LD-Pruner: Efficient Pruning of Latent Diffusion Models using Task-Agnostic Insights

Thibault Castells Hyoung-Kyu Song Bo-Kyeong Kim
Shinkook Choi
Nota AI
Seoul, Korea
{thibault, hyoungkyu.song, bokyeong.kim, shinkook.choi}@nota.ai

Abstract

Latent Diffusion Models (LDMs) have emerged as powerful generative models, known for delivering remarkable results under constrained computational resources. However, deploying LDMs on resource-limited devices remains a complex issue, presenting challenges such as memory consumption and inference speed. To address this issue, we introduce LD-Pruner, a novel performance-preserving structured pruning method for compressing LDMs. Traditional pruning methods for deep neural networks are not tailored to the unique characteristics of LDMs, such as the high computational cost of training and the absence of a fast, straightforward and task-agnostic method for evaluating model performance. Our method tackles these challenges by leveraging the latent space during the pruning process, enabling us to effectively quantify the impact of pruning on model performance, independently of the task at hand. This targeted pruning of components with minimal impact on the output allows for faster convergence during training, as the model has less information to re-learn, thereby addressing the high computational cost of training. Consequently, our approach achieves a compressed model that offers improved inference speed and reduced parameter count, while maintaining minimal performance degradation. We demonstrate the effectiveness of our approach on three different tasks: text-to-image (T2I) generation, Unconditional Image Generation (UIG) and Unconditional Audio Generation (UAG). Notably, we reduce the inference time of Stable Diffusion (SD) by 34.9% while simultaneously improving its FID by 5.2% on MS-COCO T2I benchmark. This work paves the way for more efficient pruning methods for LDMs, enhancing their applicability.

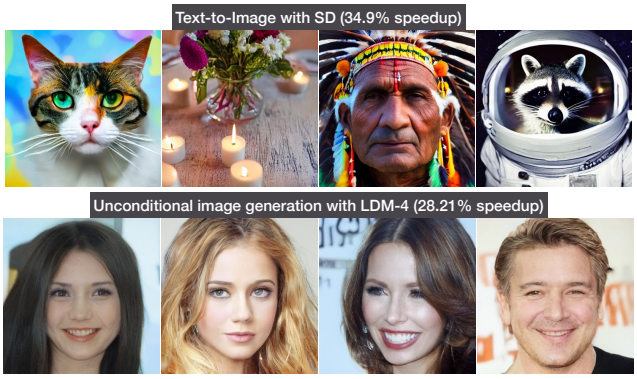


Figure 1. Samples generated using our compressed models. The proposed compression technique applies structured pruning to LDMs using task-agnostic information. Prompts (left to right): "A multi-colored cat with yellow eyes staring upward", "Candles and flowers neatly placed on a table", "Portrait of a chief indian, 4k, high definition", "A photo of a raccoon wearing an astronaut helmet, looking out of the window at night."

1. Introduction

Generative models [10, 17], which can learn a data distribution and generate a sample from it, have revolutionized numerous domains, such as computer vision and natural language processing. Among them, Diffusion Models (DMs) [13] have recently gained significant attention for their ability to generate high-quality images. LDMs, a subset of DMs that performs the diffusion process in a latent space, have witnessed rapid growth in popularity due to their fast generation capabilities and reduced computational cost [23, 31, 36]. However, their deployment on resource-limited devices remains a challenge, mainly because of large compute requirements from the Unet in LDMs [16].

A variety of strategies have been developed to compress LDMs and enhance their deployment feasibility, including quantization [20], low-rank filter decomposition [11], and

token merging [1]. The primary goal of these techniques is to reduce the model's compute cost while striving to maintain its original performance, a crucial aspect of deploying models in resource-constrained environments. The work presented in this paper takes a distinct, yet complementary, approach to these studies.

Pruning, another compression technique which is traditionally utilized for the compression of convolutional networks by eliminating non-critical connections [2, 19, 24], has been recently applied to DMs in the form of Diff-Pruning [7]. This method identifies non-contributory diffusion steps and important weights using informative gradients, and applies filter pruning, significantly reducing computational overhead. However, Diff-Pruning does not extend its application beyond UIG. Moreover, its adaptability is further curtailed by the necessity to tune a threshold hyper-parameter for determining the optimal number of steps, as the ideal threshold is found to differ across datasets.

Training an LDM from scratch is both computationally demanding and financially costly. For instance, the reported training time for SD is a staggering 150,000 A100 hours [3], translating to an estimated cost on the order of magnitude of $100,000. To ensure the retention of model performance throughout the pruning process, thus making the training faster, one might consider assessing the impact of pruning on model performance without fine-tuning [27]. While this method may prove effective for straightforward evaluative tasks like image classification, it becomes burdensome for generative models. Firstly, each different task—be it image generation, audio generation, and so on—requires its own unique evaluation tool. This introduces a lack of generalizability that compounds the complexity. Secondly, the performance evaluation of generative models is both complex and resource-intensive. Take, for example, the Fréchet Inception Distance (FID) [12], a commonly used evaluation metric for image generation. Its application requires the generation of thousands of images, a process that could take over an hour, making it impractical to use this metric for assessing the impact of each potential pruning operation. Moreover, the reliability of such metrics can be contentious [32], further complicating the process.

We introduce a novel task-agnostic metric to measure the importance of individual operators, which are fundamental building blocks of the LDM architecture, such as convolutional layers and attention layers. We leverage this metric for structured pruning of LDMs. Our approach distinguishes itself by leveraging the latent space during the pruning process, specifically by assessing the impact of modifications within the model's latent representations. Operating in the latent space, where data is compact, provides dual benefits. Firstly, it ensures our method's independence from output types, facilitating a seamless adaptation to any task without necessitating adjustments. The use of cross-attention in the Unet of conditional LDMs to blend embeddings of different tasks in the latent space serves as a prime illustration of this task-agnostic property. Secondly, it yields computational efficiency, thereby addressing the performance evaluation challenge encountered in previous works.

Our method effectively identifies and removes components that contribute minimally to the output, leading to compressed models with faster inference speed and fewer parameters, without a major drop in performance. Through this work, we hope to extend the current body of work on LDM compression and enhance the deployment of LDMs in resource-constrained environments, expanding their applicability across various scenarios.

The main contributions of this paper are:

- We propose a novel, comprehensive metric designed specifically to compare the latent representations of LDMs. This metric is underpinned by thorough experimental evaluations and logical reasoning, ensuring that each element of its design contributes effectively to the accurate and sensitive comparison of LDM latents.
- Leveraging this new metric, we formulate a novel, task-agnostic algorithm for compressing LDMs through architectural pruning. The primary focus of our proposed method is to maintain output quality during the pruning process, thereby accelerating the finetuning phase as the weights are preserved.
- We demonstrate the versatility of our approach through its application in three distinct tasks: T2I generation, UIG and UAG. The successful execution of these experiments underscores our method's potential for wide applicability across diverse tasks.

The remainder of this paper is organized as follows: Sec. 2 presents our proposed method in detail, highlighting its novelty and how it overcomes the aforementioned limitations. Sec. 3 outlines the experimental setup and evaluation metrics. Finally, Sec. 4 compares our approach to existing methods and provides further analysis of our design choices and discusses potential areas for future improvement.

2. Proposed Method

This section describes our novel algorithm for compressing LDMs. The proposed method employs structured pruning aimed at minimizing performance loss, regardless of the generation task.

2.1. Method Overview

Our ultimate goal is to enhance an LDM efficiency by minimizing the presence of less impactful operators, thus streamlining the architecture without sacrificing effectiveness. In the preliminary stage of our approach, we focus

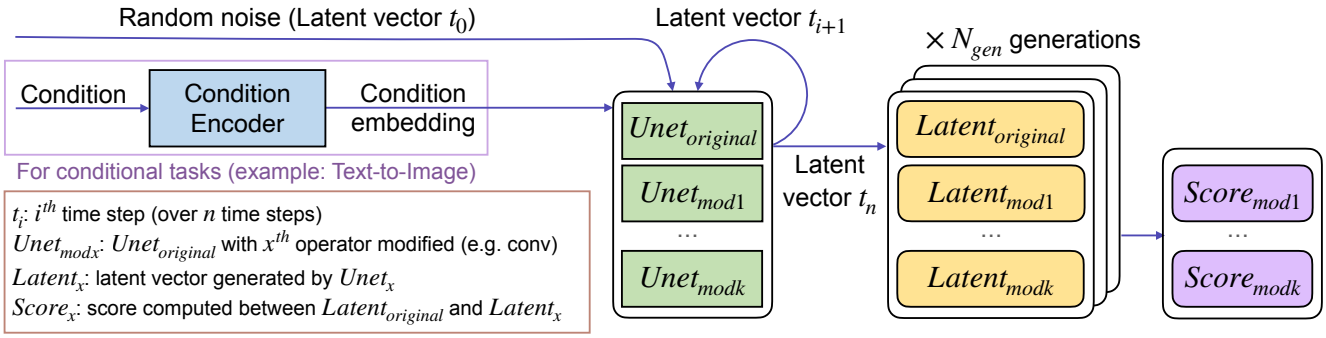


Figure 2. Overview of LD-Pruner. Given k operators in the Unet, we generate $k+1$ sets of N_{gen} latent vectors: one set for the original Unet, and one for each Unet where a single operator has been modified. The importance score of each operator is then calculated using a formula specifically designed to compare latent vectors. This formula, sensitive to shifts in both the central tendency and the variability of the latent vectors, generates a comprehensive measure of the importance of each operator.

on collecting the information needed to assess the individual significance of each operator within the Unet of the LDM. This involves systematically modifying each operator to simulate potential pruning effects, and carefully tracking these modifications by generating unique latent representations associated with each change. This information gathering enables us to determine the potential impact of removing or reducing certain components, guiding our subsequent pruning decisions. Using a tailored scoring formula, we quantify the divergence between the original and altered latent representations, offering insight into the impact of each modification. The scores derived from this process then inform our pruning phase, where we decide which operators to prune or replace. A visual depiction of this method, highlighting the key stages of the computation of our pruning score, is provided in Fig. 2.

2.2. Operator Modification and Latent Representation Collection

Transitioning to the practical aspects of our methodology, we first focus on the meticulous modification of the operators within the Unet of the LDM. In the context of our work, an operator refers to a fundamental building block of the architecture, such as convolutional layers, attention layers, normalization layers, activation functions, or more complex components like transformer blocks. For each of these operators, we consider one of two possible modifications. First, we attempt to eliminate the operator in its entirety, provided this action is feasible. However, there are situations where operator removal is not viable—specifically, when the operator's input and output dimensions do not align. In these instances, we resort to an alternative approach: replacing the operator with a less computationally demanding operation that retains the original dimensions. If there is a disparity in the number of channels, we use a 1×1 convolution operation to match the dimensions without adding significant computational overhead. In cases where the spatial resolution varies, we use average pooling or upscaling. For implementation details, please refer to the Supplementary Materials.

Following the modification of an operator, we generate multiple latent representations using the modified model. Each set of latent representations corresponds to a specific operator modification and collectively forms a comprehensive record of the model's output under various operator modifications. After each set of latent representations is obtained, we restore the modified operator to its original state. This ensures that each operator is modified in isolation, preventing the cumulative effects of multiple modifications from influencing the assessment of individual operators.

For the original, unmodified model, we also generate latent representations (referred to as the original set) that serve as a baseline for comparison. This comprehensive collection of latent representations, both from the original and modified models, forms the foundation for the subsequent evaluation of operator significance in our methodology.

2.3. Operator Significance Evaluation

The evaluation of operator significance is a critical step in our pruning method. This process is based on the assumption that a significant change in the latent space is likely to lead to a substantial change in the model's output. In order to quantify this change and thus estimate the significance of each operator, we employ a specially designed scoring formula crafted to capture the difference between two sets of latent representations.

Let $\mathbf{L}_{orig}$ and $\mathbf{L}_{mod}$ denote the original set and a modified set, respectively, each of which contains N_{gen} latent representations. We denote the i-th latent vector in $\mathbf{L}_{orig}$ and $\mathbf{L}_{mod}$ as $\mathbf{l}_{orig,i}$ and $\mathbf{l}_{mod,i}$, respectively. The scoring formula consists of two main components: the average distance, denoted as avg_{dist}, measures the distance between the average values of $\mathbf{L}_{orig}$ and $\mathbf{L}_{mod}$, and the standard de-

viation distance, denoted as std_{dist}, measures the distance between the standard deviations of $\mathbf{L}_{orig}$ and $\mathbf{L}_{mod}$. Formally, we compute avg_{dist} and std_{dist} as follows:

$$avg_{dist} = |avg_{orig} - avg_{mod}|_2, \tag{1}$$
$$std_{dist} = |std_{orig} - std_{mod}|_2 \tag{2}$$

where $|\cdot|_2$ denotes the Euclidean norm and where:

$$avg_{orig} = \frac{1}{N_{gen}} \sum_{i=1}^{N_{gen}} \mathbf{l}_{orig,i}, \tag{3}$$
$$avg_{mod} = \frac{1}{N_{gen}} \sum_{i=1}^{N_{gen}} \mathbf{l}_{mod,i} \tag{4}$$

and:

$$std_{orig} = \sqrt{\frac{1}{N_{gen}} \sum_{i=1}^{N_{gen}} (\mathbf{l}_{orig,i} - avg_{orig})^2}, \tag{5}$$
$$std_{mod} = \sqrt{\frac{1}{N_{gen}} \sum_{i=1}^{N_{gen}} (\mathbf{l}_{mod,i} - avg_{mod})^2} \tag{6}$$

The score S for each operator is then computed by summing avg_{dist} and std_{dist}:

$$\text{score} = avg_{dist} + std_{dist}. \tag{7}$$

This scoring formula is designed to be sensitive to both shifts in the central tendency and changes in the variability of the latent representations, providing a comprehensive measure of the impact of operator modification. By using this formula, our method can effectively identify the operators that are most (higher score) and least (lower score) significant to the model's performance, guiding the pruning process toward the most efficient and least disruptive modifications. We further discuss our metric in Sec. 4.2.

2.4. Model Pruning

After calculating the significance scores for each operator, the computed scores serve as a roadmap, guiding us in identifying the operators that could be pruned or substituted with the least potential impact on the model's performance. Our strategy particularly focuses on operators with the lowest scores for elimination, since these are regarded as the least contributory to the model's output.

Determining the number of operators to prune, denoted as k, requires a deliberate and systematic evaluation. This process is essentially a trade-off exercise between achieving model compression and preserving performance. An increase in the number of pruned operators leads to a more compact model, however, it may also risk a significant reduction in performance. Therefore, our goal is to identify an optimal value for k that offers a substantial degree of model compression while maintaining satisfactory performance levels. Such trade-off is discussed in Sec. 4.4.

In the case of conditional LDMs, such as SD, we conduct the evaluation for various conditions. For every operator, we aggregate the scores across all conditions. This approach ensures that the pruning does not overly specialize for a specific condition. In practice, we used 50 different prompts conditions for our SD experiment. The entire pruning process is concisely summarized in Alg. 1. Notably, in this context, an unconditional task can essentially be interpreted as a conditional task with only a single implicit condition.

Upon completing the pruning process, we engage in fine-tuning the pruned model to recoup any performance reduction that occurred as a consequence of the pruning operation.

Algorithm 1 Efficient Pruning for LDMs

Input: LDM unet $Unet$, list of condition $CList$, generation per condition N_{gen}, number of operator to prune k

$scores \leftarrow$ empty dictionary
for C in $CList$ **do**
  $latent_{orig} \leftarrow generate(Unet, C, N_{gen})$
  for operator op in $Unet$ **do**
    $Unet_{mod} \leftarrow prune(Unet, op)$
    $latent_{mod} \leftarrow generate(Unet_{mod}, C, N_{gen})$
    $s \leftarrow get\_score(latent_{orig}, latent_{mod})$ (Eq. 7)
    $scores[op.name] \leftarrow scores[op.name] + s$
  end for
end for
$Unet \leftarrow prune(Unet, scores, k)$

2.5. Complexity Analysis

We provide a time complexity analysis of our proposed algorithm, focusing on the main operations involved. Let's denote n the number of computational operations in the Unet to generate a single latent representation, m the total number of operators that are potential candidates for pruning, and k the number of latent representations to generate for each operator modification. Given these definitions, our algorithm's time complexity can be expressed as $O(nmk)$. This complexity can be effectively managed in practice. For instance, we can decrease m by filtering out operators that contribute negligibly to the overall model latency, thereby focusing our efforts on the more significant contributors. An additional advantage stems from the specific nature of our compression method, which operates in the latent space of the LDMs. In contrast to other pruning methods that require

Model	FID ↓	IS ↑	CLIP ↑	# Params	Data Size	Speedup
SD-v1.4 [31]	13.05	**36.76**	**0.2958**	1.04B	>2000M	0%
LD-Pruner (ours) (42 modifications)	**12.37**	**35.77**	**0.2894**	**0.71B**	**0.22M**	**34.89%**
Small Stable Diffusion [25]	12.76	32.33	0.2851	0.76B	229M	35.28%
BK-SDM-Base [16]	15.76	33.79	0.2878	0.76B	**0.22M**	35.28%
BK-SDM-Small [16]	16.98	31.68	0.2677	0.66B	**0.22M**	**36.98%**
DALL·E [29]	27.5	17.9	-	12B	250M	
DALL·E-2 [30]	**10.39**	-	-	5.2B	250M	
CogView [5]	27.1	18.2	-	4B	30M	
CogView2 [6]	24.0	22.4	-	6B	30M	
Make-A-Scene [9]	11.84	-	-	4B	35M	
LAFITE [37]	26.94	26.02	-	**0.23B**	3M	
GALIP (CC12M) [35]	13.86	25.16	0.2817	0.32B	12M	
GLIDE [28]	12.24	30.29	-	5B	250M	
LDM-KL-8-G [31]	12.63	-	-	1.45B	400M	
SnapFusion [21]	∼13.6	-	∼0.295	0.99B	>100M	
Würstchen-v2 [26]	22.40	32.87	0.2676	3.1B	1700M	

For the IS and FID values of comparative models, we adopt the evaluations as reported in Kim et al. [16]

Table 1. Comparison of different models for T2I Generation, on the MS-COCO 256 × 256 validation set. Speedup values are measured relatively to SD-v1.4.

the generation of full outputs, our technique only needs the latent representations. This allows us to avoid the decoding step during generation, thereby reducing the overall computational burden and accelerating the pruning process.

3. Experimental Setup

To highlight the task agnostic property of the proposed importance score, we apply it to three different tasks: T2I Generation with SD-v1.4 [31], UIG with LDM-4 [31] and UAG with AudioDiffusion [4].

3.1. Training

For each task, we finetune our compressed Unets employing Knowledge Distillation (KD), applied both at the feature and output levels [16]. For the detailed hyper-parameters, please refer to the Supplementary Materials.

T2I Generation. We finetune our compressed model on a subset of 0.22M image-text pairs from the LAION-Aesthetics V2 6.5+ dataset [34], which represents less than 0.1% of the training pairs used in the LAION-Aesthetics V2 5+ [34] for training SD-v1.4. All training is conducted on a single A100 GPU.

Unconditional Image Generation. For UIG, we leverage the complete CelebA-HQ 256 × 256 dataset [14] due to its relatively small size (approximately 30k images). All training is conducted on a single NVIDIA GeForce RTX 3090 GPU.

Unconditional Audio Generation. For UAG, we finetune using the same dataset that was employed to train AudioDiffusion. This dataset consists of 20k Mel spectrograms of size 256 × 256, generated from 5-second audio files. All training is conducted on a single NVIDIA GeForce RTX 3090 GPU.

3.2. Evaluation

Performance metric. We report the FID as our main performance metric for image generation. In the case of T2I generation, the FID is measured by generating 30k samples from the MS-COCO 256×256 validation set [22]. In Tab. 1, we additionally present the Inception Score (IS) [33], computed using the same dataset as for the FID computation. In the case of UIG, the FID is measured by generating 5k samples, and we compute the FID with the training set as commonly done with CelebA-HQ [31]. For UAG, we employ the Fréchet Audio Distance (FAD) [15], a specialized variant of the FID tailored for audio comparison. Again, the FAD is measured between 5k generated samples (following the author's recommendation) and the training set.

Computation Efficiency metric. Unlike some previous work, such as Diff-Pruning, our study focuses on real-world inference speed improvements rather than relying solely on FLOPs and MACs, which can be unreliable predictors of actual speed [8, 18]. We conduct our inference speed evaluations for T2I generation and UIG on a single NVIDIA GeForce RTX 3090 GPU, performing 30 and 200 inference steps, respectively. For UAG, we utilize a CPU (Intel Xeon Silver 4210R) due to the model's small size, and carry out 50 inference steps (∼ 5-6 seconds). To ensure the stability and reliability of our results, we perform a warmup process by initially generating 20 samples. Subsequently, we compute the average speed over 100 generated samples. This approach offers a more robust and realistic estimate of the model's operational efficiency in real-world applications.

4. Results and Discussion

4.1. Main Results

T2I Generation. In Tab. 1, we benchmark our pruned SD models against other T2I models. Relative to the manual

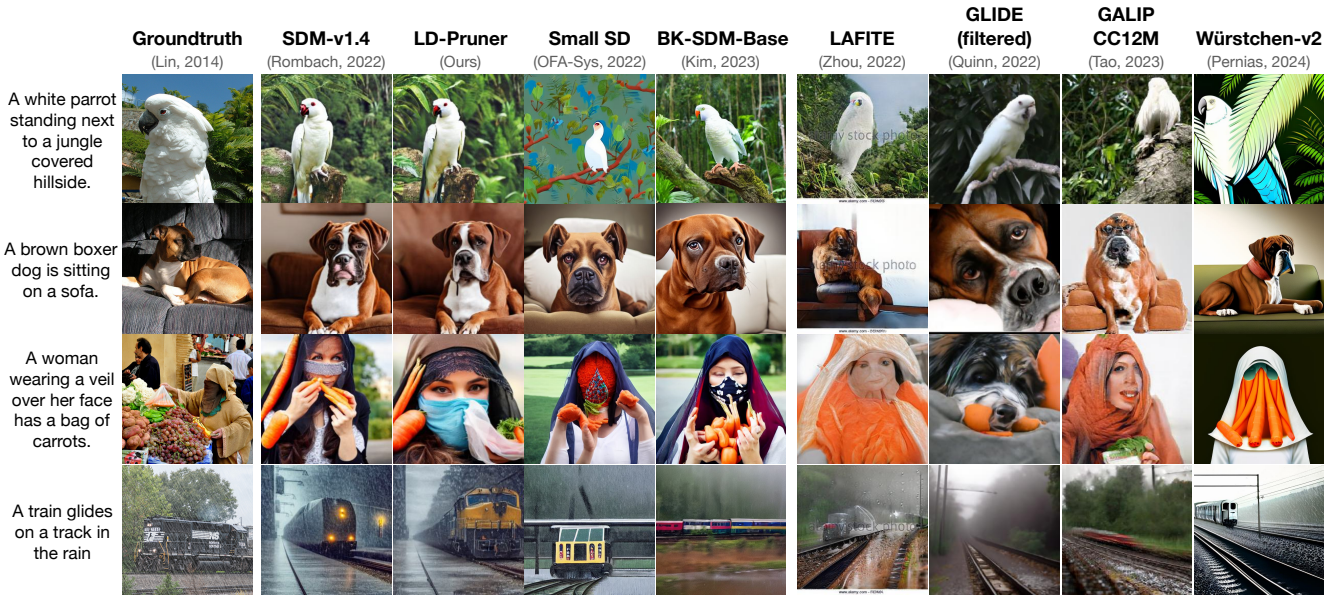


Figure 3. Qualitative comparison on zero-shot MS-COCO benchmark on T2I. The results of previous studies were obtained with their official released models.

architectural pruning of SD-v1.4 [31], our model surpasses those finetuned with large data and does so with a remarkable reduction in the number of training data samples — 1040 times fewer than used in OFA-Sys [25], a similar approach to that of Kim et al. [16]. A qualitative comparison can be found in Fig. 3. Furthermore, our pruned models demonstrate competitive performance when juxtaposed with other architecture, including those based on autoregression [5, 6, 9, 29], GANs [35, 37], and diffusion methods [26, 28, 30, 31]. Lastly, Fig. 4 visualizes the modified operators and their importance rankings. Notably, a significant portion of these operators is located near the model's output, suggesting an over-parametrization in this region.

Unconditional Image Generation. We show the evolution of the FID during training in Fig. 5. We modified 31 operators resulting in a 23.47% speedup and 39 operators for a 28.21% speedup. In both cases, the compressed model converges rapidly, reaching a minimum FID of 15.03 after 46k iterations and 15.71 after 50k iterations, respectively. For comparison, the original model reaches an FID of 13.84 after 410k iterations. Notably, with just 20k iterations and 20 modified operators (corresponding to an 18.23% speedup), we surpass the FID performance of the original model as shown in Fig. 7, achieving an FID of 13.72. This underlines the efficiency of our approach in both speed and performance dimensions.

Unconditional Audio Generation. Tab. 2 showcases the comparison between our compressed model and the baseline AudioDiffusion model. With 90 operators modified, we achieve a 19.2% speedup and a post-finetuning

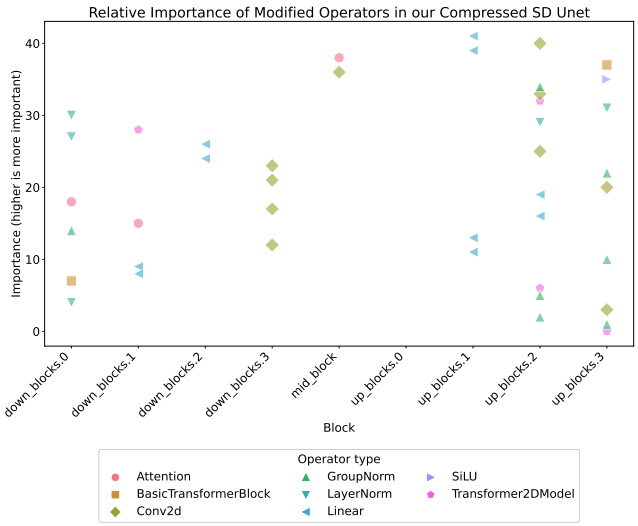


Figure 4. Type and relative importance of the modified operators in each block of our compressed SD.

FAD of 2.5 (+0.2). These results underscore LD-Pruner's ability to compress independently of the task at hand. It is worth mentioning that the audio output from the pruned model sounds the same as the audio of the original model. This can be quantified by evaluating the FAD between 5000 samples of both models, resulting in a score of 0.05.

4.2. Scoring Metric Composition

An essential aspect of our methodology is the scoring metric, as the pruning quality depends on it. Therefore, be-

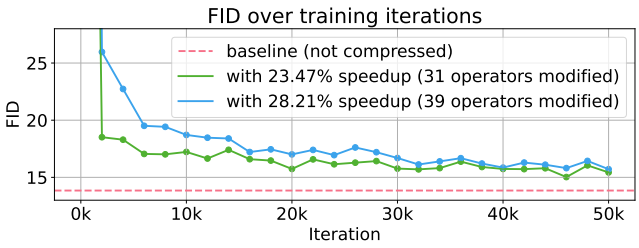


Figure 5. Evolution of the FID during the training process for the UIG task on the CelebA-HQ 256×256 dataset, for two different compression ratios.

Pruning	Finetuned	FAD ↓	# Params	Speedup
None	-	2.3	163.1M	0%
LD-Pruner	✗ (reset Unet)	13.4	85.6M	19.2%
	✗	8.7		
	✓	**2.0**		

Table 2. Compression performance on UAG task with AudioDiffusion. When finetuning, we proceed for 12k steps.

fore settling on the chosen method, we carried out extensive experimentation with alternative ways of incorporating the statistical measures.

The intention behind this metric is to award higher scores to the latent variables that best preserve output quality. A preliminary visual examination of the image outputs from the pruned models, prior to retraining, provides useful insight into the effectiveness of different approaches. This is evidenced in Fig. 6, where we illustrate the impact of various methods—summation, multiplication, average only, and standard deviation only—on the visual characteristics of pruned models. Our observations reveal that the 'average only' method allows for more pruning before degradation to noise, whereas the 'standard deviation only' method tends to preserve sharper features under low compression. When these methods are combined via summation or multiplication, we observe a balance of these attributes, resulting in outcomes that blend both qualities.

A more detailed examination, which compares the FID scores of pruned models post-finetuning for the different formulae, is presented in Fig. 7 for further insight. Notably, a direct comparison with Fig. 6 shows a correlation between the observed degradation to noise in the image before retraining and the decline in FID scores after retraining. This connection tends to support the idea that the visual analysis of models before retraining can serve as an early indicator of post-retraining performance. Interestingly, the 'sum' formula consistently achieves the best or near-best FID scores when modifying up to 40 operators. Yet, as the number of modified operators expands significantly, this positive trend does not persist. We speculate that this outcome is due to the compression process neglecting potential inter-dependencies between operators. Thus, as the number of modified operators escalates, so does the probability of inadvertently eliminating all instances of specific, possibly crucial, information. This particular challenge signals a potential direction for improvement in future research.

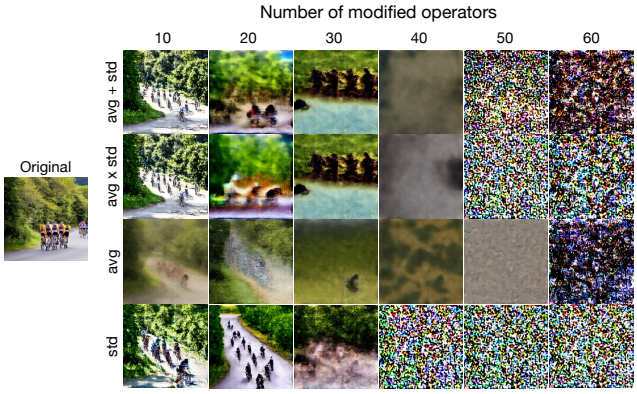


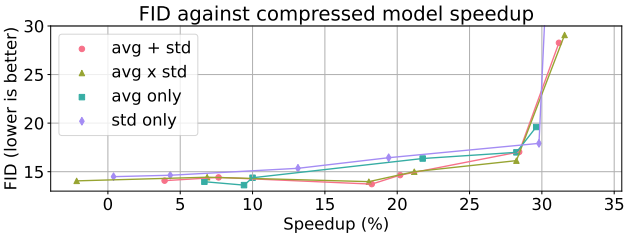
Figure 6. Qualitative comparison of the impact of various combination methods for average and standard deviation in our proposed scoring metric, with SD. The results are without finetuning. Prompt: "group of cyclists racing in a scenic countryside". More examples can be found in the Supplementary Materials.

4.3. Importance of Preserving the Weights

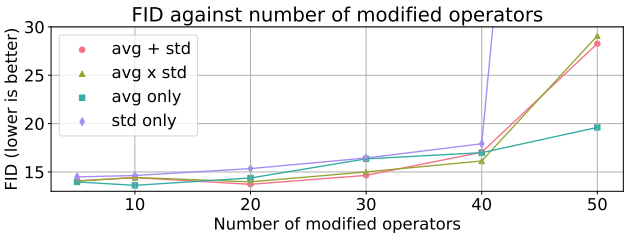
Training LDMs typically requires extensive computational resources and large datasets, making the process time-consuming and costly. To address this challenge, our proposed method focuses on weight preservation during pruning. In this subsection, we underscore the significance of this approach and provide empirical evidence of its benefits.

Our experiments, documented in Tab. 3, compare the performance of models trained from scratch to those with preserved weights, under equivalent compression and training conditions. Performance is quantified using the FID score, which provides a measure of the distance between the model-generated and real data distributions.

The results exhibit a pronounced advantage for models with preserved weights, consistently reporting lower FID scores. This signifies a closer match to the actual data distribution, hence superior image generation quality. Notably, the model with preserved weights comes close to the initial model's performance after merely $20,000$ iterations, whereas the model trained from scratch fails to match the FID score of the model with preserved weights at iteration 0, even after $50,000$ iterations. This stark performance disparity underscores the pivotal role weight preservation plays in the model training process.



(a) Comparison of the FID of the finetuned compressed model against the speed improvement over the original model



(b) Comparison of the FID of the finetuned compressed model against the number of modified operators

Figure 7. Quantitative comparison of the impact of various combination methods for average and standard deviation in our proposed scoring metric, with UIG. The FID is measured after 20k iterations of finetuning.

Number of train steps	0	4k	12k	20k	50k
From Scratch	389.96	344.27	312.86	293.07	210.43
With Preserved Weight	145.19	18.29	16.65	15.74	**15.43**

Table 3. FID scores for our compressed model (31 operators modified) trained from scratch and with preserved pre-training weights, for UIG on CelebA-HQ 256×256. In both case, the exact same training is applied. The FID for the original model is 13.85.

4.4. Speed-Performance Trade-off

The act of model compression inherently introduces a trade-off between computational speed and performance. As we increase the compression rate, the model's performance, measured by FID, gradually deteriorates. This degradation, however, is not linear but exhibits a threshold-like behavior, beyond which the performance sharply deteriorates.

This phenomenon is illustrated in Figure 7a, which demonstrates the relationship between the FID score and the percentage of speedup achieved through compression, relative to the initial model in the context of UIG. As the figure shows, the FID score remains relatively steady, hovering around 15, for compression rates up to around 30%. Beyond this point, the FID score abruptly escalates, indicating a significant drop in the quality of generated images. We observed a similar trend for other tasks, with a thresholds at 42 modified operators for T2I and 90 for UAG.

4.5. Limitations

Despite the strengths of our method, there are some limitations to acknowledge. Firstly, our approach does not extend to pruning the decoder part of the model, as it operates after the latent space. Consequently, the method is best suited for LDMs, where most computational expenses occur in the U-Net due to the recursive nature of the process. Secondly, the current approach does not account for dependencies between operators, potentially leading to pruning decisions that are not optimal. These limitations present valuable areas for further improvements to our method.

5. Conclusion

In this paper, we have presented a novel architecture pruning algorithm for LDMs that is task-agnostic and leverages the latent space to guide the pruning process. An integral part of our approach is the introduction of a new scoring metric that enables the direct comparison of latent representations, providing a robust, quantifiable measure for pruning decisions. Our approach addresses unique challenges posed by generative models, transcending task-specific limitations of existing strategies, and leads to compact models with faster inference speed and fewer parameters, without substantially sacrificing performance.

References

[1] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. *arXiv*, 2023. 2

[2] Thibault Castells and Seul-Ki Yeom. Automatic neural network pruning that efficiently preserves the model accuracy. In *2nd International Workshop on Practical Deep Learning in the Wild*, 2023. 2

[3] CompVis. Stable diffusion training. https://huggingface.co/CompVis/stable-diffusion-v1-4, 2023. 2

[4] Robert Dargavel Smith. Audiodiffusion. https://huggingface.co/teticio/latent-audio-diffusion-256, 2022. 5

[5] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 5, 6

[6] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5, 6

[7] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[8] Jared Fernandez, Jacob Kahn, Clara Na, Yonatan Bisk, and

Emma Strubell. The framework tax: Disparities between inference efficiency in research and deployment, 2023. 5

[9] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision (ECCV)*, 2022. 5, 6

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 1

[11] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*, 2023. 1

[12] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation, 2018. 5

[15] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2019. 5

[16] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. *arXiv*, 2023. 1, 5, 6

[17] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[18] Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, and Yingyan Lin (Celine). Hw-nas-bench: Hardware-aware neural architecture search benchmark. In *International Conference on Learning Representations (ICLR)*, 2021. 5

[19] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[20] Xiuyu Li, Long Lian, Yijiang Liu, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. *arXiv preprint arXiv:2302.04304*, 2023. 1

[21] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 5

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 5

[23] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 1

[24] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[25] OFA-Sys. Small stable diffusion. https://huggingface.co/OFA-Sys/small-stable-diffusion-v0, 2022. 5, 6

[26] Pablo Pernias, Dominic Rampas, Mats Leon Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024. 5, 6

[27] Qualcomm. Aimet. https://github.com/quic/aimet, 2020. 2

[28] Alexander Quinn 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 *International Conference on Machine Learning (ICML)*, 2022. 5, 6

[29] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021. 5, 6

[30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 5, 6

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 5, 6

[32] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar 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 (NeurIPS)*, 2022. 2

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 5

[34] Christoph Schuhmann and Romain Beaumont. Laion-aesthetics, 2023. 5

[35] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. Galip: Generative adversarial clips for text-to-image synthesis. 2023. 5, 6

[36] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent

point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[37] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6