Random Conditioning with Distillation for Data-Efficient Diffusion Model Compression

Dohyun Kim[1*] Sehwan Park[1*] Geonhee Han[1]
Seung Wook Kim[2] Paul Hongsuck Seo[1]
[1]Dept. of CSE, Korea University [2]NVIDIA
{a12s12, shp216, rtrt505, phseo}@korea.ac.kr, seungwookk@nvidia.com

Abstract

Diffusion models generate high-quality images through progressive denoising but are computationally intensive due to large model sizes and repeated sampling. Knowledge distillation—transferring knowledge from a complex teacher to a simpler student model—has been widely studied in recognition tasks, particularly for transferring concepts unseen during student training. However, its application to diffusion models remains underexplored, especially in enabling student models to generate concepts not covered by the training images. In this work, we propose Random Conditioning, a novel approach that pairs noised images with randomly selected text conditions to enable efficient, image-free knowledge distillation. By leveraging this technique, we show that the student can generate concepts unseen in the training images. When applied to conditional diffusion model distillation, our method allows the student to explore the condition space without generating condition-specific images, resulting in notable improvements in both generation quality and efficiency. This promotes resource-efficient deployment of generative diffusion models, broadening their accessibility for both research and real-world applications. Code, models, and datasets are available at: https://dohyun-as.github.io/Random-Conditioning

1. Introduction

Diffusion models have emerged as powerful generative frameworks capable of producing high-quality outputs in various domains, such as image [17, 51, 54–57, 60], video [1, 2, 11, 18, 19, 76], and audio [20, 34, 80], by progressively denoising random noise through a sequence of learned steps. Particularly, text-to-image diffusion models trained on large-scale datasets—such as Stable Diffusion [51, 55–57]—excel at generating visually appealing images that accurately align with text prompts. Despite their impressive performance, these models come with significant computational demands, driven by a large number of sampling steps and extensive model parameters. Consequently, there has been growing interest in developing more efficient versions of these models. In this work, we focus on compressing conditional diffusion models to make them more efficient, especially in common real-life scenarios where access to large-scale data is limited, due to practical challenges such as hardware limitations, privacy concerns and licensing restrictions.

Knowledge distillation is a technique that transfers knowledge from one trained network, often a more complex model called the *teacher*, to another, typically a simpler network known as the *student*. Through the use of soft targets [15, 81, 86, 88] or intermediate features [6, 31, 58, 85]

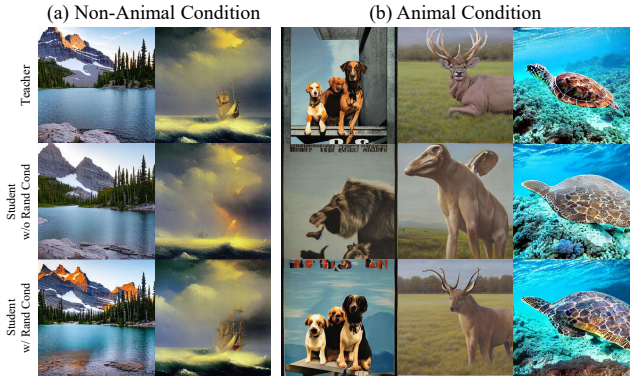


Figure 1. **Qualitative Comparison of Baseline and Our Method Trained Without Animal Image Data.** We train models on a dataset excluding animal-related images, both without and with random conditioning. Each row represents (from top to bottom) the teacher model, the model trained without random conditioning, and the model trained with random conditioning. In (a), samples are generated conditioned on captions unrelated to animals, and in (b), samples are generated conditioned on captions related to animals. The captions used to generate these samples are provided in Sec. L of the Supp. Mat. for reference.

*Equal contribution

from the teacher model, which captures the relationships between concepts, distillation techniques are known to transfer not only seen but also unseen concepts to the student model. For instance, [15] demonstrates that the student model learns to recognize the digit '3' on MNIST [26] although it was never provided an image of '3' during the distillation. Similarly, [44] provides a detailed analysis showing that a teacher's knowledge across multiple domains can be transferred to a student model, even when distillation is performed using data from a single domain. This capability to transfer knowledge of unseen concepts enhances the efficiency of training the student model, making it possible to achieve effective learning even with limited data.

However, unlike in recognition models, this phenomenon is not observed in the context of conditional diffusion models, as we demonstrate in Sec. 3.2 and Fig. 2. The generative function in conditional diffusion models maps the semantic conditioning space to a much larger image space, making it harder for the student model to generalize to unseen concepts. The output noise is also specific to the current input, capturing minimal relationships across different output images. Additionally, each denoising step relies not only on the input condition but also on the intermediate noised images, which further complicates the mapping function. As a result, it becomes challenging for the student model to infer unseen concepts effectively through distillation, necessitating exploration of the entire conditioning space with a large set of condition-image pairs to fully distill the teacher model's generative capacity. However, acquiring such large-scale text-image pairs is often complicated by issues like copyright, privacy, and the storage constraints associated with handling image data. Furthermore, even when images are generated using the teacher model from a text-only dataset, synthesizing them for all possible text prompts can be prohibitively expensive in terms of both computational resources and time.

To address these challenges, we propose a novel technique called random conditioning, where a noised image is paired with a randomly selected, potentially unrelated text condition during training. This method allows the model to learn generalizable patterns without the need to generate images for every text prompt in the dataset, enabling efficient image-free distillation. By reducing the computational and storage demands associated with full image-text mappings, random conditioning preserves strong performance while significantly lowering resource requirements. Our preliminary experiments offer insights into the effectiveness of random conditioning, while extensive main experiments demonstrate that it enables student models to explore an extended condition space. Consequently, as illustrated in Fig. 1, the student learns to generate images containing unseen concepts (*e.g.*, animals in Fig. 1) even when images of these concepts are never provided during the distillation process.

Our main contributions are threefold:

- We provide a novel insight that conditional diffusion models fail to learn teacher knowledge for conditions that are not explicitly explored during the distillation process.
- We propose a novel technique, random conditioning, which allows the student model to explore conditions without requiring paired images.
- Leveraging this technique, we achieve efficient, image-free distillation of conditional diffusion models, producing compact models with competitive generative quality.

2. Related Work

Knowledge Distillation for Model Compression Knowledge distillation is a common approach for model compression, where a smaller model learns to mimic the soft outputs [15, 81, 86, 88] or intermediate features [6, 31, 58, 85] of a larger model, achieving significant compression with minimal performance loss. This technique has been effectively applied across various domains [28, 48, 77], including large language models (LLMs) [22, 63, 73] and vision transformers (ViTs) [12, 74], enabling the creation of models suitable for resource-constrained environments. In these applications, student models successfully learn to generalize to inputs not explicitly exposed during distillation [15, 44]. However, in the context of conditional diffusion models, transferring knowledge for uncovered concepts through distillation remains underexplored. Thus, we investigate this aspect within the scope of data-efficient model compression for diffusion models.

Size-Reduced Diffusion Models While diffusion-based generative models [3–5, 51, 55–57] have shown strong performances, their large parameter counts and model sizes make them difficult to deploy in resource-constrained settings. To address these challenges, various studies [7, 9, 79] have focused on reducing model size through techniques such as quantization [67, 68], architecture evolution [30], and knowledge distillation [24, 27]. Notably, BK-SDM [24] compresses stable diffusion [55, 56] into smaller versions by applying block pruning and feature distillation while KOALA [27] compresses SDXL [32, 51, 64] by employing layer-wise removal and self-attention-based knowledge distillation. We build on previous studies by analyzing the effectiveness of knowledge distillation in conditional diffusion models and propose a general approach for more efficient distillation of diffusion models.

Diffusion Acceleration Recent studies on accelerating diffusion models have focused on reducing the number of sampling steps, rooted in the iterative refinement process of diffusion models. A line of studies aims at accelerating denoising process in diffusion models without training [23, 38, 87], resulting in dramatically reduced sampling steps from a thousand to 10–25. However, fur-

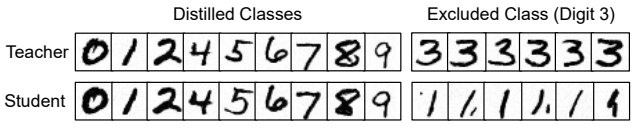


Figure 2. **Generated MNIST Images of Distilled and Excluded Digits by Teacher and Student.** When the student is distilled using a dataset containing only a subset of digits, it fails to generate the excluded digit ('3'). Images from both the teacher and student models are generated with the same random seed for comparison.

ther reductions often cause a steep decline in performance. Distillation-based accelerating methods [10, 25, 35, 36, 39, 40, 61, 71, 83, 84] approach this challenge through knowledge distillation, enabling student models to consolidate multi-step outputs into single-step predictions. For example, Consistency Distillation [25, 39, 71] trains models to produce self-consistent outputs across timesteps, facilitating accurate single-step predictions. These works do not focus on compressing model size; instead, they aim to create few- or one-step models based on a base model. Our research, on the other hand, aims to develop a compressed base model, which could serve as a complementary foundation for step-acceleration methods, enhancing their effectiveness.

3. Method

In this section, we present our novel approach for distilling conditional diffusion models into smaller student models. Sec. 3.1 outlines the problem we aim to address and the associated challenges encountered in this process. Sec. 3.2 describes a naïve baseline approach to tackle this problem, while Sec. 3.3 introduces our proposed method called random conditioning, including its motivation and key observations.

3.1. Distilling Diffusion Models for Compression

Our task is to compress a conditional diffusion model, and in this work we showcase this with Stable Diffusion model for text-to-image generation [51, 55–57], as it is one of the most widely used conditional diffusion models. In other words, we distill the knowledge within a teacher diffusion model $\mathcal{T}$ trained at scale to an arbitrary student model $\mathcal{S}$ that can have a different architecture with a significantly smaller number of parameters. Notably, this task differs from diffusion acceleration via knowledge distillation [10, 25, 35, 36, 39, 40, 61, 71, 83, 84], where the primary aim is to distill a model to decrease the number of diffusion steps required for inference. We approach this task in an image-free setting, where only text prompts are available, without access to any images. This configuration is especially useful, as collecting large-scale image-text pairs is challenging. The process is costly, requires intensive labor for accurate annotation, and is further complicated by privacy concerns and licensing restrictions, which limit access to diverse, high-quality datasets. In certain domains, these issues are even more pronounced, where data scarcity or heightened privacy concerns make it especially difficult to obtain well-annotated image-text pairs.

Applying knowledge distillation to diffusion models without images introduces additional challenges due to the iterative nature of the denoising process. In diffusion models, the forward and reverse processes are defined on some time interval $[0, T]$, and the teacher model predicts the noise $\boldsymbol{\epsilon}_{\mathcal{T}}(\mathbf{x}_t, t, c)$ to be removed from $\mathbf{x}_t$ at each timestep $t \in [0, T]$ given a text condition c. Therefore, knowledge transfer from the teacher model to the student model must occur at each timestep t. However, without access to images, generating the intermediate noisy input $\mathbf{x}_t$, which is typically created by adding noise to the original image $\mathbf{x}_0$ [17, 69], becomes challenging. This limitation prevents us from performing knowledge distillation for $t \neq T$ where T is the total number of denoising steps, as we lack the necessary input image at intermediate timesteps.

3.2. Naïve Baseline Approach

A naïve approach for image-free distillation would involve generating images for all available text prompts to construct a paired dataset $\mathcal{D} = \{(\mathbf{x}^n, c^n)\}_{n=1}^N$ where $\mathbf{x}^n$ is the generated image that serves as original image $\mathbf{x}_0$ for the text condition c^n allowing us to construct noisy input image $\mathbf{x}_t$ for any timestep t and condition c^n. Since diffusion models are time-intensive for image generation, we need to generate and cache these images in advance to build the dataset. The teacher model can then be distilled into a student model with the following loss function:

$$\mathcal{L}_{\text{out}} = \mathbb{E}_{(\mathbf{x}_t, c) \in \mathcal{D}, t} \left[\| \boldsymbol{\epsilon}_{\mathcal{T}}(\mathbf{x}_t, c, t) - \boldsymbol{\epsilon}_{\mathcal{S}}(\mathbf{x}_t, c, t) \|_2^2 \right], \quad (1)$$

where $\boldsymbol{\epsilon}_{\mathcal{T}}$ and $\boldsymbol{\epsilon}_{\mathcal{S}}$ are the predicted noises by the teacher and student models, respectively. Here, $(\mathbf{x}_t, c)$ is a pair sampled from the dataset $\mathcal{D}$ with noise injected into the image based on t, which is uniformly distributed between 0 and T. In addition, we may incorporate a feature-level knowledge distillation loss function, which is given by

$$\mathcal{L}_{\text{feat}} = \mathbb{E}_{(\mathbf{x}_t, c) \in \mathcal{D}, t} \left[\sum_l \| \mathbf{f}_{\mathcal{T}}^l(\mathbf{x}_t, c, t) - \mathbf{f}_{\mathcal{S}}^l(\mathbf{x}_t, c, t) \|_2^2 \right], \quad (2)$$

where $\mathbf{f}_{\mathcal{T}}^l$ is the feature maps from layer l of the teacher model and $\mathbf{f}_{\mathcal{S}}^l$ denotes the feature maps from the corresponding layer of the student models. Note that $\mathcal{T}$ and $\mathcal{S}$ do not need to have the same architecture; we can incorporate additional temporary modules for distillation to project arbitrary intermediate features of $\mathcal{S}$ to $\mathbf{f}_{\mathcal{S}}^l$ with the same dimensionality as the corresponding features $\mathbf{f}_{\mathcal{T}}^l$. These additional

Figure 3. **Effects of Altered Conditioning on Generated Results from an Input Image across Timesteps.** Generated results conditioned on the rightmost column using the input image from the leftmost column at each timestep for both MNIST [26] and MSCOCO [33]. First, $\mathbf{x}_t$ is derived from the initial image $\mathbf{x}_0$, associated with the image label, at the timestep t shown above each image using the forward process and then, $\mathbf{x}_0$ is regenerated through the reverse process, conditioned on the displayed rightmost column.

projection modules are then discarded after the distillation process. This feature-level loss, combined with the noise prediction loss, encourages the student model to replicate both the external outputs and the internal processing of the teacher model. It allows the student to learn and replicate the teacher model's denoising behavior for the text conditions c encountered during the distillation process.

While this naïve approach enables effective knowledge transfer from the teacher to the student model, it presents several limitations. The method requires generating images $\mathbf{x}_0$ for a diverse set of text prompts to sufficiently cover the text condition space. Without covering the entire condition space, the student model may fail to generate images for those conditions that have never been observed during distillation. Our preliminary experiment on MNIST [26] in Fig. 2 illustrates the importance of covering the condition space. Although the teacher model can generate the digit '3', the student model fails to produce this digit when it has not been exposed to this condition during distillation. Since the text condition space is exceedingly large—unlike the 10-digit space in MNIST—synthesizing $\mathbf{x}_0$ for all possible prompts becomes prohibitively costly in terms of computation, time, and storage. This challenge is particularly significant with diffusion models, which rely on multiple timesteps during inference, further compounding the computational demands for each generated image.

3.3. Random Conditioning

To address the above challenges, we propose random conditioning illustrated in Fig. 4 that allows us to cache images generated from only a subset of text prompts (blue box). Formally, given an extensive set of M text prompts $\mathcal{C}$, we construct a dataset of N image-text pairs $\mathcal{D} = (\mathbf{x}^n, c^n)$

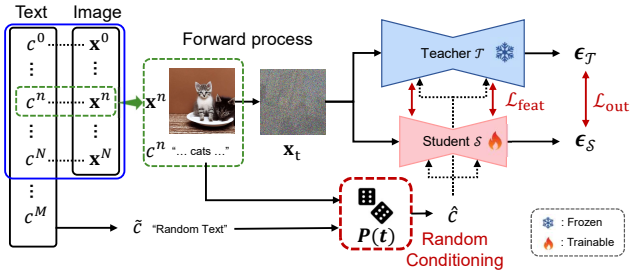


Figure 4. **Overview of the Random Conditioning Approach.** When distilling knowledge from the teacher model to a smaller student model, instead of pairing each training image dataset sample $\mathbf{x}_t^n$ with its original condition c^n, we replace it with a random condition $\tilde{c}$ from the text dataset based on a predefined probability $p(t)$ at each timestep t. This approach enables the student model to learn the teacher's behavior even for conditions without explicit image pairs.

where $N \ll M$. As discussed above, training a student model on this paired dataset $\mathcal{D}$ would limit the knowledge transfer in distillation as there are many uncovered parts in the text condition space that could be covered by those texts in $\mathcal{C}$. Note that this limitation arises from the absence of noisy input images $\mathbf{x}_t$, which are typically constructed from the original image $\mathbf{x}_0$, with the generated images in $\mathcal{D}$ serving as these originals. In our approach, we leverage not only $\mathcal{D}$, which contains a limited number of generated images, but also $\mathcal{C}$, allowing the student model to explore all text conditions in $\mathcal{C}$. This approach enhances the distilled knowledge, enabling the model to generalize across the full condition space.

Precisely, we first sample a paired data $\mathbf{x}^n$ and c^n from

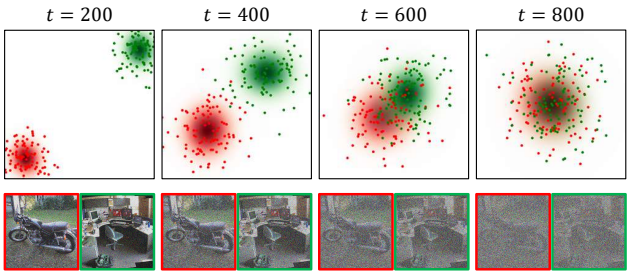


Figure 5. **Distributions of $p(\mathbf{x}_t|c^n)$ and $p(\mathbf{x}_t|\tilde{c})$.** Visualization of the distributions of toy 2D data samples at timesteps 200, 400, 600, and 800, along with corresponding $\mathbf{x}_t$ images at each timestep. As the timestep increases, the distributions progressively overlap with each other.

$\mathcal{D}$ and construct $\mathbf{x}_t$ from $\mathbf{x}^n$. Then, before performing distillation, we apply a predefined random conditioning probability $p(t)$ to sample a random text from $\mathcal{C}$. Specifically, the text condition $\hat{c}$ is determined by

$$\hat{c} = \begin{cases} c^n & \text{with probability } 1 - p(t), \\ \tilde{c} \in \mathcal{C} & \text{with probability } p(t), \end{cases} \tag{3}$$

where $\tilde{c}$ is randomly sampled from $\mathcal{C}$. Finally, $\hat{c}$ is paired with $\mathbf{x}_t$ to compute both distillation losses defined in Eq. (1) and (2).

Observations and Motivation While the proposed random conditioning technique may initially appear counterintuitive, it is grounded in our empirical observation that diffusion models incorporate conditioning information in a manner that varies with the timestep t. Fig. 3 shows the generated outputs during the denoising process, starting from $\mathbf{x}_t$ at various timesteps t on the MNIST [26] and MS-COCO [33] datasets. In each row, $\mathbf{x}_t$ is derived from the same initial image $\mathbf{x}_0$ corresponding to the leftmost column, and the generated outputs share the same conditioning, displayed in the rightmost column. Notably, this condition differs from the label associated with the original image $\mathbf{x}_0$. The generated images primarily align with either the original image label or the conditioning value, with only a narrow range of t producing outputs with noticeable artifacts. Specifically, when t is small, the generated images tend to reflect the original image label (red boxes) due to the low noise magnitude characteristic of later steps in the denoising process. Conversely, when t is large, the generated images predominantly follow the conditioning value (blue boxes), as the input $\mathbf{x}_t$ becomes nearly indistinguishable from pure noise. These results also indicate that the condition c does not need to be strongly correlated with the noised input $\mathbf{x}_t$ supporting the proposed random conditioning technique. This is due to: (1) the model's tendency to rely almost entirely on the input condition c at large t where the original semantics of $\mathbf{x}_0$ are nearly lost, and (2) the model's primary focus on denoising the input $\mathbf{x}_t$ while disregarding the condition c when t is small. Furthermore, Fig. 5 demonstrates that, as the noise level or timestep t increases during the forward process, the distributions $p(\mathbf{x}_t|c^n)$ and $p(\mathbf{x}_t|\tilde{c})$ become closer to each other, eventually merging into the same Gaussian distribution as t approaches T. This observation implies that the input image and condition do not need to be directly aligned at every timestep. It supports both the effectiveness and validity of our random conditioning method, highlighting its flexibility in associating conditions with diverse inputs. Based on these observations and the motivation, we empirically explored $p(t)$. When $p(t)$ was set as a constant value, such as $p(t) = 1$, the results were suboptimal. In particular, reducing $p(t)$ for intermediate time steps, where the pairing between the image and condition becomes relatively more important, led to improved performance. Among these, we used an exponential function for $p(t)$ in our experiments. Further experiments regarding $p(t)$ are provided in Sec. D of the Supp. Mat.

Extended Exploration of Condition Space As explored in Fig. 2, the student model effectively learns to generate images for conditions explicitly covered by the paired dataset $\mathcal{D}$ during distillation, but generating images for every text prompt in $\mathcal{C}$ poses a significant bottleneck. Random conditioning alleviates this by allowing the use of conditions not included in $\mathcal{D}$ to be applied without requiring paired images. Consequently, the student can explore text prompts beyond those paired with images, even when the number of conditions far exceeds the available images. This setup helps the student replicate the teacher's behavior under novel conditions, thereby broadening its generative capabilities.

4. Experiments

4.1. Datasets

LAION We use LAION [65, 66] consisting of 400M image-text pairs. Following [24], we use 212K samples from the LAION-Aesthetics V2 (L-Aes) 6.5+ [65], which is a subset of LAION. To simulate image-free training, we extract text prompts only from those 212K samples and generate their images. For random conditioning, we use 20M extra text prompts randomly sampled from 400M pairs of LAION. It is worth noting that original images from LAION are still used in baseline methods, as these methods are developed for setups requiring image access.

MS-COCO The MS-COCO [33] dataset is a large-scale text-image paired dataset with diverse and detailed annotations, including 80 object classes. Following previous practices [53, 57, 60], we use 30K image-text pairs sampled from the MS-COCO validation split, which consists of 41K images, each with five human-annotated captions. For

each image, we use a single caption preselected from [24] for evaluation.

4.2. Experimental Settings

Models For our experiments, we use the Stable Diffusion (SD) v1.4 model [55] as the teacher model. BK-SDM [24] serves as our baseline method, representing the current SOTA in diffusion model compression using knowledge distillation. Unlike our approach, which operates in an image-free setup, BK-SDM utilizes both the original images and text. BK-SDM proposes three compressed architectures—Base, Small, and Tiny—by selectively removing blocks from the teacher network to achieve compression. For a fair comparison, we evaluate our method using the same compressed architectures. Additionally, we evaluate four further compressed architectures that reduce the number of channels while preserving all layers, offering an alternative compression approach. Three of these architectures are sized to match the Base, Small, and Tiny configurations from BK-SDM, while the fourth is even smaller than the Tiny architecture (C-Micro), pushing compression further.

Evaluation Metrics We evaluate the models using standard metrics commonly applied in text-to-image generation: Fréchet Inception Distance (FID) [14], Inception Score (IS) [62], and the CLIP score [13, 52]. FID and IS focus on measuring the visual fidelity and diversity of generated images. The CLIP score evaluates the alignment between the generated image and the text prompt. We use the Inception-v3 model for computing FID and IS, while the ViT-g/14 model is used for calculating the CLIP score.

Implementation Details We adopt all hyperparameters from [24] except for null condition proportion [16], which is set to 10%. We utilize four 40GB NVIDIA A100 GPUs with a batch size of 256 for distillation. We train the models using the AdamW [37] optimizer with a learning rate of 5e-5. We use the two losses of Eq. (1) and Eq. (2), with equal weights of 1. For Eq. (2), feature distance is reduced after each block in the U-Net [8, 59].

4.3. Results

Effects of Random Conditioning Tab. 1 demonstrates the effectiveness of random conditioning. The top three rows show scores without random conditioning, while the bottom three rows display their corresponding scores with random conditioning applied. In particular, comparing Rows 1 and 4, random conditioning shows a substantial performance boost with 14.72% decrease in FID and 8.29% increase in IS, indicating its significant impact. Previous studies [24] have demonstrated that initializing the student with teacher weights can enhance performances. Here, by comparing Rows 1 and 2, as well as Rows 4 and 5, we observe a similar performance increase through initialization. Notably, even when teacher initialization is applied, random conditioning still adds meaningful performance gains. Furthermore, models with random conditioning and random initialization achieve comparable or even superior performance to those with teacher initialization but without random conditioning, highlighting the powerful impact of random conditioning on model performance.

#	Rand Cond	T Init	Real image	FID↓	IS↑	CLIP↑
1	✗	✗	✗	18.13	31.84	0.2728
2	✗	✓	✗	18.15	33.81	0.2864
3	✗	✓	✓	15.76	33.79	0.2878
4	✓	✗	✗	15.46	34.48	0.2834
5	✓	✓	✗	15.76	36.03	0.2895
6	✓	✓	✓	**15.00**	**36.14**	**0.2933**

Table 1. **Impact of Random Conditioning.** We compare models trained with and without random conditioning across various settings, varying teacher initialization and the availability of real images on MS COCO-30k. All models are based on the B-Base architecture. "Rand Cond" denotes whether random conditioning is applied, "T Init" indicates whether the model is initialized from the teacher model, and "Real image" specifies the use of real images during training. Notably, Row 3 is the same as BK-SDM [24].

In Rows 5 and 6, the scores are nearly identical, underscoring that our method maintains strong performance without real image usage. Random conditioning contributes more significantly to achieving high scores than the impact of real image usage, establishing it as the key factor in score enhancement. By efficiently distilling knowledge from the teacher model, our approach achieves comparable results without needing real images, providing a practical and robust solution for knowledge distillation in conditional diffuion model even without access to actual image data.

Knowledge Transfer of Unseen Concepts To demonstrate the effect of random conditioning in transferring knowledge of unseen concepts—specifically, conditions excluded from the paired training dataset $\mathcal{D}$—we train student models on a dataset that omits all images containing animals as the unseen concepts. To exclude animal images from training, we apply a filtering process to the original 212K LAION [65] dataset using GPT [47], BLIP [29], and keyword elimination. This process yields 188K non-animal text prompts from the original 212K samples. The models are trained with generated images from these 188K prompts and for those with random conditioning, we utilize additional text prompts without generating images. For evaluation, we test models on two subsets—33K non-animal prompts and 8K animal-related prompts—as well as on the MS-COCO 30K. Detailed process of filtering animal-related data from both the training and evaluation sets is provided respectively in Sec. J of the Supp. Mat.

Tab. 2 illustrates the results in this configuration. Without random conditioning (Row 1), this model fails to learn the unseen concepts of animals showing poor performances

#	Rand Cond	Additional Texts	Seen (Non-animal)			Unseen (Animal)			Seen+Unseen		
			FID↓	IS↑	CLIP↑	FID↓	IS↑	CLIP↑	FID↓	IS↑	CLIP↑
(Teacher)			13.29	32.47	0.2954	22.53	18.63	0.3035	12.67	36.71	0.2971
1	✗	None	15.24	28.11	0.2801	37.86	**17.73**	0.2478	15.66	29.62	0.2734
2	✓	24K animal-related texts	**14.42**	27.86	0.2788	**23.26**	17.18	0.2833	**13.50**	31.30	0.2797
3	✓	24K+20M	15.37	**30.27**	**0.2879**	24.71	17.39	**0.2913**	14.47	**34.06**	**0.2886**

Table 2. **Knowledge Transfer of Unseen Concepts through Random Conditioning.** The row with a gray background shows the performance of the teacher model [55] for reference. "Rand Cond" indicates the use of random conditioning, and "Additional Texts" specifies the amount and source of extra text data used for random conditioning. All models are trained on images generated from approximately 188K non-animal prompts, obtained by excluding 24K animal-related samples from the original 212K LAION dataset. In addition, Row 2 leverages those 24K animal-related texts excluded from the set used to generate training images, while Row 3 further utilizes 20M LAION texts. Evaluation is conducted across three setups: "Seen," which tests generation of non-animal concepts; "Unseen," for animal-related concepts; and "Seen+Unseen," which includes both. All student models use the B-Base architecture.

#	Rand Cond	Data Source	FID↓	IS↑	CLIP↑
(Teacher)			13.05	36.76	0.2958
1	✗	LAION	18.15	33.81	0.2864
2	✓	LAION	15.76	36.03	0.2896
3	✓	GPT	**14.98**	**36.70**	**0.2952**

Table 3. **Model Comparisons with Varying Data Constraints.** The row with a gray background shows the performance of the teacher model [55] for reference. "Rand Cond" indicates whether random conditioning is used, and "Data Source" specifies the text data used for both paired image generation and additional conditioning. Row3 represents the fully data-free configuration, where even text data are unavailable and are generated automatically using an LLM (*e.g.*, GPT). For fair comparison, all models are trained with 212K generated images. Row2 uses additional 20M LAION captions, while Row 3 uses 2.2M GPT-generated prompts. All student models are based on the B-Base architecture.

whereas it maintains comparable performances in generating images with seen concepts. When random conditioning is applied, the model (Row 2) achieves significant improvements especially in FID and the CLIP score by facilitating the filtered 24K texts but without their images. Finally, extending the text dataset with prompts from LAION (Row 3) leads to further improvements in both IS and CLIP scores for both seen and unseen cases.

Beyond unseen categories, our model also surpasses those Base models without random conditioning in all metrics for seen concepts, achieving scores that closely approach those of the teacher model. This suggests that random conditioning not only enhances knowledge of unseen concepts but also boosts overall generation quality. Among our models, those utilizing more text data generally exhibit better performance overall. Qualitative results in Fig. 1 further illustrate that the quality of generated images for unseen concepts is distinctly better when random conditioning is applied, compared to when it is not. Detailed analysis of the impact of extra text dataset sizes and more qualitative examples are provided in Secs. C and M of the Supp. Mat.

Data-Free Distillation In Tab. 3, we evaluate the effectiveness of random conditioning in a fully data-free setup, where even text data are unavailable for distillation. In this setting (Row 3), text prompts are automatically generated by an LLM, as described in Sec. A of the Supp. Mat., and a 212K subset is used to synthesize images, forming a paired dataset. Remarkably, even without real text data, the model in Row 3 not only outperforms the baseline without random conditioning (Row 1) but also achieves performance comparable to the model trained with real text data and random conditioning (Row 2). This demonstrates the scalability and adaptability of our method in resource-constrained settings. Furthermore, LLM-generated captions in this setup can be tailored to the target domain, offering the potential to steer the student model toward specific generation styles or tasks.

Comparisons to Other Text-to-Image Models We build our models by applying two compression strategies: block compression, which removes UNet blocks, and channel compression, which reduces channel widths. Block-compressed models (B-Base, B-Small, B-Tiny) follow [24], use pretrained teacher weights, and achieve significant parameter reduction with minimal performance drops. Channel compression allows greater flexibility for higher compression rates. We design C-Base, C-Small, C-Tiny with parameter counts comparable to block-compressed models, and introduce C-Micro, which has 30% fewer parameters than B-Tiny. Due to channel size mismatches, channel-compressed models cannot reuse teacher weights. Details on multiply-accumulate operations (MACs), UNet parameter counts, and additional comparisons between these models are provided in Sec. B of the Supp. Mat.

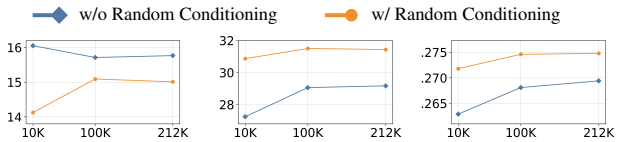
Tab. 4 compares our compressed models with other diffusion models, presenting total parameter counts, number of real images used for training, and performances. Our B-Base, B-Small, and B-Tiny models share the same architecture as their corresponding BK-SDM models and are distilled from the same teacher model. However, our enhanced distillation approach yields superior performance. Note that our models with higher compression rates outperform BK-SDM's larger models, despite BK-SDM being trained with real images and teacher initialization. For ex-

Models	#Params	#Images	FID↓	IS↑	CLIP↑
SDM-v1.4 [55, 57]†	1.04B	>2000M	13.05	36.76	0.2958
Small SD [50]†	0.76B	229M	12.76	32.33	0.2851
BK-SDM-Base †	0.76B	0.22M	15.76	33.79	0.2878
BK-SDM-Small †	0.66B	0.22M	16.98	31.68	0.2677
BK-SDM-Tiny †	0.50B	0.22M	17.12	30.09	0.2653
B-Base [Ours]	0.76B	0	14.47	36.50	0.2932
B-Small [Ours]	0.66B	0	16.22	35.99	0.2804
B-Tiny [Ours]	0.50B	0	16.71	35.46	0.2782
C-Base [Ours]	0.73B	0	14.45	34.92	0.2904
C-Small [Ours]	0.61B	0	14.43	34.58	0.2888
C-Tiny [Ours]	0.49B	0	13.90	33.18	0.2860
C-Micro [Ours]	0.40B	0	13.42	32.64	0.2813
GLIDE [43]†	3.5B	250M	12.24	-	-
LDM-KL-8-G [57]†	1.45B	400M	12.63	30.29	-
DALL·E-2 [54]†	5.2B	250M	10.39	-	-
SnapFusion [30]†	0.99B	>100M	∼13.6	-	∼0.295
Würstchen-v2 [49]†	3.1B	1700M	22.40	32.87	0.2676
Pixart-alpha [3]	5.4B	25M	23.43	34.54	0.3072
SDXL-Base-1.0 [51]	3.5B	-	12.15	35.12	0.3199
SD 3.5 Medium [72]	7.9B	-	16.23	39.81	0.3246

Table 4. **Comparison with Other Models on MS-COCO 30K.** Despite having significantly fewer parameters than other large models, our model achieves comparable performance with minimal quality degradation. "#Params" refers to the total number of parameters. "#Images" refers to the quantity of real images used in training. †Results reported from [24].

Figure 6. **Qualitative Comparison between Our Models and Baseline Models.** From left to right: samples generated from the teacher model, BK-SDM Base, BK-SDM Tiny, B-Base (ours), and C-Micro (ours) using the same prompts and seeds. The captions used are provided in Sec. L of the Supp. Mat.

ample, our smallest model, C-Micro, surpasses BK-SDM Small across all evaluation metrics, even with 50% fewer parameters in the UNet compared to BK-SDM Small as discussed with Tab. A. It is important to note that while BK-SDM Small benefits from teacher weights and real images, C-Micro is trained from random initialization without use of any real images. Finally, B-Base shows significant improvements across all three metrics over BK-SDM Base with the same architecture, and even approaches the performance levels of the teacher model (SDM-v1.4). Fig. 6 compares the generated images of our B-Base and C-Micro with those of the teacher model, BK-SDM Base and BK-SDM



(a) FID (b) IS (c) CLIP

Figure 7. **Impact of Random Conditioning by Cache Size.** We evaluate the models trained with and without random conditioning, using different cache sizes of 10k, 100k, and 212k. We test with B-Base architecture with 125K training iterations.

Tiny. Notably, C-Micro demonstrates high-quality images despite its compact size. Compared to other large diffusion models that rely on hundreds of millions of training images, our models achieve comparable performance with far fewer parameters and without using any real images, by distilling knowledge from a well-trained teacher model. These results underscore the effectiveness of our method, providing an efficient compression solution that maintains high performance.

Impact of the Number of Generated Images In Fig. 7, we compare B-Base models with and without random conditioning across different numbers of generated images: 10K, 100K, and 212K. Across FID, IS, and CLIP scores, models trained with random conditioning consistently outperform those without it. Notably, the performance gap widens with fewer generated images (*e.g.*, 10K), showing that random conditioning enables effective distillation even with limited data. Remarkably, the model trained on 10K images with random conditioning outperforms the one trained on 212K images without it, highlighting the strength of the proposed approach.

5. Conclusion

Our work shows that random conditioning enables the student model to learn to generate images of concepts beyond those present in the training image dataset. This capability allows the student to explore a wide text condition space during conditional diffusion model distillation, enhancing performance. This method effectively compresses large diffusion models into smaller, efficient versions. Additionally, our development of a compact base diffusion model supports use in resource-limited settings and encourages further research advancements. In this work, the teacher model employed in our experiments is based on Stable Diffusion v1.4. We expect that using more advanced versions, such as SDXL, would lead to improved performance due to their enhanced capabilities. Although our random conditioning method is broadly applicable to distilling conditional diffusion models, our experiments were conducted exclusively on text-to-image models. To generalize our findings, future works include extending this approach to diffusion models for other modalities.

Acknowledgements

This research was supported by the IITP grants (IITP-2025-RS-2020-II201819, IITP-2025-RS-2024-00436857, RS-2024-00398115), the NRF grants (NRF-2021R1A6A1A03045425) and the KOCCA grant (RS-2024-00345025) funded by the Korea government (MSIT, MOE and MSCT).

References

[1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[2] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024. 1

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 2, 8, 16

[4] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024.

[5] Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-δ: Fast and controllable image generation with latent consistency models, 2024. 2

[6] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[7] Yu-Hui Chen, Raman Sarokin, Juhyun Lee, Jiuqiang Tang, Chuo-Ling Chang, Andrei Kulik, and Matthias Grundmann. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *CVPR-Workshop*, 2023. 2

[8] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 6

[9] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*, 2023. 2

[10] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. BOOT: Data-free distillation of denoising diffusion models with bootstrapping. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. 3, 14

[11] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations*, 2024. 1

[12] Zhiwei Hao, Jianyuan Guo, Ding Jia, Kai Han, Yehui Tang, Chao Zhang, Han Hu, and Yunhe Wang. Learning efficient vision transformers via fine-grained manifold distillation. In *NeurIPS*, 2022. 2

[13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 6

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS 2014 Deep Learning Workshop*, 2015. 1, 2

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 6, 16

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 1, 3

[18] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, 2022. 1

[19] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 1

[20] Jiawei Huang, Yi Ren, Rongjie Huang, Dongchao Yang, Zhenhui Ye, Chen Zhang, Jinglin Liu, Xiang Yin, Zejun Ma, and Zhou Zhao. Make-an-audio 2: Temporal-enhanced text-to-audio generation, 2023. 1

[21] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 15

[22] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020. 2

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 2

[24] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *Computer Vision – ECCV 2024*, pages 381–399, Cham, 2025. Springer Nature Switzerland. 2, 5, 6, 7, 8, 13, 15, 16

[25] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *ICLR*, 2024. 3

[26] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 2, 4, 5

[27] Youngwan Lee, Kwanyong Park, Yoorhim Cho, Yong-Ju Lee, and Sung Ju Hwang. Koala: Empirical lessons toward memory-efficient and fast diffusion models for text-to-image synthesis, 2023. 2, 15

[28] Gang Li, Xiang Li, Yujie Wang, Shanshan Zhang, Yichao Wu, and Ding Liang. Knowledge distillation for object detection via rank mimicking and prediction-guided feature imitation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):1306–1313, 2022. 2

[29] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 6, 16

[30] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. In *NeurIPS*, 2023. 2, 8

[31] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and G. Wang. Knowledge distillation via the target-aware transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[32] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 2

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4, 5

[34] Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883, 2024. 1

[35] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 3

[36] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024. 3

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[38] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022. 2

[39] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3

[40] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, 2023. 3

[41] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. 13

[42] Thuan Hoang Nguyen and Anh Tran. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 14

[43] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 8

[44] Utkarsh Ojha, Yuheng Li, Anirudh Sundara Rajan, Yingyu Liang, and Yong Jae Lee. What knowledge gets distilled in knowledge distillation? In *Advances in Neural Information Processing Systems*, 2023. 2

[45] OpenAI. Gpt-3.5-turbo. https://platform.openai.com/docs/models/gpt-3.5-turbo, 2023. 13, 16

[46] OpenAI. Gpt-4o. https://platform.openai.com/docs/models/gpt-4o, 2024. 16

[47] OpenAI. Gpt-4 technical report, 2024. 6

[48] S. Panchapagesan, D. S. Park, C.-C. Chiu, Y. Shangguan, Q. Liang, and A. Gruenstein. Efficient knowledge distillation for rnn-transducer models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021. 2

[49] Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Wuerstchen: An efficient architecture for large-scale text-to-image diffusion models. *arXiv preprint arXiv:2306.00637*, 2023. 8

[50] Justin Pinkney. Small stable diffusion. https://huggingface.co/OFA-Sys/small-stable-diffusion-v0, 2023. 8

[51] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 8, 15, 16

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[53] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 5

[54] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1, 8

[55] Robin Rombach and Patrick Esser. Stable diffusion v1-4. https://huggingface.co/CompVis/stable-diffusion-v1-4, 2022. 1, 2, 3, 6, 7, 8, 15

[56] Robin Rombach and Patrick Esser. Stable diffusion v1-5. https://huggingface.co/runwayml/stable-diffusion-v1-5, 2022. 2

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 5, 8

[58] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 1, 2

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 6

[60] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 1, 5

[61] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 3

[62] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 6

[63] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS Workshop*, 2019. 2

[64] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. 2

[65] Christoph Schuhmann and Romain Beaumont. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics, 2022. 5, 6, 16

[66] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021. 5

[67] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, 2023. 2

[68] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[69] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 3

[70] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 16

[71] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 3

[72] Stability AI. Stable diffusion 3.5 medium. https://huggingface.co/stabilityai/stable-diffusion-3.5-medium, 2024. 8, 16

[73] S. Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Conference on Empirical Methods in Natural Language Processing*, 2019. 2

[74] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. 2

[75] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 16

[76] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report, 2023. 1

[77] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020. 2

[78] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022. 17, 18, 19

[79] Qianlong Xiang, Miao Zhang, Yuzhang Shang, Jianlong Wu, Yan Yan, and Liqiang Nie. Dkdm: Data-free knowledge distillation for diffusion models with any architecture, 2024. 2, 14

[80] Jinlong Xue, Yayue Deng, Yingming Gao, and Ya Li. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *arXiv preprint arXiv:2401.01044*, 2024. 1

[81] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2021. 1, 2

[82] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22552–22562, 2023. 16

[83] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024. 3

[84] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 3

[85] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2017. 1, 2

[86] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11953–11962, 2022. 1, 2

[87] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. DPM-solver-v3: Improved diffusion ODE solver with empirical

model statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[88] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias–variance tradeoff perspective. In *International Conference on Learning Representations*, 2021. 1, 2

[89] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning*, 2024. 14

[90] Mingyuan Zhou, Zhendong Wang, Huangjie Zheng, and Hai Huang. Guided score identity distillation for data-free one-step text-to-image generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 14

[91] Ligeng Zhu. Thop: Pytorch-opcounter. https://github.com/Lyken17/pytorch-OpCounter, 2018. 13