutions, respectively. We observe that our KD method consistently outperforms the BK [21] method in both U-Net and Diffusion Transformer backbones in Tabs. 7 and 8. In addition, KOALA-Lightning-700M outperforms SSD-1B [10], which is trained using the BK method, at 3× faster speed. Furthermore, KOALA-Lightning-700M achieves 4× faster speed and 3× model efficiency than SDXL-Base while exhibiting satisfactory generation quality. Lastly, to validate its practical impact, we perform inference analysis on a variety of ***consumer-grade*** **GPUs** with different memory sizes (*e.g.*, 8GB, 11GB, and 24GB), and the results show that SDXL models cannot be mounted on an 8GB GPU, whereas our KOALA models can operate on it, demonstrating that our KOALA models are cost-effective alternatives for practitioners[1] in resource-constrained environments.

[1] https://civitai.com/

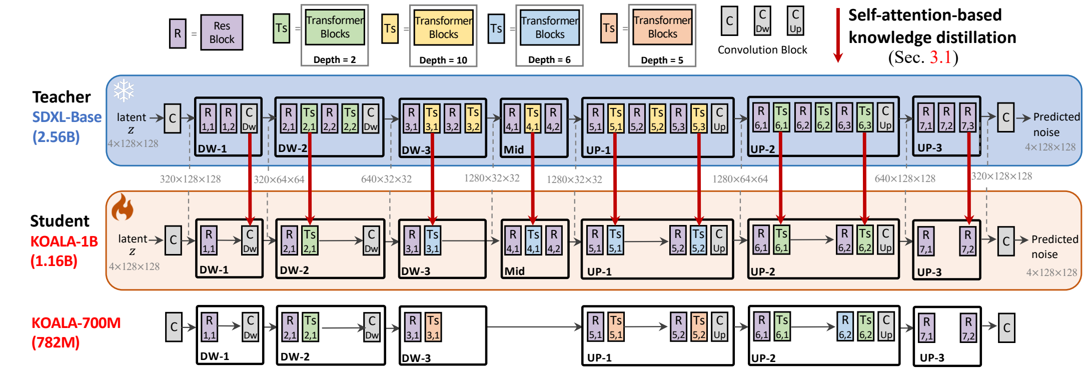


Figure 2: **Overview of KnOwledge-DistillAtion in LAtent diffusion model based on SDXL and architecture of KOALA.** We omit skip connections for simplicity. We perform feature distillation in transformer blocks using self-attention layers.

Our main contributions are as follows:

1. We design two efficient denoising U-Net with model sizes (1.13B/782M) that are more than twice as compact SDXL's U-Net (2.56B).
2. We perform a comprehensive analysis to build efficient T2I models, drawing three key lessons: self-attention distillation, the impact of data characteristics, and the influence of the teacher model.
3. We conduct a systematic analysis of inference on consumer-grade GPUs, highlighting that our KOALA models operate efficiently on an 8 GB GPU, unlike other state-of-the-art models.

2 In-depth Analysis: Stable Diffusion XL

Table 1: **SDXL-Base-1.0 model budget.** Latency is measured under the image scale of 1024^2, FP16-precision, and 25 denoising steps in NVIDIA 4090 GPU (24GB).

SDXL	Text Enc.	VAE Dec.	U-Net
#Param.	817M	83M	2,567M
Latency (s)	0.008	0.002	3.133

Table 2: **U-Net Comparison.** Tx means Transformer. SDM-v2.0 [36] uses 768^2 resolution, while SDXL and KOALA models use 1024^2 resolution. CKPT means the trained checkpoint file.

U-Net	SDM-v2.0	SDXL-1.0	BK-SDXL	**KOALA-1B**	**KOALA-700M**
#Param.	865M	2,567M	1,717M	1,161M	782M
CKPT size	3.46GB	10.3GB	6.8GB	4.4GB	3.0GB
Tx blocks	[1, 1, 1, 1]	[0, 2, 10]	[0, 2, 10]	[0, 2, 6]	[0, 2, 5]
Mid block	✓	✓	✓	✓	✗
Latency	1.13s	3.13s	2.42s	1.60s	1.25s

SDXL [31], the latest version of the SDM series [35; 36; 34], exerts a significant influence on both the academic community and the industry due to its unprecedented 1024^2 high-resolution image quality and open source resources. It has several key improvement points from the previous SDM-v2.0 [36], *e.g.*, multiple sizes- & crop-conditioning, an improved VAE, a much larger U-Net, and an ad hoc style of refinement module, which leads to significantly improved generation quality. However, the significant enlargement of U-Net in model size results in increased computational costs and significant memory (or storage) requirements, hampering the accessibility of SDXL. Thus, we investigate the U-Net in SDXL to design a more lightweight U-Net for knowledge distillation. We dissect the components of SDXL, quantifying its size and latency during the denoising phase, as detailed in Tab. 1. The enlarged U-Net (2.56B) is the primary cause of the increasing SDXL model size (vs. SDM-v2.0 (865 M)). Furthermore, the latency of U-Net is the main inference time bottleneck in SDXL. Therefore, it is necessary to reduce U-Net's model budget for better efficiency.

The SDXL's U-Net varies in the number of transformer blocks for each stage, unlike SDM-v2.0, which employs a transformer block for each stage (see Tab. 2). At the highest feature levels (*e.g.*, `DW-1&UP-3` in Fig. 2), SDXL uses only residual blocks without transformer blocks, instead distributing more transformer blocks to lower-level features. So, in Fig. 13, we analyze the parameter distribution of each stage in the U-Net. Most parameters (83%) are concentrated on the transformers with ten blocks in the lowest feature map (*e.g.*, 32^2 of `DW-3`, `Mid`, `UP-1` in Fig. 2), making the main parameter bottleneck. Thus, it is essential to address this bottleneck when designing an efficient U-Net architecture.

3 Three lessons for building an efficient text-to-image model

In this section, we introduce three empirical lessons to realize an efficient text-to-image synthesis; first, we design a lightweight U-Net architecture and perform a comprehensive analysis with the proposed efficient U-Net to find knowledge-distillation (KD) strategies in Sec. 3.1.2. Secondly, we investigate the training data characteristics that affect image generation quality in Sec. 3.2. Finally, we explore how different teacher models influence the student model in Sec. 3.3. For these empirical studies, we adopt two evaluation metrics, Human Preference Score (HPSv2) [55] and CompBench [17], instead of FID. Recently, several works [2; 55; 31] have claimed that FID [13] is not closely correlated with visual fidelity. HPSv2 [55] is for a visual aesthetics metric, which allows us to evaluate visual quality in terms of more specific types. As an image-text alignment metric, Compbench [17] is a more comprehensive benchmark for evaluating the compositional text-to-image generation capability than the single CLIP score [12]. We report average scores for HPS and Compbench, respectively.

3.1 Lesson. 1: Self-attention based Knowledge Distillation with Efficient U-Net Architecture

3.1.1 Efficient U-Net architecture

A prior strategy for compressing the text-to-image model is to remove a pair of residual and transformer blocks at each stage, namely block removal [21; 10]. While this method may be sufficient for compressing shallow U-Nets like in SDM-v1.4 [35], it shows limited effectiveness for recent, more complex U-Nets. For cases of SDXL [31], the compression rate is reduced only from 2.5B to 1.7B, as shown in Tab. 2. To address this limitation, we investigate a new approach for compressing these heavier U-Nets to achieve more efficient text-to-image generation.

Transformer layer-wise removal is the core of efficient U-Net architecture design. According to the discussion in Sec. 2, the majority of parameters are concentrated in the transformer blocks at the lowest feature levels. Each block comprises multiple consecutive transformer layers, specifically ten layers per block in SDXL (see Fig. 2). We address this computational bottleneck by reducing the number of transformer layers (i.e., depth), a strategy we term *layer-wise removal*.

Using this removal strategy as the core, we instantiate two compressed U-Net variants: KOALA-1B and KOALA-700M. First, we apply the prior block-removal strategy [21] to the heavy U-Net of SDXL. We note that in the decoder part (*e.g.*, `UP-1` to `UP-3`), we remain more blocks than in the encoder because the decoder part plays a more important role in knowledge distillation, which is addressed in Sec. 3.1.2 and Tab. 3b. On this block-removed backbone, we then adopt *layer-wise removal* at different ratios. Specifically, we reduce the transformer layers at the lowest feature level (*i.e.*, `DW-3`, `Mid` and `UP-1` in Fig. 2) from 10 to 5 for KOALA-700M and to 6 for KOALA-1B. For KOALA-700M, we also removed the `Mid` block. An overview of the compressed U-Nets is presented in Tab. 2 and Fig. 2. Our KOALA-1B model has 1.16B parameters, making it half the size of SDXL (2.56B). KOALA-700M, with 782M parameters, is comparable in size to SDM-v2.0 (865M).

3.1.2 Exploring Knowledge Distillation for SDXL

Table 3: **Analysis of feature level knowledge distillation of U-Net in SDXL [31].**

Type	HPSv2
Baseline	25.53
SA	**26.74**
CA	26.11
Res	26.27
FFN	26.48
LF	26.63

(a) **Distillation type**

Loc.	HPSv2
Baseline	25.53
DW-2	25.32
DW-3	25.57
Mid	25.66
UP-1	**26.52**
UP-2	26.05

(b) **Distill stage**

Prior work [21] that attempts to distill an early series of stable diffusion (*i.e.*, SDM-v1.4 [35]) directly follows traditional knowledge distillation literature [37; 11]. The compressed student U-Net model S_θ is jointly trained to learn the target task and mimic the pre-trained U-Net of SDM-v1.4 as a teacher network. Here, the target task is the reverse denoising process [16], and we denote the corresponding learning signal as $\mathcal{L}_{\text{task}}$. Besides the task loss, the compressed student model is trained to match the output of the pre-trained U-Net at both output and feature levels. $\mathcal{L}_{\text{out}}$ and $\mathcal{L}_{\text{feat}}$ represent the knowledge distillation (KD) loss at the output- and feature-level, respectively. For designing the feature-level KD-loss, BK-SDM [21] simply considers only the last feature (`LF`) map of the teacher $f_T^i(\cdot)$ and

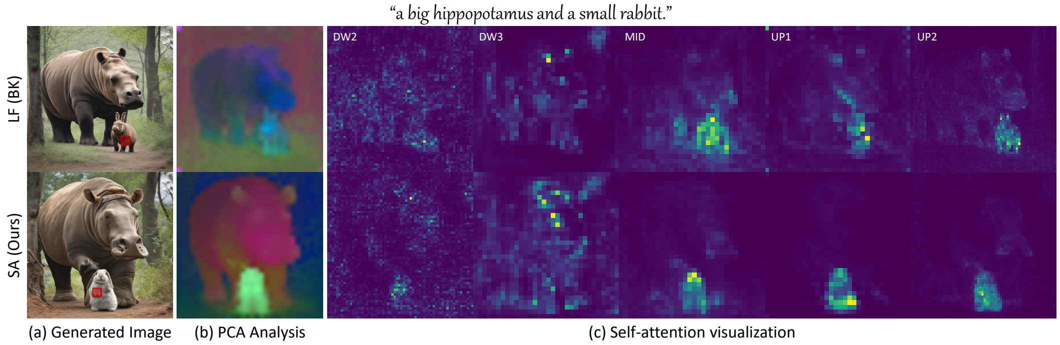


Figure 3: **Analysis on self-attention maps of distilled student U-Nets.** (a) Generated images of LF- and SA-based distilled models, which are BK-SDM [21] and our proposal, respectively. In BK-SDM's result, a rabbit is depicted like a hippopotamus (*i.e.*, appearance leakage). (b) Visualization of PCA analysis results on self-attention maps of `UP-1` stage. (c) Representative visualization of self-attention map from different U-Net stages. Red boxes denote the query patches. Note that from the `MID` stage, the `SA`-based model ***attends*** to the rabbit more ***discriminatively*** than the `LF` model, demonstrating that self-attention-based KD allows to generate objects more distinctly.

student network $f_S^i(\cdot)$ at each stage as follows:

$$\mathcal{L}_{\text{featKD}} = \min_{S_\theta} \mathbb{E}_{z,\epsilon,c,t} || \sum_i f_T^i(z_t,t,c) - f_S^i(z_t,t,c)||_2^2, \tag{1}$$

where t and c denote given diffusion timestep and text embeddings as conditions. Thus, the feature distillation approach for text-to-image diffusion models has ***not been sufficiently explored***, leaving room for further investigation.

In this section, we extensively explore feature distillation strategies to distill the knowledge from the U-Net of SDXL effectively to our efficient U-Net, KOALA-1B. We start from a baseline trained only by $\mathcal{L}_{\text{task}}$ and add $\mathcal{L}_{\text{featKD}}$ without $\mathcal{L}_{\text{outKD}}$ to validate the effect of feature distillation. More training details are described in Sec. 4.1 and App. A.

Self-attention-based knowledge distillation transfers discriminative image representation.

With the increasing complexity of U-Net and its stage, relying solely on the last feature (`LF`) as in BK [21] may not be sufficient to mimic the intricate behavior of the teacher U-Net. Thus, we revisit which features provide the richest guidance for effective knowledge distillation. We focus on key intermediate features from each stage: outputs from the self-attention (`SA`), cross-attention (`CA`), and feedforward net (`FFN`) in the transformer block, as well as outputs from convolutional residual block (`Res`) and `LF`. Tab. 3a summarizes the experimental results. While all types of features help obtain higher performance over the naïve baseline with only the task loss, distilling ***self-attention features*** achieves the most performance gain. Considering the prior studies [22; 46; 49] which suggest that `SA` plays a vital role in capturing semantic affinities and the overall structure of images, the results emphasize that such information is crucial for the distillation process.
To understand the effects more clearly, we illustrate a representative example in the Fig. 3. To reason about how the distilled student U-Net captures self-similarity, we perform a PCA analysis [19; 50] on self-attention maps. Specifically, we apply PCA on self-attention maps from `SA`- and `LF`-based models and show the top three principal components in Fig. 3-(b). Interestingly, in the `SA`-based model, each principal component distinctly represents individual objects (*i.e.*, unique color assignments to each object). This indicates that the `SA`-based model effectively distinguishes different objects in modeling self-similarity, which plays a crucial role in accurately rendering the distinct appearance of each object. In contrast, the `LF`-based model exhibits less distinction between objects, resulting in *appearance leakage* between them (*e.g.*, a small hippo with rabbit ears). More PCA analyses are detailed in Fig. 10.

Self-attention at the decoder has a larger impact on the quality of generated images.

We further explore the role and significance of each self-attention stage. To this end, we first visualize the self-attention map in Fig. 3-(c). The self-attention maps initially capture general contextual

Table 4: **Training Data** comparison. AR and ACL mean average resolution and average caption length, respectively. synCap means synthetic captions by LLaVA-v1.5 [24]

ID	Data	#Imgs	AR	ACL	HPSv2	CompBench
(a)	LAION-A-6+ [43]	8M	580 × 676	13	27.43	0.3791
(b)	(a) + synCap	8M	580 × 676	72	27.61	0.4168
(c)	LAION-POP [40]	491K	1274 × 1457	81	**27.79**	**0.4290**

Table 5: **Teacher model** comparison. We use KOALA-700M as a student model.

Teacher model	Step	HPSv2	CompBench
SDXL-Base-1.0	25	27.79	0.4290
SDXL-Turbo	10	27.88	0.4470
SDXL-Lightning	10	**28.13**	**0.4538**

information (*e.g.*, DW-2&DW-3) and gradually focus on localized semantics (*e.g.*, MID). In the decoder, self-attentions increasingly correlate with higher-level semantics (*e.g.*, object) to accurately model appearances and structures. Notably, in this stage, the SA-based model attends corresponding object regions (given the query patch, red box) more *discriminatively* than the LF-based model, which results in improved compositional image generation performance.

In addition, we ablate the significance of each self-attention stage in the distillation process. Specifically, we adopt an SA-based loss at a single stage alongside the task loss. As shown in Tab. 3b, the results align with the above understanding: distilling self-attention knowledge within the ***decoder*** stages significantly enhances generation quality. In comparison, the impact of self-attention solely within the encoder stages is less pronounced. Consequently, we opt to retain more SA layers within the decoder (see Fig. 2).

In summary, we train our efficient KOALA U-Nets using the following objectives: $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{outKD}} + \mathcal{L}_{\text{featKD}}$. We apply our proposed self-attention-based knowledge distillation (KD) methods to $\mathcal{L}_{\text{featKD}}$. Further analyses on featKD, including how to locate features and combine different types of features, are provided in App. B.1.

3.2 Lesson 2. Data: the impact of sample size, image resolution, and caption length

We investigate various data factors—such as image resolution, caption length, and the number of samples—that impact the quality of the final text-to-image model. To ensure reproducibility, we design three data variants using open-source data. (i) LAION-Aesthetic-6+ (LAION-A-6+) [43] includes a large volume of image-text pairs (8,483,623) with images filtered for high aesthetics. Most images are low-resolution (average 580 × 676), and the corresponding captions are brief (average length of 13 words). (ii) Description-augmented LAION-A-6+ is designed to demonstrate the impact of detailed descriptions. For each image in LAION-A-6+, we use a large multimodal model [24] (LMM) to generate detailed descriptions. These synthesized captions, referred to as synCap, convey significantly more semantic information and are longer (e.g., an average length of 72 words; more details on synCap in App. A.1). This data source is denoted in the second row of the table. (iii) LAION-POP [40] features high-resolution images (average 1274 × 1457) and descriptive captions (average length of 81 words), although the dataset size is relatively small (e.g., 491,567 samples). The descriptions are generated by LMM, CogVML [54] and LLaVA-v1.5.

We train KOALA-700M models using the same training recipes for each data source. From the results summarized in Tab. 4, we make several observations. First, detailed captions significantly boost performance, enabling the model to learn detailed correspondences between images and text (See (a) and (b) in the Compbench score). Second, high-resolution images, which convey complex image structures, are a valuable source for training T2I models. Despite having fewer samples, LAION-POP further boosts overall performance. Based on these findings, we opt to use LAION-POP as the main training data, as it features high-resolution images and descriptive captions.

3.3 Lesson 3. The influence of Teacher model

Following the tremendous success of SDXL [31], recent large-scale text-to-image models have adopted its U-Net backbone. SDXL-Turbo [39] and SDXL-Lightning [23] are notable examples, enabling high-quality image generation in low-step regime through progressive distillation [38]. This section investigates whether our distillation framework can effectively exploit these diverse models. To this end, we leverage SDXL-Base and its variants, SDXL-Turbo and SDXL-Lightning, as teacher models, transferring their knowledge into KOALA-700M. We apply the former two lessons and more training details are described in App. A.2.

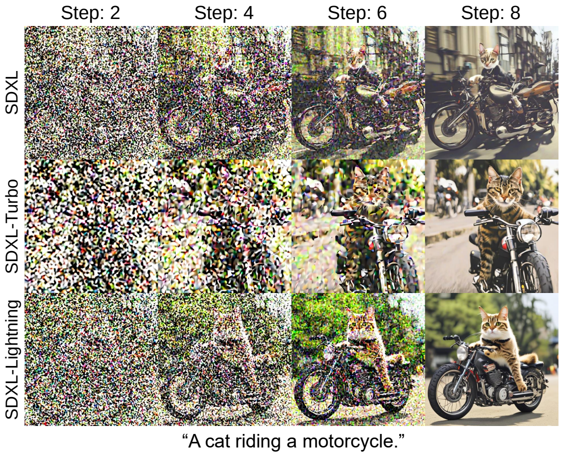


Figure 5: Denoising process of different teacher models.

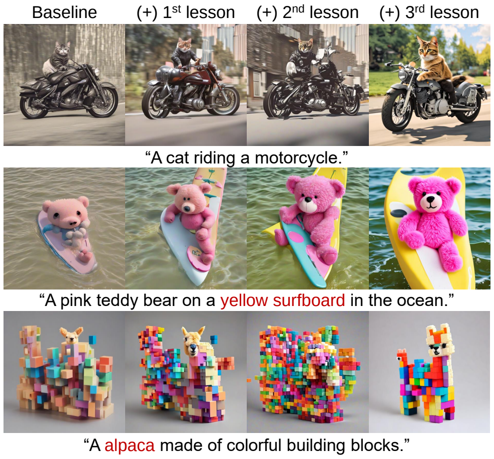

Baseline
(+) 1st lesson
(+) 2nd lesson
(+) 3rd lesson
"A cat riding a motorcycle."
"A pink teddy bear on a yellow surfboard in the ocean."
"A alpaca made of colorful building blocks."

Figure 6: Qualitative analysis on proposed lessons.

As shown in Tab. 5, all KOALA-700M models distilled from different teacher models demonstrate decent image generation capabilities. This highlights the generality of our knowledge distillation framework. More interestingly, when using SDXL-Turbo and SDXL-Lightning as teachers, KOALA-700M models exhibit comparable or even better image quality than when SDXL-Base is used as the teacher, despite requiring fewer denoising steps (*e.g.*, 10 vs. 25). Note that the KD framework or noise schedule (Euler discrete schedule [20], the same as SDXL-Base) for the different KOALA models does not require specific modifications. Thus, KOALA models seamlessly inherit the ability to illustrate realistic structures and details in images, even in the short-step regime, from their step-distilled teachers (See 2^{nd} and 3^{rd} rows in Fig. 5). This results in robust performance across a diverse range of denoising steps (See Fig. 4). In contrast, when SDXL is used as the teacher model, KOALA models struggle to depict realistic structures in a few steps, leading to flawed features (e.g., missing handles and lights on a motorcycle in Fig. 5). For more efficient text-to-image synthesis, we leverage step-distilled teachers, enabling KOALA models to generate high-quality images in just a few steps.

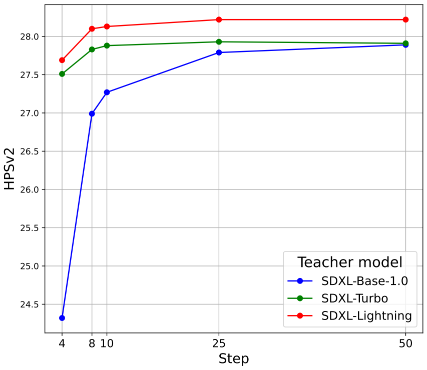


Figure 4: Teacher model comparisons across denoising steps with KOALA-700M.

Therefore, combining all these lessons, we build **K**n**O**wledge-distill**A**tion-based **LA**tent diffusion models, called KOALA. As discussed in each section, the three proposed lessons complement each other. Both image quality and image-text alignment improve progressively as each lesson is added, as shown in Fig. 6. In particular, we build two types of KOALA models: KOALA-Turbo with 512px and KOALA-Lightning with 1024px using two KOALA U-Net (1B&700M), respectively.

4 Experiments

4.1 Implementation details

Dataset. As the datasets used for state-of-the-art methods in Tab. 6 are proprietary or not released, we opt for the *publicly accessible* only LAION dataset [44] to ensure the reproducibility of our work. From the data recipe in Sec. 3.2, we finally use LAION-POP [40] for KOALA models in Tab. 6. More details of the dataset we used are described in App. A.1

Training. According to the recipe about the Teacher model in Sec. 3.3, we use SDXL-Turbo [39] and SDXL-Lightning [23] as teacher models, building two types of KOALA models, KOALA-Turbo and KOALA-Lightning. Since SDXL-Turbo and -Lightning are based on SDXL-Base model [31], we use the same two text encoders, OpenCLIP ViT-bigG [18] and CLIP ViT-L [33] and only replace the

Table 6: **Performance comparison to state-of-the-art models**. We measure latency and memory usage with a bath size of 1 on NVIDIA 4090 GPU. We obtain HPSv2 and Compbench scores of all models on the same GPU and library environment by using their official weights. We highlight the best value in green, and the second-best value in blue. The full scores of HPSv2 and Compbench are shown in Tab. 10.

Model	Resolution	Steps	Latency (s)	U-Net Param.	Memory	HPSv2	CompBench
SDM-v2.0 [36]	768^2	25	1.236	0.86B	5.6GB	25.86	0.3672
SDXL-Base-1.0 [31]	1024^2	25	3.229	2.56B	11.9GB	30.82	0.4445
SDXL-Turbo [39]	512^2	8	0.245	2.56B	8.5GB	29.93	0.4489
SDXL-Lightning [23]	1024^2	8	0.719	2.56B	11.7GB	32.18	0.4445
Pixart-α [5]	1024^2	25	3.722	0.6B	17.3GB	32.06	0.3880
Pixart-Σ [6]	1024^2	25	3.976	0.6B	17.3GB	31.75	0.4612
SSD-1B [10]	1024^2	25	2.094	1.3B	9.4GB	31.43	0.4497
SSD-Vega [10]	1024^2	25	1.490	0.74B	8.2GB	32.17	0.4461
KOALA-Turbo-700M	512^2	10	0.194	0.78B	4.9GB	29.98	0.4555
KOALA-Turbo-1B	512^2	10	0.238	1.16B	5.7GB	29.84	0.4560
KOALA-Lightning-700M	1024^2	10	0.655	0.78B	8.3GB	31.50	0.4505
KOALA-Lightning-1B	1024^2	10	0.790	1.16B	9.1GB	31.71	0.4590

Table 7: **Comparison to BK [21]**. All models are trained for 50K iterations same as BK-SDM.

KD method	Backbone	HPSv2	CompBench
BK [21]	BK-Small	26.72	0.3237
Ours	BK-Small	**26.86**	**0.3417**
BK [21]	KOALA-1B	27.01	0.3599
Ours	KOALA-1B	**27.15**	**0.3712**

Table 8: **KD feature types in Diffusion Transformer (Pixart-Σ [6])**.

KD Type.	HPSv2	CompBench
SA	**25.16**	**0.4281**
CA	24.94	0.4279
FFN	24.80	0.4191
LF in BK [21]	21.62	0.3527

original U-Net with our efficient KOALA U-Net. Our U-Nets are initialized with the teacher's U-Net weights at the exact block location. Using our self-attention-based KD method in Sec. 3.1.2, we train our KOALA models on the LAION-POP dataset using four NVIDIA A100 (80GB) GPUs with 512^2 and 1024^2 resolutions for KOALA-Turbo and KOALA-Lightning, respectively. **Inference.** We use Euler discrete scheduler [20] as the same sampler in SDXL [31]. All KOALA models generate images with 10 denoising steps, FP16, and cfg-sale [15] of 3.5. Please see further details of training and inference in App. A.2 and App. A.3, respectively.

4.2 Main results

vs. SDXL models: Compared to SDXL-Base-1.0 [31], our KOALA-Lightning-700M/-1B models achieve better performance in terms of HPSv2 and CompBench while showing about 5× and 4× faster speed, respectively. Compared to SDXL-Turbo [39] and SDXL-Lightning [23], our KOALA-Turbo and KOALA-Lightning models show comparable or inferior HPSv2 scores but achieve higher CompBench scores with up to 3× smaller model sizes and 1.7× lower memory usage. **vs. Pixart:** KOALA-Lightning models fall short in HPSv2 and CompBench. Especially, Pixart-Σ [6] achieves the best CompBench and the second-best HPSv2 scores. This result is attributed to data quality, as Pixart-Σ collects high-quality internal data consisting of 33 million images above 1K resolution and uses synthetic longer captions (with an average length of 184 words). However, our KOALA-Lightning-700M shows 6× faster speed and 2× better memory efficiency. **vs. SSD:** Note that due to the difference in training datasets, we cannot make a direct comparison with SSD models, which are trained by BK [21]'s KD method. Except for HPSv2 of SSD-Vega [10], KOALA-Lighting models show better HPSv2 and CompBench scores while achieving up to 3× faster speed. More qualitative comparisons in App. C support the quantitative results.

4.3 Discussion

Comparison with BK-SDM. For a fair comparison to BK-SDM [21], we train our KOALA U-Net backbones with their distillation method under the same data setup (See training details in A.2). As shown in Tab. 7, our KD method consistently achieves higher HPSv2 and CompBench scores than the BK-SDM [21] when using different U-Net backbones. These results demonstrate two main implications as follows: 1) the proposed distillation of the self-attention layer is more helpful for

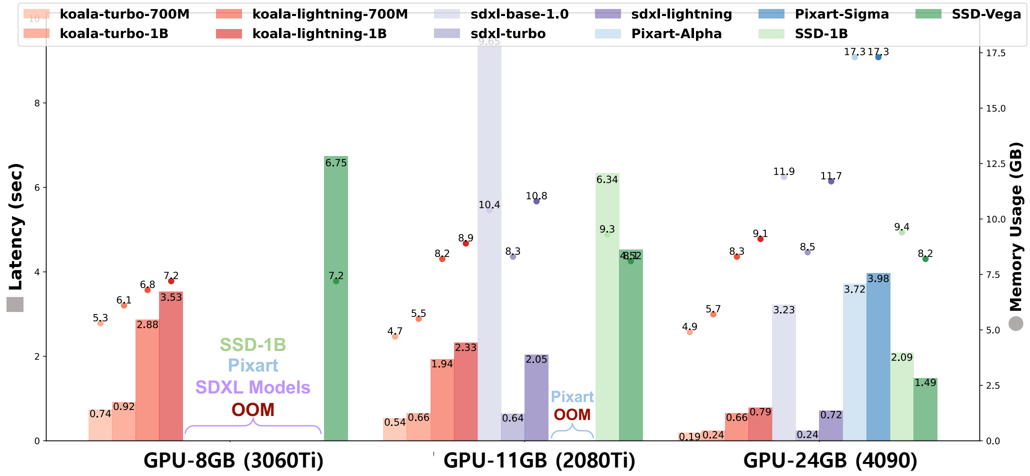


Figure 7: **Latency and Memory** comparison on across different ***consumer-grade*** **GPUs**. We run each model with the denoising steps in Tab. 6 and FP16. For a fair comparison, we use the official pre-trained weights and inference code in the Hugginface without any other tricks such as `torch.compile` or quantization. Note that **only our KOALA models and SSD-Vega can run all types of GPUs**.

Table 9: **Synergy effect with a step-distilled method**, PCM [53]. We conduct step-distillation training using PCM with our KOALA backbones and compare with PCM-SDXL-Base.

Method	Teacher	Student	#Step	Param. (B)	Memory (GB)	Latency	HPS	CompBench
PCM [53]	SDXL-Base	SDXL-Base	2	2.56	12.1	0.345	**29.99**	**0.4169**
PCM [53]	SDXL-Base	**KOALA-700M**	2	**0.78**	**8.2**	**0.222**	28.78	0.3930
PCM [53]	SDXL-Base	**KOALA-1B**	2	1.16	9.0	0.235	29.04	0.4055

visual aesthetics than simply distilling the last layer feature by BK [21]. 2) our self-attention-based KD approach allows the model to learn more discriminative representations between objects or attributes so that it can follow prompts faithfully (as shown in Sec. 3.1.2 and Fig. 3). More qualitative comparisons are demonstrated in Fig. 19.

Applicability of self-attention based KD to Diffusion Transformer. To validate the generality of our self-attention distillation method, we also apply it to a diffusion transformer (DiT) based T2I model, Pixart-Σ [6]. To this end, We compress the DiT-XL [30] backbone and build DiT-M by reducing the number of layers from 28 to 14 with the same hidden dimension (see more training details in App. A.4.). Following our KD strategy, we conduct an ablation study by simply changing the distillation location due to DiT's architectural simplicity, which consists of only transformer blocks without resolution changes. Tab. 8 illustrates that distilling the self-attention feature outperforms other features while using the last features proposed in BK [21] shows the worst performance, demonstrating that the self-attention layer is still the most crucial part for diffusion transformer.

Synergy Effect with Step-Distillation Method, PCM [53]. Since step-distillation methods [23; 39; 53] and our KD approach are orthogonal, applying our KOALA backbones to the step-distillation methods could yield synergistic effects, leading to further speed improvements. To verify the synergy effect between the step-distillation method (*e.g.*, PCM [53]) and our KOALA backbones, we conduct step-distillation training using PCM with our KOALA backbones, and the results are presented in Tab. 9. Thanks to their efficiency, our KOALA backbones allow PCM[2,3] to achieve additional speed-up with only a slight performance drop compared to using the SDXL-Base backbone. Furthermore, we provide qualitative comparisons between PCM-KOALA models and PCM-SDXL-Base in Fig. 20, demonstrating that the generated images achieve visual quality comparable to those of SDXL-Base.

[2] PCM is the only work that officially provides step-distillation training code

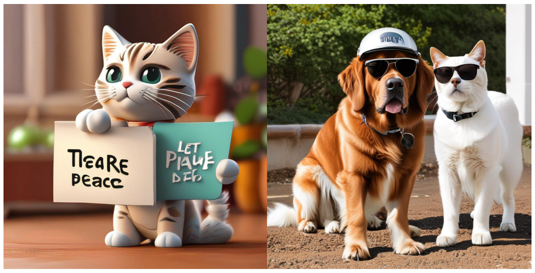
[3] `https://github.com/G-U-N/Phased-Consistency-Model`

4.4 Model budget comparison on consumer-grade GPUs

We further validate the efficiency of our model by measuring its inference speed and memory usage on a variety of ***consumer-grade* GPUs** with different memory sizes, such as 8GB (3060Ti), 11GB (2080Ti), and 24GB (4090), because the GPU environment varies for individual practitioners. On the GPU-8GB, all SDXL models can't fit, while only KOALA models and SSD-Vega [10] can run. KOALA-Lightning-700M consumes comparable GPU memory but shows 2× faster than SSD-Vega. On the GPU-11GB, SDXL models can run, but KOALA-Lightning-700M still runs at approximately 5× faster speed than SDXL-Base [31]. It is noted that Pixart-α [5] & $-\Sigma$ [6] cannot operate on GPUs with 8GB and 11GB of memory due to their higher memory usage, but they can run on a GPU-24GB, albeit at the slowest speed. It is worth noting that our KOALA models can operate *efficiently* on all types of GPUs, highlighting the versatility and accessibility of our approach. Furthermore, our KOALA-Lighting-700M is the best alternative for high-resolution image generation that can replace SDXL models in resource-constrained GPU environments.

5 Limitations

While our KOALA models generate images with decent aesthetic quality, such as photo-realistic or 3D-art renderings, they still show limitations in synthesizing legible texts in the generated image as shown in Fig. 8 (Left). Also, our models have difficulty in generating complex prompts with multiple attributed or object relationships, as shown in Fig. 8 (Right). Additionally, since SDXL is the de facto T2I model, we have tried to compress the SDXL U-Net by addressing its bottlenecks. However, this approach is somewhat specific to the SDXL U-Net and heuristic. This limitation arises because the SDXL U-Net has a complex and heterogeneous architecture, comprising both convolutional and transformer blocks, which hinders the formulation of a more general compression principle. More detailed investigations and examples are described in App. D.



A 3d art character of a cute kitty holding a sign that says "Let there be peace"

A brown big dog wearing sunglasses standing on the left and a white small cat wearing helmet sitting on the right

Figure 8: **Failure cases of KOALA-700M**

6 Conclusion

In this work, we have explored how to build memory-efficient and fast T2I models, designing compact denoising U-Nets and presenting three critical lessons for boosting the performance of the efficient T2I models: 1) the importance of self-attention in knowledge distillation, 2) data characteristics, and 3) the influence of teacher models. Thanks to these empirical insights, our KOALA-Lightning-700M model substantially reduces the model size (69%↓) and the latency (79%↓) of SDXL-Base while exhibiting satisfactory generation quality. We hope that our KOALA models can serve as cost-effective alternatives for practitioners in limited GPU environments and that our lessons benefit the open-source community in their attempts to improve the efficiency of T2I models.

Additionally, since we have identified the potential of applying our self-attention-based KD to Diffusion Transformer (DiT) models [30; 5; 6] in Tab. 8 due to their architectural simplicity compared to the U-Net in SDXL, we plan to further explore more general model compression methods for DiT, as in the language model literature [9; 28], and KD techniques based on our self-attention distillation.

7 Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00187238, Development of Large Korean Language Model Technology for Efficient Pre-training, 45%), (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 45%) and (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%).

References

[1] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., and Ramesh, A. Improving image generation with better captions. https://cdn.openai.com/papers/dall-e-3.pdf, 2023. 1, 15

[2] Betzalel, E., Penso, C., Navon, A., and Fetaya, E. A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935*, 2022. 4

[3] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1

[4] Bohan, O. B. Sdxl-vae-fp16-fix. https://huggingface.co/madebyollin/sdxl-vae-fp16-fix, 2023. 16, 17

[5] Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 1, 8, 10, 15, 16

[6] Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., and Li, Z. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. 1, 8, 9, 10, 16, 17

[7] Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., and Wei, F. Textdiffuser: Diffusion models as text painters. *NeurIPS*, 2024. 1

[8] Chen, Y.-H., Sarokin, R., Lee, J., Tang, J., Chang, C.-L., Kulik, A., and Grundmann, M. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *CVPR-Workshop*, 2023. 1

[9] Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P., and Roberts, D. A. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024. 10

[10] Gupta, Y., Jaddipal, V. V., Prabhala, H., Paul, S., and Platen, P. V. Progressive knowledge distillation of stable diffusion xl using layer level loss, 2024. 2, 4, 8, 10, 16, 21, 26

[11] Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., and Choi, J. Y. A comprehensive overhaul of feature distillation. In *ICCV*, 2019. 1, 4

[12] Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 4

[13] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 4

[14] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[15] Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 8, 17

[16] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 4, 16, 17

[17] Huang, K., Sun, K., Xie, E., Li, Z., and Liu, X. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS*, 2023. 4, 16

[18] Ilharco, G., Wortsman, M., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., and Schmidt, L. Openclip. https://github.com/mlfoundations/open_clip, 2021. URL https://doi.org/10.5281/zenodo.5143773. 7, 16

[19] Jolliffe, I. T. and Cadima, J. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. 5, 19

[20] Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 7, 8, 17

[21] Kim, B.-K., Song, H.-K., Castells, T., and Choi, S. On architectural compression of text-to-image diffusion models. *arXiv preprint arXiv:2305.15798*, 2023. 1, 2, 4, 5, 8, 9, 16, 17, 18, 21

[22] Kolkin, N., Salavon, J., and Shakhnarovich, G. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, 2019. 5

[23] Lin, S., Wang, A., and Yang, X. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 1, 2, 6, 7, 8, 9, 16, 21, 26

[24] Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning, 2023. 6, 15

[25] Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning, 2023. 15

[26] Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 16

[27] Luo, Y., Ren, X., Zheng, Z., Jiang, Z., Jiang, X., and You, Y. Came: Confidence-guided adaptive memory efficient optimization. *arXiv preprint arXiv:2307.02047*, 2023. 17

[28] Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., and Chen, W. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024. 10

[29] Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. In *CVPR*, 2023. 1

[30] Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *ICCV*, 2023. 9, 10, 17

[31] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 3, 4, 6, 7, 8, 10, 16, 17, 19, 21, 26

[32] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Stabilityai: Sdxl-base-1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2023. 17

[33] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 7, 16

[34] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[35] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. Stable-diffusion-v1.4. https://github.com/CompVis/stable-diffusion, 2022. 2, 3, 4, 16

[36] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. Stable-diffusion-v2.0. https://github.com/Stability-AI/stablediffusion, 2022. 3, 8, 16

[37] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 4

[38] Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 1, 6

[39] Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. Adversarial diffusion distillation, 2023. 1, 2, 6, 7, 8, 9, 16, 21, 27

[40] Schuhmann, C. and Bevan, P. Laion pop: 600,000 high-resolution images with detailed descriptions. https://huggingface.co/datasets/laion/laion-pop, 2023. 2, 6, 7, 15, 17, 21

[41] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-aestheics v2. https://laion.ai/blog/laion-aesthetics/, 2022. 15

[42] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-aesthetics v2 6.5+. https://huggingface.co/datasets/ChristophSchuhmann/improved_aesthetics_6.5plus, 2022. 16

[43] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-aesthetics v2 6+. https://huggingface.co/datasets/ChristophSchuhmann/improved_aesthetics_6plus, 2022. 6, 15, 16

[44] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 7

[45] Shang, Y., Yuan, Z., Xie, B., Wu, B., and Yan, Y. Post-training quantization on diffusion models. In *CVPR*, 2023. 1

[46] Shechtman, E. and Irani, M. Matching local self-similarities across images and videos. In *CVPR*, 2007. 5

[47] Shi, Y., Wang, P., Ye, J., Long, M., Li, K., and Yang, X. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 1

[48] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 17

[49] Tumanyan, N., Bar-Tal, O., Bagon, S., and Dekel, T. Splicing vit features for semantic appearance transfer. In *CVPR*, 2022. 5

[50] Tumanyan, N., Geyer, M., Bagon, S., and Dekel, T. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 5

[51] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 16

[52] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers/blob/main/examples/text_to_image/train_text_to_image_sdxl.py, 2023. 16

[53] Wang, F.-Y., Huang, Z., Bergman, A. W., Shen, D., Gao, P., Lingelbach, M., Sun, K., Bian, W., Song, G., Liu, Y., Li, H., and Wang, X. Phased consistency model. In *NeurIPS*, 2024. 9

[54] Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., Xu, J., Xu, B., Li, J., Dong, Y., Ding, M., and Tang, J. Cogvlm: Visual expert for pretrained language models, 2023. 6, 15

[55] Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., and Li, H. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 4, 16, 19

[56] Yang, L., Yu, Z., Meng, C., Xu, M., Ermon, S., and Cui, B. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. *arXiv preprint arXiv:2401.11708*, 2024. 1

[57] Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 15

[58] Zhang, S., Wang, J., Zhang, Y., Zhao, K., Yuan, H., Qin, Z., Wang, X., Zhao, D., and Zhou, J. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 1

Appendix

Contents

A Implementation details

A.1 Data

LAION-Aesthetics V2 6+ [41; 43] includes some imperfections; thus, we conduct careful data preprocessing. We first filtered out trivial imperfections such as blank text and corrupted images, resulting in 8,483,623 image-text pairs. Despite this, we observed that the text prompts in the LAION dataset are notably brief, which could hinder learning the accurate image-text correspondence.

Instruction:
“Describe the image and its style in a very detailed manner.”

LLaVA:
“The image is a panoramic view of a city skyline with the Statue of Liberty in the foreground. The statue is holding a torch and is surrounded by a body of water…”

LAION caption: “Why Visit New York.?”

LLaVA caption (Pixart-α’s style): “The image is a panoramic view of a city skyline with the Statue of Liberty in the foreground. The statue is holding a torch and is surrounded by a body of water. The city skyline is filled with skyscrapers, and there are several boats visible in the water. The scene is captured during sunset, giving it a warm and serene atmosphere.”

LLaVA SynCap (Ours): “Why Visit New york.?, The image is a panoramic view of a city skyline with the Statue of Liberty in the foreground. The statue is holding a torch and is surrounded by a body of water. The city skyline is filled with skyscrapers, and there are several boats visible in the water. The scene is captured during sunset, giving it a warm and serene atmosphere.”

Figure 9: **Sythesizing captions by LLaVA-1.5 [24].** We append the synthesized captions by LLaVA to the original ones, leveraging the existing contextual information such as proper nouns, *e.g.*, New York .

Synthesized captions, `synCap`**.** As some work [57; 1; 5] show that making caption data richer in information improves its generation quality, refining (cleaning) the training caption data and synthesizing more detailed captions corresponding to each paired image by a large multimodal model, *e.g.*, LLaVA [25; 24] would improve the image-text alignment capability. To do this, we utilize LLaVA-1.5 [24] to synthesize captions corresponding to images in LAION-Aesthetics V2 6+. As shown in Fig. 9, when we input the instruction to LLaVA to describe the details of the input image, we can get highly augmented captions. It is worth noting that contrary to Pixart-α [5], which replaces original captions with synthesized ones, our approach appends augmented captions to the original ones, leveraging the contextual richness of existing proper nouns (*e.g.*, New York).

LAION-POP [40] has a rather smaller number of images (*e.g.*, 491,567) but it has images with a higher resolution (1274 × 1457) and longer average prompt length (81) which is also generated by LMM models, CogVML [54] and LLaVA-v1.5. We can download the dataset in the Huggingface repository[4]. We train the main models, KOALA-Turbo and KOALA-Lightning, on LAION-POP dataset in Tab. 6 and Tab. 10.

[4] `https://huggingface.co/datasets/Ejafa/ye-pop`

Table 10: **Quantitative comparison to state-of-the-art models** with HPSv2 [55] (Left) for **visual aesthetics** and with T2I-CompBench [17] (Right) for **Image-text alignment**.

Model	#Param.	HPSv2					Attribute			Object Relationship		Complex	Average
	U-Net	Anime	Paintings	Photo	Concept-art	Average	Color	Shape	Texture	Spatial	Non-spatial		
SDM-v2.0 [36]	0.86B	26.34	25.41	26.46	25.24	25.86	0.5065	0.4221	0.4922	0.1342	0.3096	0.3386	0.3672
SDXL-Base-1.0 [31]	2.56B	32.50	30.98	29.02	30.76	30.82	0.6210	0.5451	0.5909	0.1971	0.3123	0.4005	0.4445
SDXL-Turbo [39]	2.56B	31.48	28.17	30.00	30.06	29.93	0.6531	0.5157	0.6181	0.1963	0.3133	0.3968	0.4489
SDXL-Lightning [23]	2.56B	33.6	30.23	32.42	32.48	32.18	0.6553	0.5106	0.5816	0.2133	0.3080	0.3984	0.4445
Pixart-alpha [5]	0.6B	33.45	30.80	32.07	31.93	32.06	0.4618	0.4565	0.5108	0.1923	0.3072	0.3991	0.3880
Pixart-sigma [6]	0.6B	33.13	30.64	31.64	31.59	31.75	0.6107	0.5463	0.6172	0.2538	0.3091	0.4302	0.4612
SSD-1B [10]	1.3B	32.90	31.78	28.87	32.18	31.43	0.6333	0.5313	0.5914	0.2139	0.3174	0.4108	0.4497
SSD-Vega [10]	0.74B	33.56	32.53	29.65	32.95	32.17	0.6445	0.5102	0.6064	0.2009	0.3129	0.4021	0.4461
KOALA-Turbo-700M	0.78B	31.03	28.57	30.11	30.20	29.98	0.6664	0.5137	0.6331	0.1844	0.3141	0.4216	0.4555
KOALA-Turbo-1B	1.16B	31.51	28.21	29.80	29.85	29.84	0.6571	0.5192	0.6284	0.1882	0.3148	0.4282	0.4560
KOALA-Lightning-700M	0.78B	32.26	30.09	31.76	31.87	31.50	0.6605	0.5179	0.5953	0.1969	0.3102	0.4223	0.4505
KOALA-Lightning-1B	1.16B	32.52	30.54	31.86	31.93	31.71	0.6706	0.5345	0.5940	0.2177	0.3114	0.4261	0.4590

A.2 Training

Common training protocol. First, we describe a common training protocol for all experiments in our work. We base our framework on the officially released SDXL-Base-1.0[5] and `Diffusers` library [51; 52]. We mainly replace computationally burdened SDXL's U-Net with our efficient U-Net. We keep the same two text encoders, OpenCLIP ViT-bigG [18] and CLIP ViT-L [33], used in SDXL. For VAE, we use `sdxl-vae-fp16-fix` [4], which enables us to use FP16 precision for VAE computation. We initialize the weights of our U-Net with the teacher's U-Net weights at the same block location. We freeze the text encoders, VAE, and the teacher U-Net of SDXL and only fine-tune our U-Net. When training, we use a discrete-time diffusion schedule [16], size- and crop-conditioning as in SDXL [31], AdamW optimizer [26], a batch size of 128, a constant learning rate of 10^{-5}, and FP16 precision.

For the ablation study on the knowledge distillation strategies in Tabs. 3a, 3b and 11, following the common training protocol except for batch size and training iteration, for fast verification, we train our KOALA models for 30k iterations with a batch size of 32 and 1024 × 1024 resolution on LAION-Aesthetics V2 6+ [43] dataset using one NVIDIA A100 (80GB) GPU.

For the ablation study in Lesson 2. Data in Tab. 4, following our common training protocol, we train all cases, *e.g.*, (a), (b), and (c) on each dataset (a) LAION-Aesthetics V2 6+ (b) LAION-Aesthetics-V2-6+ with `synCAP` and (c) LAION-POP with a batch size of 128 and 1024 × 1024 resolution for 100K iterations using 4 NVIDIA A100 (80GB) GPUs.

For the ablation study in Lesson 3. Teacher in Tab. 4, following our common training protocol, we train all cases on the LAION-POP dataset for 100K iterations using 4 NVIDIA A100 (80GB) GPUs. In particular, for using SDXL-Turbo as a Teacher model, SDXL-Turbo was originally trained with 512 × 512 resolution, so we perform KD-training using the SDXL-Turo teacher with 512 × 512 resolution. We use the officially released checkpoint[6] in Hugginface. In contrast, for using SDXL-Base and SDXL-Lightning as Teacher models, we follow their original papers with 1024 × 1024 resolution. For the SDXL-Lightning teacher model, we use the officially released 4-step unet-checkpoint[7] in Hugginface.

For the main results in Tab. 6, following our common training protocol, we finally train KOALA-Turbo and KOALA-Lightning equipped with two KOALA U-Net backbones with a batch size of 128 for 500K iterations on LAION-POP dataset using 4 NVIDIA A100 (80GB) GPUs.

For a fair comparison to our counterpart BK [21] in Tab. 7, we train SDM-Small proposed in BK-SDM [21] with our self-attention-based KD using SDM-v1.4 [35] as a Teacher model, following the BK-SDM training recipe for 50K iteration with a batch size of 256 on LAION-Aesthetics V2 6.5+ [42]. On the other hand, we train our KOALA-1B U-Net with the BK method and compare it with our KD method under the same training setup such as the same SDXL-Base-1.0 Teacher model, following the common training protocol except for 50K training iterations.

[5] `https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0`

[6] `https://huggingface.co/stabilityai/sdxl-turbo`

[7] `https://huggingface.co/ByteDance/SDXL-Lightning`

Table 11: **Analysis of feature level knowledge distillation of U-Net in SDXL [31].** SA, CA, and FFN denote self-attention, cross-attention, and feed-forward net in the transformer block. Res is a convolutional residual block and LF denotes the last feature (same in BK [21]). For the ablation study, we train our KOALA-1B as student U-Net for 30K iterations with a batch size of 32.

Distill type	HPSv2
SD-loss	25.53
SA	**26.74**
CA	26.11
Res	26.27
FFN	26.48
LF	26.63

(a) **Distillation type**

Distill loc.	HPSv2
SD-loss	25.53
DW-2	25.32
DW-3	25.57
Mid	25.66
UP-1	**26.52**
UP-2	26.05

(b) **Distill stage**

SA loc.	HPSv2
SA-bottom	**26.74**
SA-inter	26.58
SA-up	26.48

(c) **SA location.**

Combination	HPSv2
Baseline (SA only)	26.74
SA + LF at DW-1 & UP-3	**26.98**
SA + Res at DW-1 & UP-3	26.94
SA + LF all	26.83
SA + Res all	26.80
SA+CA+Res+FFN+LF all	26.39

(d) **Combination.**

A.3 Inference

When generating samples, we also generate images with 1024×1024 and 512×512 for KOALA-Lightning and KOALA-Turbo, FP16-precision and `sdxl-vae-fp16-fix` [4] for VAE-decoder. Note that in the SDXL original paper [31], authors used DDIM sampler [48] to generate samples in the figures while the diffuser's official SDXL code [32] used Euler discrete scheduler [20] as the default scheduler. Therefore, we also use the Euler discrete scheduler for generating samples. For KOALA-Lighting and KOALA-Turbo, we infer with 10 denoising steps. we set classifier-free guidance [15] to 3.5. For measuring latency and memory usage in fair conditions, we construct the same software environments across machines with different GPUs. Specifically, we use `Pytorch==v2.1.2` and for a fair comparison, we don't use any speed-up tricks such as `torch.compile` and quantization.

A.4 Knowledge Distillation for Diffusion Transformer

Following our KD strategies, we first compress Diffusion Transformer (DiT [30]) backbone, DiT-XL, in Pixart-Σ [6] by reducing the number of 28 transformers layers to 14 based on our finding in Tab. 11c, building DiT-M. Specifically, we select the bottom layers, *e.g.*, from 0 to 14-th layers and remove the upper layers, *e.g.*, from 14-th to 27-th layers. We maintain the same embedding dimension size of 1152 for DiT-M as in DiT-XL, resulting in a model size of approximately 313M for DiT-M (compared to 611M for DiT-XL). Then, we initialize the weights of DiT-M from the corresponding layers in DiT-XL. For training, we optimize the same objective as ours: $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{outKD}} + \mathcal{L}_{\text{featKD}}$. We conduct the ablation study in Tab. 8 by changing the feature location from the teacher model. Using the training recipe, codebase and `PixArt-Sigma-XL-2-512-MS` checkpoint[8] in Pixart-Σ, we train DiT-M with 512×512 resolution, a batch size of 192, multi-scale augmentation, CAME optimizer [27], a constant learning rate of $2e^{-5}$, and FP16 precision for 50 epochs on LAION-POP [40] dataset.

A.5 Detailed formulation of training objectives

We detail the two objectives, the $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{out}}$, which are omitted in the main paper. First, the target task loss $\mathcal{L}_{\text{task}}$ to learn reverse denoising process [16] is summarized as:

$$\mathcal{L}_{\text{task}} = \min_{S_\theta} \mathbb{E}_{z_t,\epsilon,t,c} ||\epsilon_t - \epsilon_{S_\theta}(z_t, t, c)||_2^2, \tag{2}$$

where ϵ_t is the ground-truth sampled Gaussian noise at timestep t, c is text embedding as a condition, and $\epsilon_{S_\theta}(\cdot)$ denotes the predicted noise from student U-Net model, respectively. Second, the output-level knowledge distillation (KD) loss is formulated as:

$$\mathcal{L}_{\text{outKD}} = \min_{S_\theta} \mathbb{E}_{z,\epsilon,t,c} ||\epsilon_{T_\theta}(z, t, c) - \epsilon_{S_\theta}(z, t, c)||_2^2, \tag{3}$$

where $\epsilon_{T_\theta}(\cdot)$ denotes the predicted noise from each U-Net in the teacher model.

[8] https://huggingface.co/PixArt-alpha/PixArt-Sigma-XL-2-512-MS

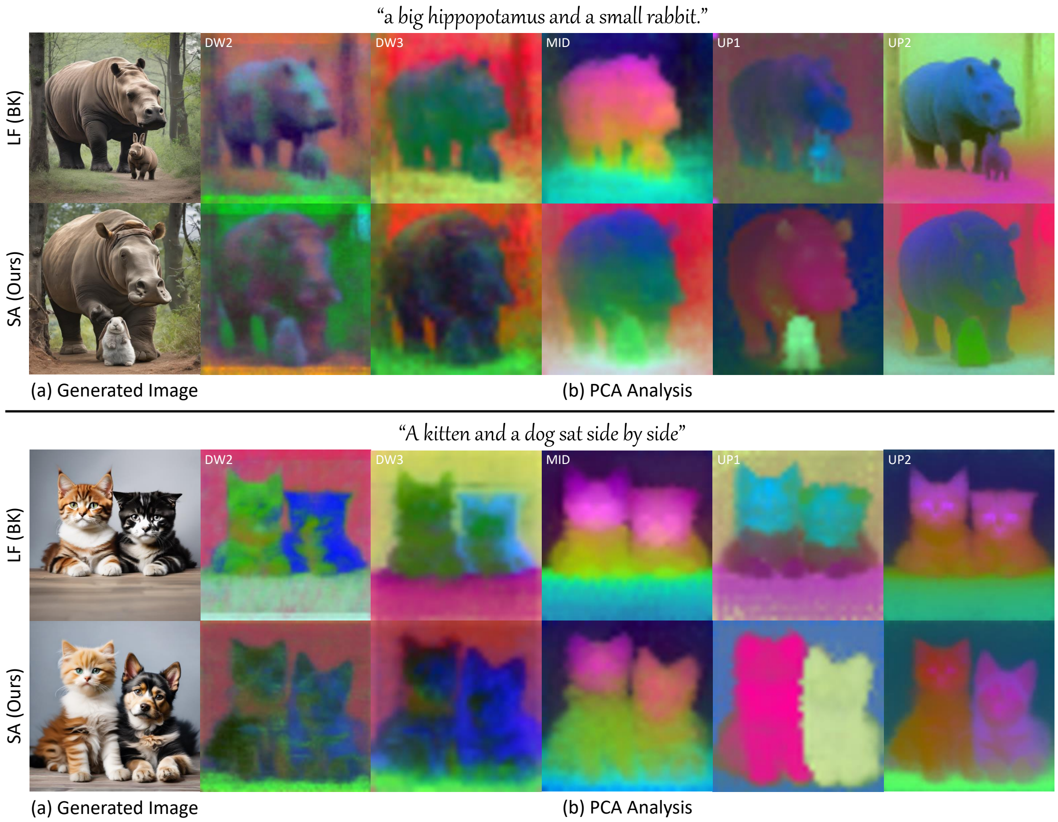


Figure 10: **Extended analysis on self-attention maps of distilled student U-Nets.** (a) Generated images of LF- and SA-based distilled models, which are BK-SDM [21] and our proposal, respectively. In BK-SDM's result, a rabbit or dog is depicted like a hippopotamus or cat, repectively (*i.e.*, appearance leakage). (b) Visualization of PCA analysis results. Note that from the `UP-1` stage, the `SA`-based model ***attends*** to the corresponding object (*i.e.*, rabbit or dog) more ***discriminatively*** than the `LF` model, demonstrating that self-attention-based KD allows to generate objects more distinctly.

B Additional Analysis

B.1 Self-attention based Feature-level Knowledge distillation

In this section, we further perform analyses for how to effectively distill feature information from the Teacher model.

Which SA's location is effective in the transformer blocks? At the lowest feature level, the depth of the transformer blocks is 6 for KOALA-1B, so we need to decide which locations to distill from the 10 transformer blocks of teacher U-Net. We assume three cases for each series of transformer blocks; (1) `SA-bottom`: $\{f_T^l \mid l \in \{1, 2, 3, 4, 5\}\}$, (2) `SA-interleave`: $\{f_T^l \mid l \in \{1, 3, 5, 7, 9, 10\}\}$, and (3) `SA-up`: $\{f_T^l \mid l \in \{6, 7, 8, 9, 10\}\}$ where l is the number of block. Tab. 11c shows that `SA-bottom` performs the best while `SA-up` performs the worst. This result suggests that the features of the early blocks are more significant for distillation. A more empirical analysis is described in App. B.3. Therefore, we adopt the `SA-bottom` strategy in all experiments.

Which combination is the best? In SDXL's U-Net, as shown in Fig. 2, there are no transformer blocks at the highest feature levels (*e.g.*, `DW-1&UP-3`); consequently, self-attention features cannot be distilled at this stage. Thus, we try two options: the residual block (`Res at DW-1&UP-3`) and the last feature (`LF at DW-1&UP-3`) as BK-SDM [21]. To this end, we perform SA-based feature distillation at every stage except for `DW-1` and `UP-3`, where we use the above two options, respectively. In addition, we try additional combinations: `SA+LF all`, `SA+Res all`, and `SA+CA+Res+FFN+LF all` where `all` means all stages. Tab. 11d demonstrates that adding more feature distillations

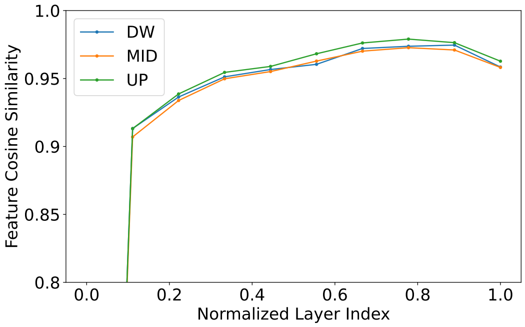


Figure 11: **Feature cosine similarity analysis.** We plot the cross-layer cosine similarity against the normalized layer indexes of transformer block.

to the SA-absent stage (*e.g.*, `DW-1&UP-3`) consistently boots performance, and especially `LF at DW1&UP3` shows the best. Interestingly, both `+LF all` and `+Res all` are worse than the ones at only `DW-1&UP-3` and `SA+CA+Res+FFN+LF all` is also not better, demonstrating that the SA features are not complementary to the other features.

B.2 Attention visualization for Tab. 11a and Tab. 11b

In Section 4.3 of the main paper, we provide empirical evidence demonstrating the paramount importance of self-attention features in the distillation process. Our findings particularly highlight the significant impact of specific self-attention (`SA`) stages (*e.g.*, `UP-1&UP-2`) on enhancing performance. To support these results, we extensively analyze self-attention maps in the main paper. To complete the analysis, we expand our Principal Component Analysis [19] (PCA) on self-attention maps to encompass all layers in Fig. 10.

As elaborated in the main paper, self-attention begins by capturing broad contextual information (*e.g.*, `DW-2&DW-3`) and then progressively attends to localized semantic details (*e.g.*, `MID`). Within the decoder, self-attentions are increasingly aligned with higher-level semantic elements `UP-1&UP-2`), such as objects, for facilitating a more accurate representation of appearances and structures. Notably, at this stage, the `SA`-based model focuses more on specific object regions than the `LF`-based model. This leads to a marked improvement in compositional image generation performance.

B.3 Feature cosine similarity analysis for Tab. 11c

KOALA models compress the computationally intensive transformer blocks in the lowest feature levels (*i.e.*, `DW-3&Mid&UP-1` stages). Specifically, we reduce the depth of these transformer blocks from 10 to 5 for KOALA-700M and to 6 for KOALA-1B. For this purpose, we demonstrate that distilling knowledge from the consecutive bottom layers of transformer blocks is a simple yet effective strategy (see third finding (F3) in the main paper).

To delve deeper into the rationale behind this strategy, we conducted a thorough feature analysis of the original SDXL model [31]. In particular, we investigate the evolution of the features within the transformer blocks. We compute the cross-layer cosine similarity between the output features of each block and those of its predecessors. A lower similarity score indicates a significant contribution of the current block, whereas a higher score implies a marginal contribution.

For this analysis, we leverage the diverse domain of prompts in the HPSv2 dataset [55]. We compute the cross-layer cosine similarity for each stage (`DW&Mid&UP`) and average these values across all prompts. The results are illustrated in Fig. 11. For all stages, transformer blocks exhibit a tendency of feature saturations: While early transformer blocks generally show a significant contribution, later blocks have less impact. This motivates us to distill the learned knowledge of consecutive bottom layers of transformer blocks for minimal performance degradation.

B.4 Implementation details of `SA-bottom`

Fig. 12 illustrates how to choose transformer blocks when distilling self-attention (SA) features at `DW3 & MID & UP1` as described in App. B.1 and Tab. 11c. In Fig. 12, the Transformer blocks (yellow) with

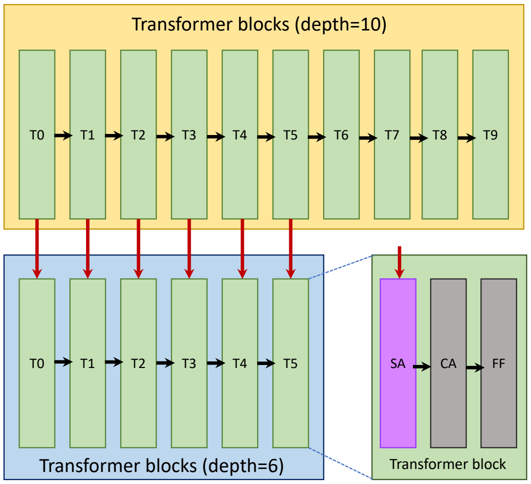


Figure 12: `SA-bottom` **illustration** in Tab. 11c.

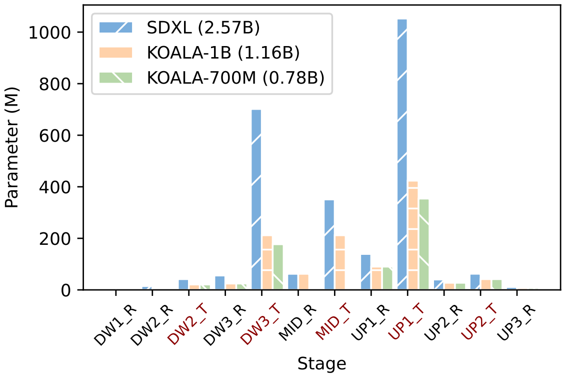


Figure 13: **Dissection of U-Net in SDXL.** DWi and UPi indicate i-*th* stage of the down and the up block, and R and T denote the Residual block and Transformer block, respectively.

a depth of 10 is from the original SDXL's U-Net teacher model, and the Transformer blocks (blue) with a depth of 6 is from our KOALA-1B's U-Net student model. For `SA-bottom` in Tab. 11c, we perform feature distillation by selecting consecutive blocks from the teacher model's transformer blocks, starting with the first one, and comparing to each transformer's self-attention (SA) features from the student model's transformer blocks.

C Qualitative results

C.1 Represenative prompts in Fig. 1

We use the following prompts for Fig. 1. From left-top to right-bottom:

- A 4k DSLR photo of a raccoon wearing an astronaut suit, photorealistic.
- A koala making latte art.
- A highly detailed zoomed-in digital painting of a cat dressed as a witch wearing a wizard hat in a haunted house, artstation.
- Cartoon of a cute hedgehog with tangled fur, standing character, looking surprised and awkward standing in a dirty puddle, dark circles under its eyes due to lack of sleep, depicted with comic exaggeration, spotlight effect highlighting its unkempt spines, use of vivid colors, high-definition digital rendering.

- A photorealistic render of an origami white and tan mini Bernadoodle dog standing in a surrealistic field under the moonlit setting.
- Peter Pan aged 60 years old, with a black background.
- A teddy bear wearing a sunglasses and cape is standing on the rock. DSLR photo.
- A photograph of a sloth wearing headphones and speaking into a high-end microphone in a recording studio.

More qualitative results are illustrated in Figs. 14 to 16.

C.2 Comparison to other methods

We compare our KOALA-Lightning with SDXL-Base-1.0 [31], SDXL-Lightning [23], SSD-1B [10] and SSD-Vega [10] using 1024 × 1024 resolution in Fig. 17. In addition, we compare our KOALA-Turbo with SDXL-Turo [39] using 512 × 512 resolution in Fig. 18.

C.3 Comparison to BK-SDM

In addition to the quantitative comparisons in the main paper, we also provide a qualitative comparison with BK-SDM [21]. As illustrated in Fig. 19, BK-SDM occasionally overlooks specific attributes or objects mentioned in the text prompt and generates structurally invalid images. On the contrary, our proposed model consistently generates images with enhanced adherence to the text, showcasing a superior ability to capture the intended details accurately.

D Failure cases

Fig. 21 illustrates that the KOALA-Lightning-1B model faces challenges in rendering legible text (the first row), accurately depicting human hands (the 2nd row), and complex compositional prompts with multiple attributes (the third row). We conjecture that these limitations may stem from the dataset, LAION-POP dataset [40], we used to train, whose images don't have enough of those styles.

Rendering long legible text. We have observed that the model has difficulty in synthesizing long-legible texts in the generated image. For example, as shown in Fig. 21 (1st-row), it renders unintended letters and sometimes doesn't generate correct characters. **Complex prompt with multiple attributes.** When attempting to compose an image using prompts that include various attributes of an object or scene, KOALA sometimes generates instances that do not perfectly follow the intended description. **human hands details.** While we have confirmed that the model excels at representing human faces, it still struggles to render human hands. This may be because we haven't learned enough about the structure of the hand itself, as human hands are more often seen in conjunction with other objects or situations than in isolation.

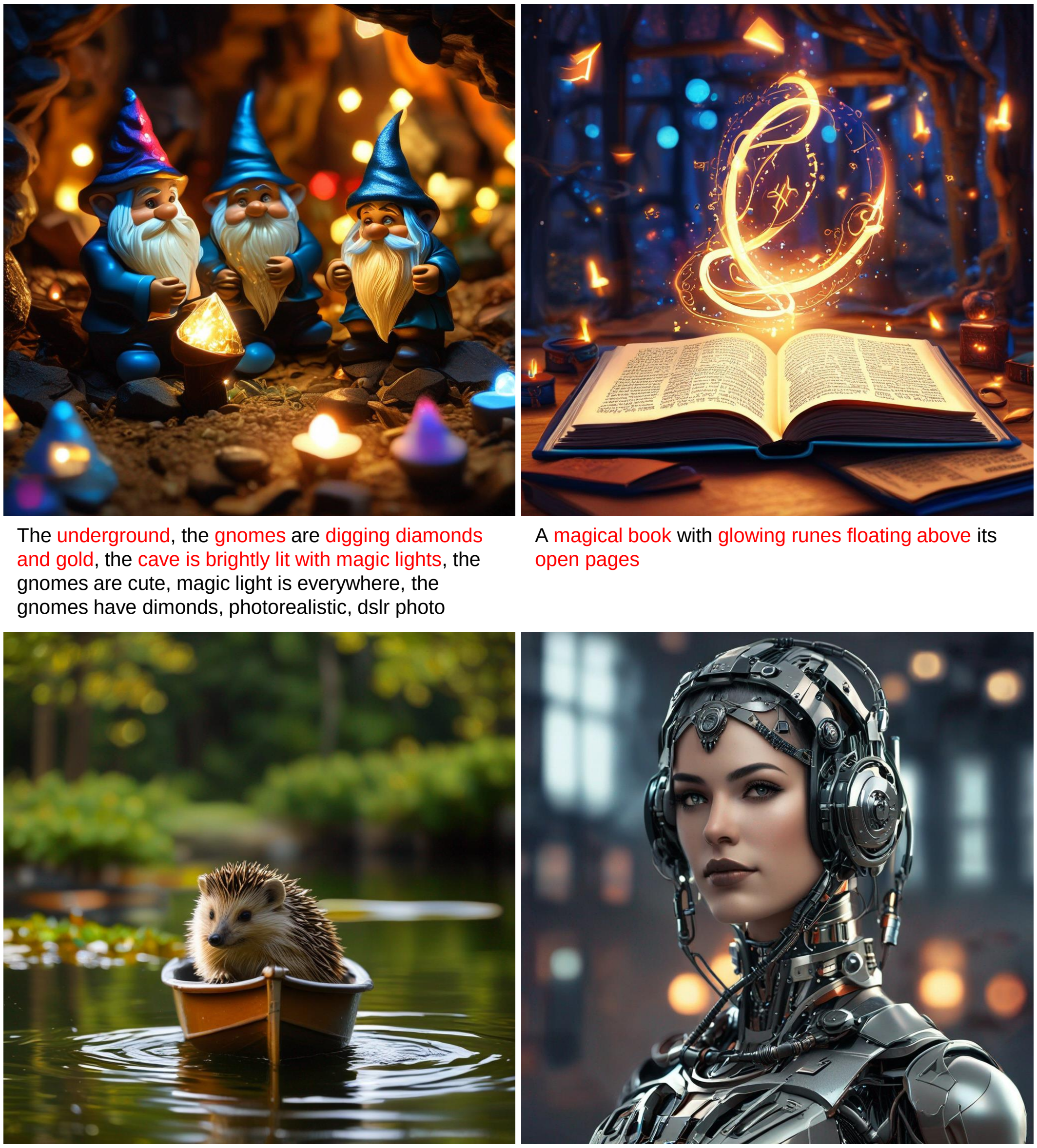
E Societal Impacts

The text-to-image generation model, our KOALA models developed in this study, has the potential to significantly advance the field of visual content creation by enabling the automated generation of diverse and creative images from textual descriptions. This innovation has numerous applications across various industries, including entertainment, education, advertising, and more. However, it is crucial to acknowledge and address the potential risks associated with the misuse of such technology, particularly concerning the generation of Not Safe For Work (NSFW) content.

To mitigate the risks associated with NSFW content, our model leverages the NSFW content detection capabilities provided by Huggingface and the transformers library. By integrating these tools, we ensure that any potentially harmful, violent, or adult content generated by KOALA is identified and filtered out before reaching the end-users. Specifically, the NSFW score is calculated for each generated image, and images with scores exceeding a predefined threshold are automatically discarded. This approach helps maintain the ethical and responsible use of our technology, promoting a safer and more positive user experience.

The adoption of such filtering mechanisms is essential to prevent the spread of inappropriate content and to adhere to ethical standards in AI development. By implementing robust NSFW detection and filtering strategies, we demonstrate our commitment to addressing broader societal concerns and promoting the responsible use of AI-generated content.

In conclusion, while the KOALA model offers significant benefits and opportunities for innovation, we recognize the importance of proactive measures to prevent its potential misuse. Our integration of NSFW content detection serves as a crucial safeguard, ensuring that our contributions to the field align with ethical guidelines and societal values.



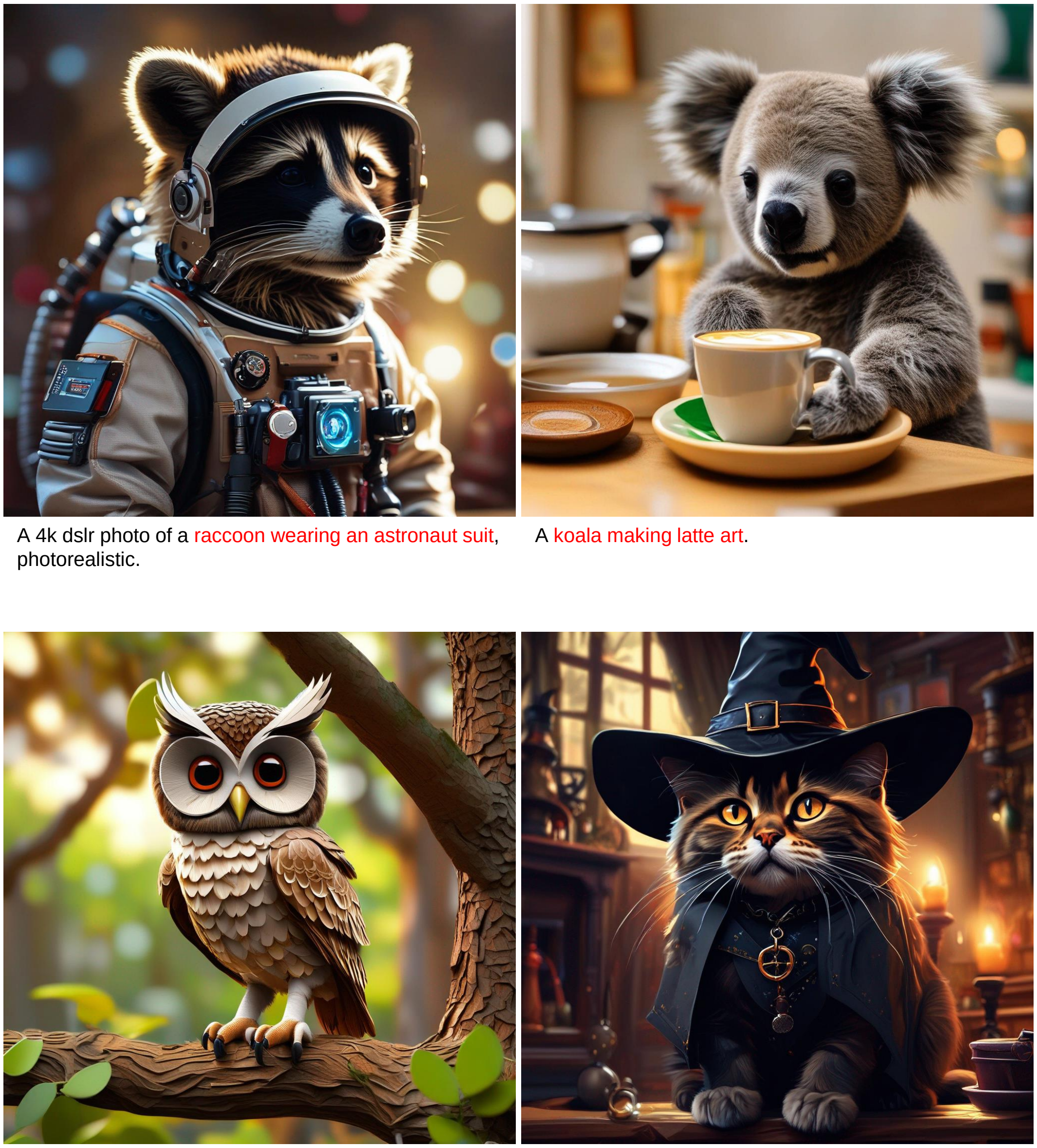
The underground, the gnomes are digging diamonds and gold, the cave is brightly lit with magic lights, the gnomes are cute, magic light is everywhere, the gnomes have dimonds, photorealistic, dslr photo

A magical book with glowing runes floating above its open pages

A 4K dslr photo of a hedgehog sitting in a small boat in the middle of a pond. There are a few leaves in the background.

Portrait photo of a beautiful female cyborg from 1920, trending on artstation

Figure 14: **Qualitative results of KOALA-Lightning-1B models with 1024^2 resolution**.



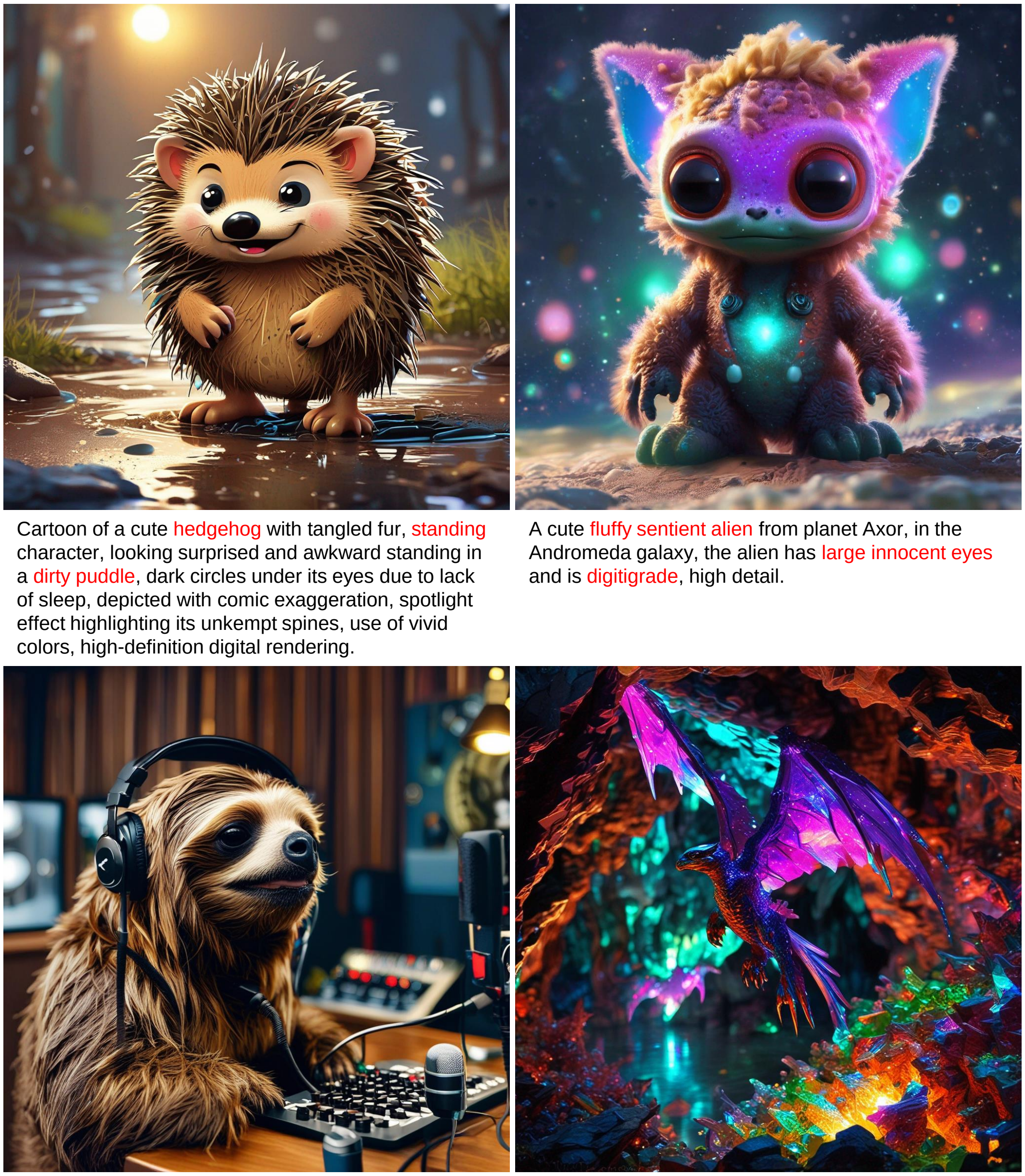
A 4k dslr photo of a raccoon wearing an astronaut suit, photorealistic.

A koala making latte art.

An baby owl on a tree, 3d animation.

A highly detailed zoomed-in digital painting of a cat dressed as a witch wearing a wizard hat in a haunted house, artstation.

Figure 15: **Qualitative results of KOALA-Lightning-1B models with** 1024^2 **resolution**.



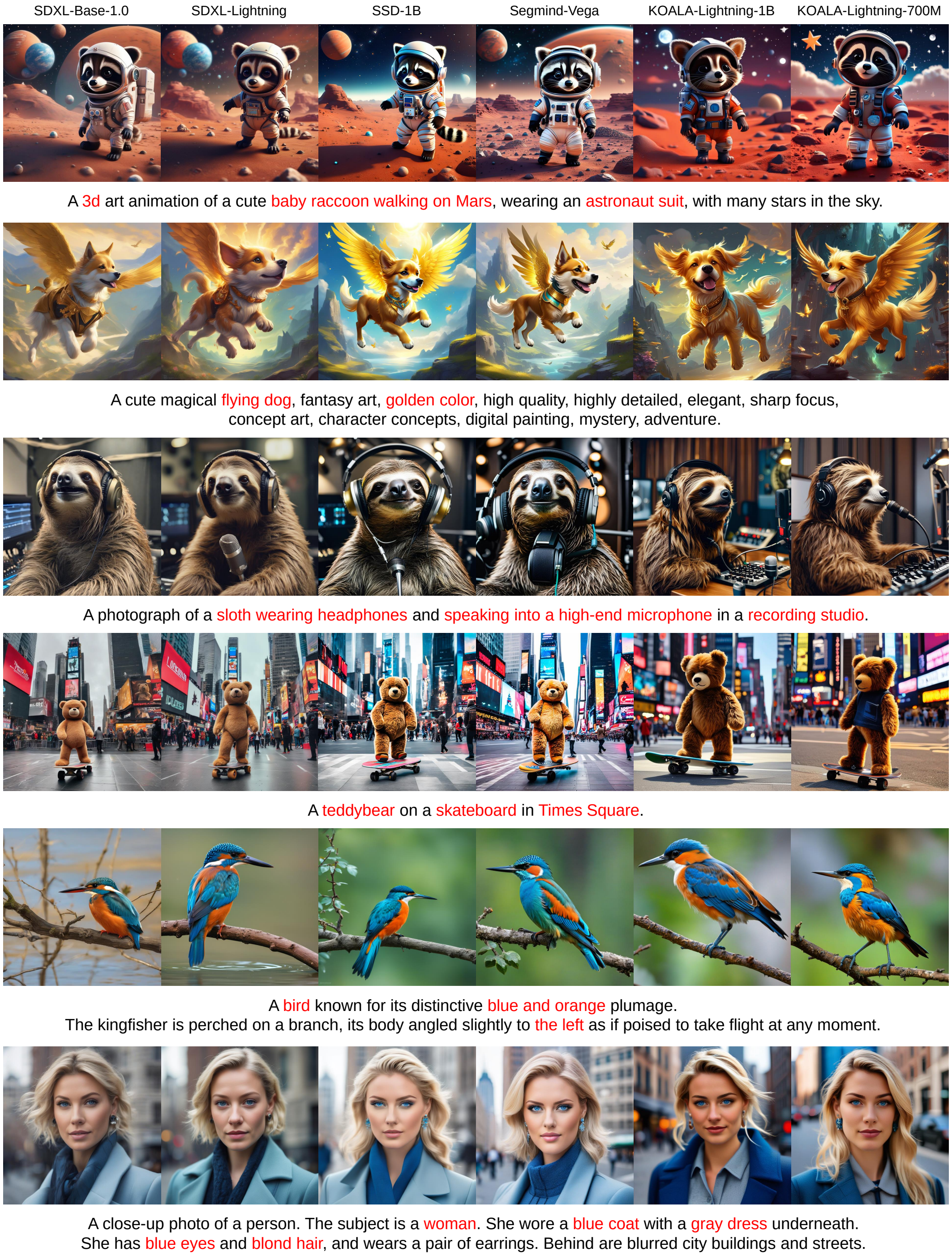
Cartoon of a cute hedgehog with tangled fur, standing character, looking surprised and awkward standing in a dirty puddle, dark circles under its eyes due to lack of sleep, depicted with comic exaggeration, spotlight effect highlighting its unkempt spines, use of vivid colors, high-definition digital rendering.

A cute fluffy sentient alien from planet Axor, in the Andromeda galaxy, the alien has large innocent eyes and is digitigrade, high detail.

A photograph of a sloth wearing headphones and speaking into a high-end microphone in a recording studio.

Iridescent crystal cave, flying lizards creatures from hell

Figure 16: **Qualitative results of KOALA-Lightning-1B models with 1024^2 resolution**.



A 3d art animation of a cute baby raccoon walking on Mars, wearing an astronaut suit, with many stars in the sky.

A cute magical flying dog, fantasy art, golden color, high quality, highly detailed, elegant, sharp focus, concept art, character concepts, digital painting, mystery, adventure.

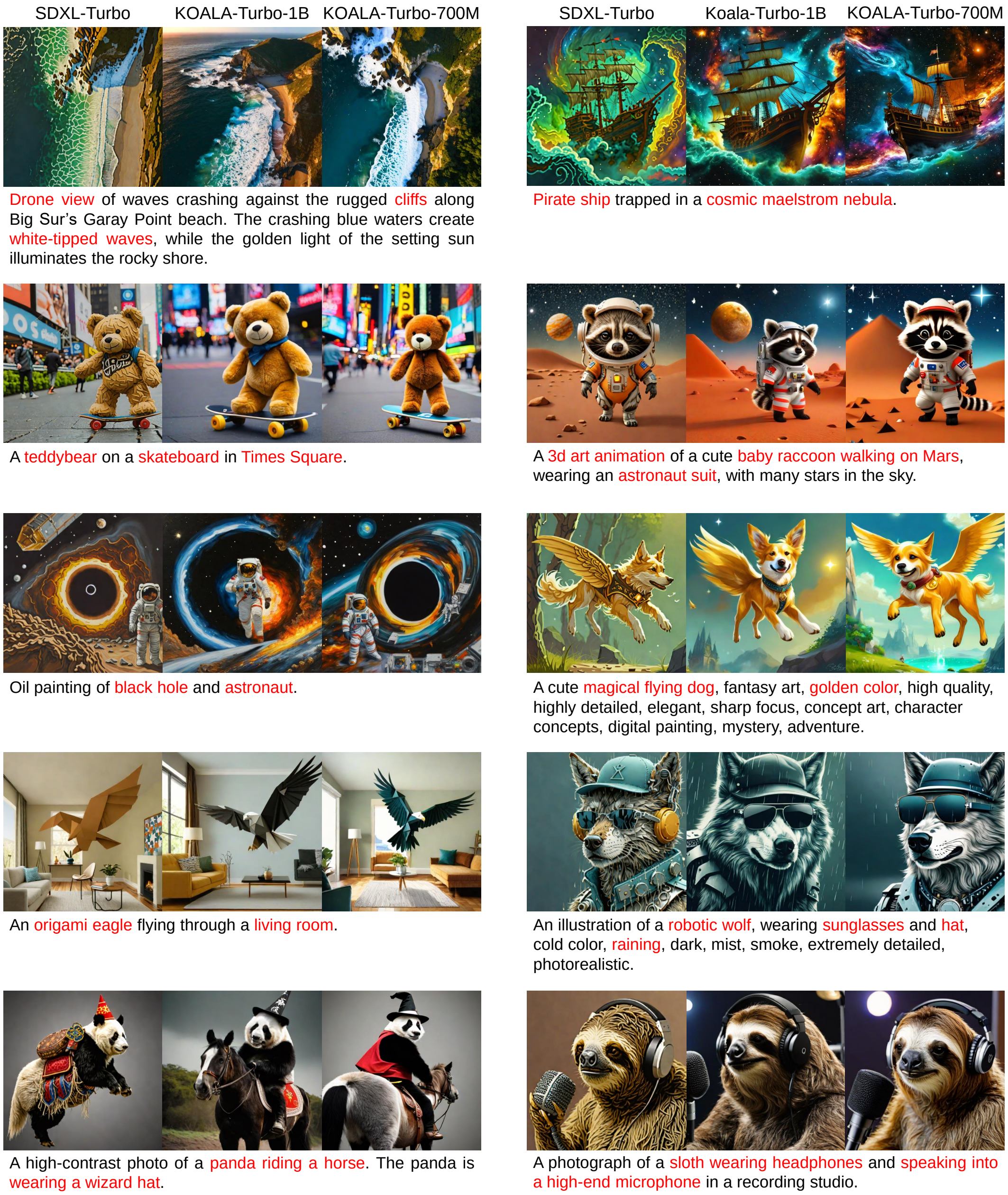
A photograph of a sloth wearing headphones and speaking into a high-end microphone in a recording studio.

A teddybear on a skateboard in Times Square.

A bird known for its distinctive blue and orange plumage.
The kingfisher is perched on a branch, its body angled slightly to the left as if poised to take flight at any moment.

A close-up photo of a person. The subject is a woman. She wore a blue coat with a gray dress underneath.
She has blue eyes and blond hair, and wears a pair of earrings. Behind are blurred city buildings and streets.

Figure 17: **Qualitative comparison with state-of-the-art SDXL models: SDXL-Base-1.0 [31], SDXL-Lightning [23], SSD-1B [10] and SSD-Vega [10] with 1024^2 resolution.**



Drone view of waves crashing against the rugged cliffs along Big Sur's Garay Point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore.

Pirate ship trapped in a cosmic maelstrom nebula.

A teddybear on a skateboard in Times Square.

A 3d art animation of a cute baby raccoon walking on Mars, wearing an astronaut suit, with many stars in the sky.

Oil painting of black hole and astronaut.

A cute magical flying dog, fantasy art, golden color, high quality, highly detailed, elegant, sharp focus, concept art, character concepts, digital painting, mystery, adventure.

An origami eagle flying through a living room.

An illustration of a robotic wolf, wearing sunglasses and hat, cold color, raining, dark, mist, smoke, extremely detailed, photorealistic.

A high-contrast photo of a panda riding a horse. The panda is wearing a wizard hat.

A photograph of a sloth wearing headphones and speaking into a high-end microphone in a recording studio.

Figure 18: **Qualitative comparison of SDXL-Turbo [39]** and our KOALA-Turbo models with 512^2 resolution.

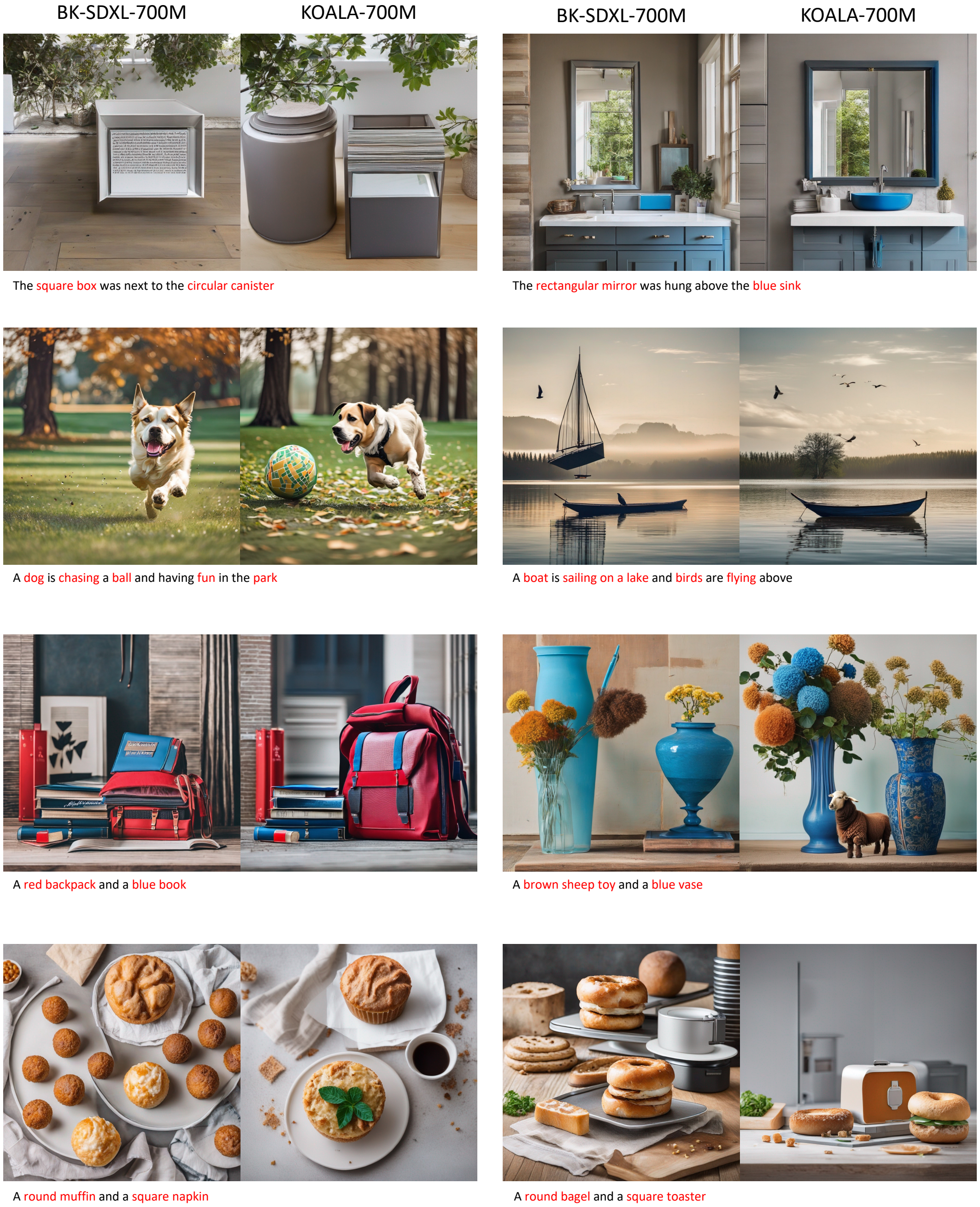

BK-SDXL-700M
KOALA-700M
BK-SDXL-700M
KOALA-700M
The square box was next to the circular canister
The rectangular mirror was hung above the blue sink
A dog is chasing a ball and having fun in the park
A boat is sailing on a lake and birds are flying above
A red backpack and a blue book
A brown sheep toy and a blue vase
A round muffin and a square napkin
A round bagel and a square toaster

Figure 19: **Qualitative comparison between BK-Base-700M vs. KOALA-700M (ours)**. These models are trained with the same training recipe, such as the LAION-A+6 dataset and SDX-Base-1.0 teacher model.

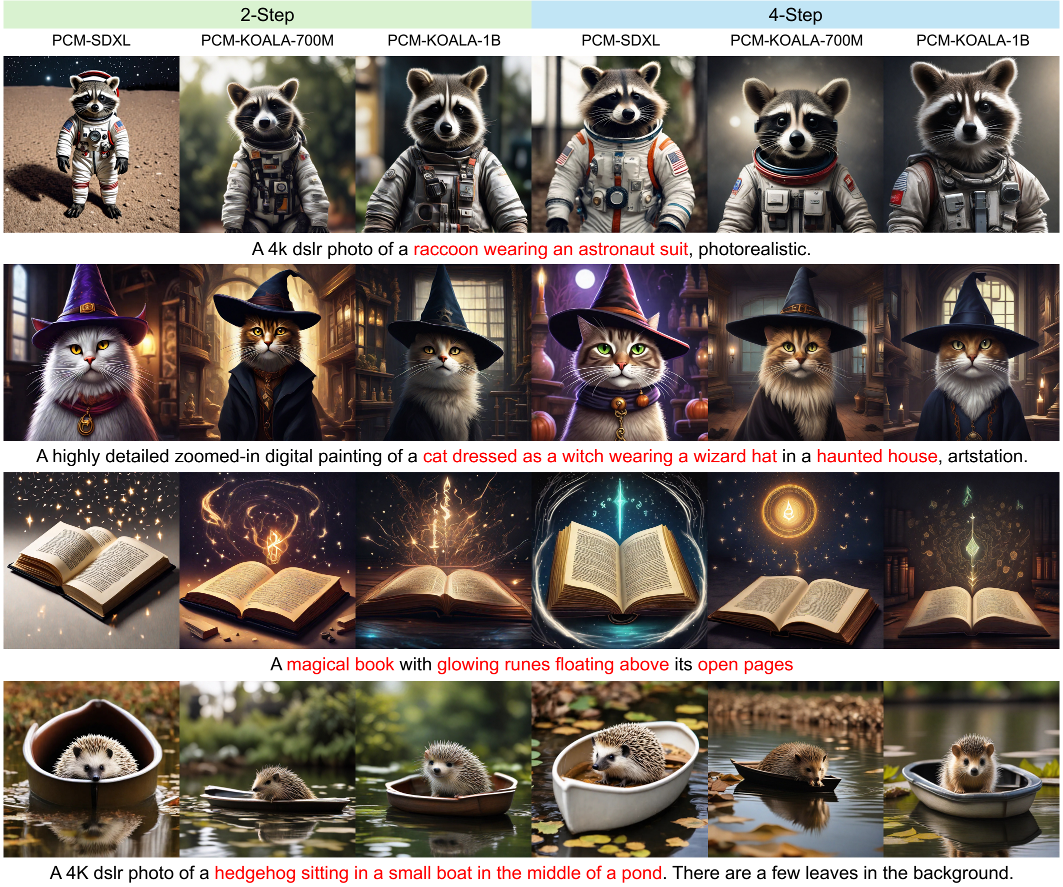
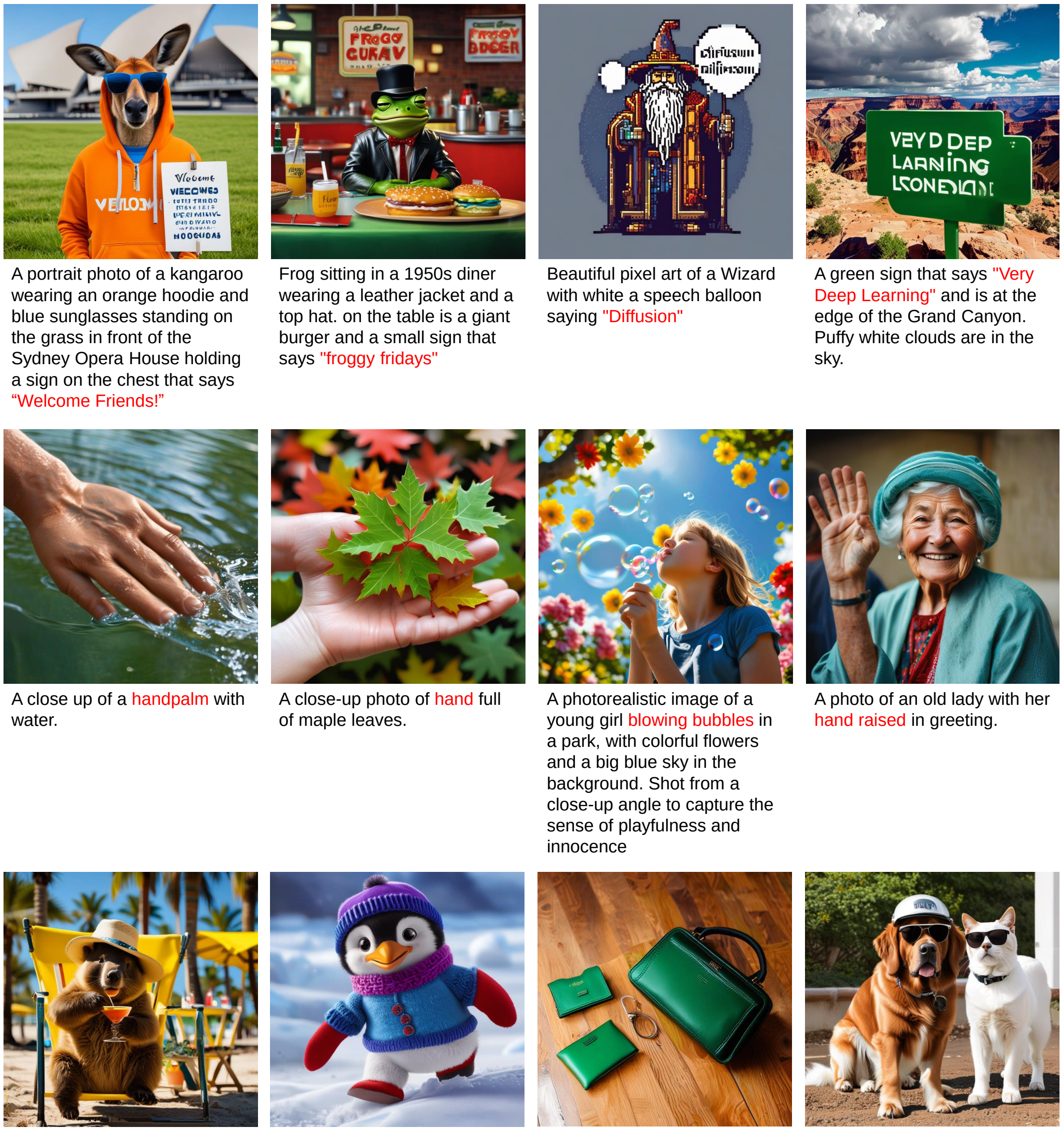

2-Step
4-Step
PCM-SDXL
PCM-KOALA-700M
PCM-KOALA-1B
PCM-SDXL
PCM-KOALA-700M
PCM-KOALA-1B
A 4k dslr photo of a raccoon wearing an astronaut suit, photorealistic.
A highly detailed zoomed-in digital painting of a cat dressed as a witch wearing a wizard hat in a haunted house, artstation.
A magical book with glowing runes floating above its open pages
A 4K dslr photo of a hedgehog sitting in a small boat in the middle of a pond. There are a few leaves in the background.

Figure 20: **Qualitative comparison between PCM-SDXL and our PCM-KOALA models with 1024^2.**



A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says "Welcome Friends!"

Frog sitting in a 1950s diner wearing a leather jacket and a top hat. on the table is a giant burger and a small sign that says "froggy fridays"

Beautiful pixel art of a Wizard with white a speech balloon saying "Diffusion"

A green sign that says "Very Deep Learning" and is at the edge of the Grand Canyon. Puffy white clouds are in the sky.

A close up of a handpalm with water.

A close-up photo of hand full of maple leaves.

A photorealistic image of a young girl blowing bubbles in a park, with colorful flowers and a big blue sky in the background. Shot from a close-up angle to capture the sense of playfulness and innocence

A photo of an old lady with her hand raised in greeting.

A wombat sits in a yellow beach chair, while sipping a martini. The wombat is wearing a white panama hat and a floral Hawaiian shirt. Out-of-focus palm trees in the background. DSLR photograph. Wide-angle view.

A baby penguin wearing a blue hat, red gloves, and purple sweater. Running ice land

A photo of one black handbag and two green wallet on the wooden floor

A brown big dog wearing sunglasses standing on the left and a white small cat wearing helmet sitting on the right

Figure 21: **Failure cases of KOALA-Lightning-1B**. KOALA-Lightning-1B model faces challenges in complex scenarios, such as rendering legible text (1st row), accurately depicting human hands (2nd row), and complex compositional prompts with multiple attributes (3rd row).

NeurIPS Paper Checklist

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 made sure that the claims made in the abstract and introduction accurately reflect our 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: We discuss the limitations of our method in Sec. 5, including failure cases generated by our KOALA and our simple and specific compression method.

   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: This 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: We provide detailed instructions of how to reproduce all of the experiments, hyperparameters, and the models and datasets used, all of which are open-source.

   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: [No]

Justification: We already provided where we can download the training datasets and evaluation benchmarks. However, we cannot provide the training code and checkpoint weights due to the internal policy of our company. After getting permission from our company, we will release the trained weights in Huggingface to generate images.

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 include details for dataset preparation and experimental settings in App. A.

   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: [No]

   Justification: In our study, we compared various models using quantitative evaluations, specifically focusing on HPSv2 and Compbench. The evaluation of HPSv2 required generating 2,400 images, while Compbench required 24,000 images, resulting in a significant inference cost that made repeated inferences challenging. Despite this, we believe that our performance evaluations are reliable as the performance deviations are expected to be minimal based on the sufficient number of test images used in our evaluations. Furthermore, we did not cherry-pick any images during the quantitative evaluation due to the sheer volume of images generated and evaluated. This extensive evaluation approach ensures that the reported performance metrics are representative and robust.

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: We provide the relevant details in App. A and Sec. 4.4.

   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: We have thoroughly reviewed the ethics guidelines and we have made sure to preserve anonymity for the review process.

   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: [Yes]

Justification: We already addressed the broader impacts in App. E.

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: [Yes]

Justification: As a text-to-image generation model, our KOALA, addresses the potential risks of generating Not Safe For Work (NSFW) content, which includes harmful, violent, or adult imagery. To mitigate these risks, KOALA integrates NSFW content detection capabilities provided by Huggingface and the transformers library. By calculating the NSFW score for each generated image and filtering out those that exceed a predefined threshold, our model effectively prevents the creation and dissemination of inappropriate content. This approach ensures the ethical and responsible use of AI technology, aligning our contributions with societal values and ethical guidelines.

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: we have clearly cited the teacher models used for training KOALA: SDXL-Base, SDXL-Turbo, and SDXL-Lightning in the main paper and Appendix.

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: [NA]

Justification: We didn't provide new assets except for our main paper.

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: We have not used crowdsourcing or 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: We have not used crowdsourcing or 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.