SlimSAM:
0.1% Data Makes Segment Anything Slim

Zigeng Chen, Gongfan Fang, Xinyin Ma, Xinchao Wang[*]
National University of Singapore
zigeng99@u.nus.edu, xinchao@nus.edu.sg

Abstract

Current approaches for compressing the Segment Anything Model (SAM) yield commendable results, yet necessitate extensive data to train a new network from scratch. Employing conventional pruning techniques can remarkably reduce data requirements but would suffer from a degradation in performance. To address this challenging trade-off, we introduce SlimSAM, a novel data-efficient SAM compression method that achieves superior performance with extremely less training data. The essence of SlimSAM is encapsulated in the alternate slimming framework which effectively enhances knowledge inheritance under severely limited training data availability and exceptional pruning ratio. Diverging from prior techniques, our framework progressively compresses the model by alternately pruning and distilling distinct, decoupled sub-structures. Disturbed Taylor pruning is also proposed to address the misalignment between the pruning objective and training target, thereby boosting the post-distillation after pruning. SlimSAM yields significant performance improvements while demanding **over 10 times less** training data than any other existing compression methods. Even when compared to the original SAM, SlimSAM achieves approaching performance while reducing parameter counts to merely **1.4% (9.1M)**, MACs to **0.8% (23G)**, and requiring only **0.1% (10k)** of the SAM training data. Code is available at https://github.com/czg1225/SlimSAM

1 Introduction

Segment Anything Model (SAM) [25] has attracted considerable attention from the community since its inception. A plethora of studies [48, 19, 34, 43, 31, 2, 59, 55, 18, 53, 35] have achieved substantial progress by incorporating SAM as a fundamental component. Nevertheless, despite its remarkable performance, SAM's substantial model size and high computational demands render it inadequate for practical applications on resource-constrained devices. This limitation consequently hinders the advancement and broader application of SAM-based models.

To mitigate these constraints, many efforts [29, 62, 60, 50, 63, 45] have been made to effectively compress SAM. Without exception, these endeavors opt to replace the originally heavyweight image encoder with a lightweight and efficient architecture. This invariably entails training a new network from scratch. With regard to scratch training, an unavoidable challenging trade-off arises between training costs and model performance. Existing methods all inevitably compromise performance when training with very limited data.

The crux of the above issue is their inability to fully exploit the capability of pre-trained SAM. To overcome the high training data demands by reusing the robust prior knowledge of pre-trained SAM, a straightforward strategy involves the application of pruning techniques [38, 14, 3, 52, 8] to directly

[*] Correspoding Author

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

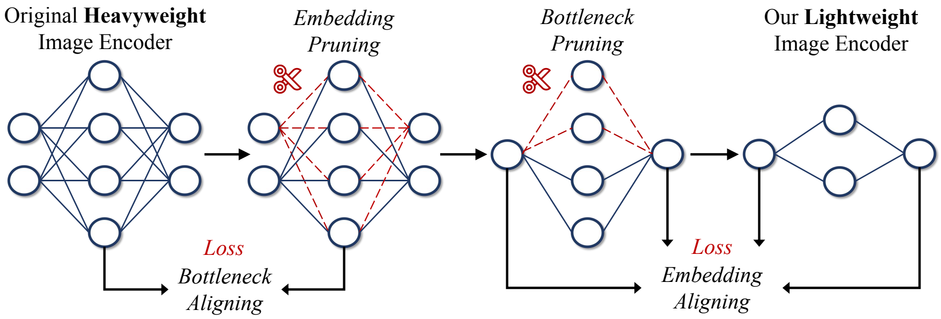


Figure 1: A simple overall diagram of the proposed alternate slimming process.

compress the sizable SAM by removing redundant parameters from the network and fine-tuning the streamlined model with a minimal dataset [16, 36, 10, 30]. Nevertheless, following this conventional procedure leads to unexpected steep performance degradation, particularly when the pruning ratio is set aggressively high and the available data is extremely scarce.

In response to the challenges outlined above, we present SlimSAM, a data-efficient method for SAM compression. Initiating with a standard pruning-finetuning workflow, we gradually "modernize" the compression procedure by introducing our novel designs customized for severely limited data availability and the intricate coupled structure of SAM, culminating in exceptional efficacy while requiring minimal training data. Central to the method are our pioneering contributions: the alternate slimming framework and the disturbed Taylor pruning.

The alternate slimming framework, presented in Figure 1, boosts performance by minimizing divergence from the original model and enabling the intermediate feature alignment via consistent dimensionality. Diverging from prior methods, it alternates between pruning and distillation within decoupled model components. The process begins by targeting the embedding dimensions for pruning and aligning the consistent bottleneck dimensions for distillation. It then shifts focus to pruning the bottleneck dimensions in ViTs [6], aligning the unchanged embedding dimensions for distillation. Observing the misalignment between the pruning object and the distillation target impedes the efficacy of compression, we introduce a novel label-free importance estimation criterion called disturbed Taylor importance to address this misalignment effectively, thereby enhancing the recovery process and obviating the need for labeled data.

Comprehensive assessments across performance metrics, efficiency, and training data requirements reveal that SlimSAM markedly enhances compression performance, concurrently achieving superior lightweight and efficiency with markedly reduced training data requirements. Notably, our entire compression can be completed using only 10k un-labeled images on a single Titan RTX GPU.

In summary, our contribution is a data-efficient SAM compression method called SlimSAM, which effectively repurposes pre-trained SAMs without the necessity for extensive retraining. This is achieved through a novel modernized pruning-distillation procedure. By proposing the alternate slimming framework and introducing the concept of disturbed Taylor importance, we realize greatly enhanced knowledge retention in data-limited situations. When compared to the original SAM-H, SlimSAM achieves approaching performance while reducing the parameter counts to **1.4% (9.1M)**, MACs to **0.8% (23G)**, and requiring mere **0.1% (10k)** of the training data. Extensive experiments demonstrate that our method realizes significant superior performance while utilizing **over 10 times less** training data when compared to any other compression methods.

2 Related Works

Model Pruning. Due to the inherent parameter redundancy in deep neural networks [13], model pruning [16, 14, 30, 3, 27, 33, 52, 36, 10] has proved to be an effective approach for accelerating and compressing models. Pruning techniques can be generally classified into two main categories: structural pruning [30, 27, 56, 8, 4, 56, 7, 9] and unstructured pruning [5, 26, 40, 42, 12]. Structural pruning is focused on eliminating parameter groups based on predefined criteria, while unstructured pruning involves the removal of individual weights, typically requiring hardware support.

Efficient Learning. *Efficient Learning* refers to a range of techniques [54, 57, 20, 21, 58, 11, 22, 28] aimed at reducing the training costs of deep models while maintaining performance. *Knowledge Distillation* (KD) [17] is a prominent method under this category, where knowledge is transferred from a larger, powerful teacher model to a smaller, more efficient student model. This approach leverages soft targets and a temperature parameter to enable the student model to learn more effectively. KD [46, 44, 61, 1, 51, 32, 47, 37, 39] has proven to be an effective strategy for model compression, making it highly applicable in scenarios requiring resource-efficient deployment.

SAM Compression. The formidable model size and computational complexity of SAM pose challenges for edge deployment, prompting an extensive array of research focused on devising compression techniques for SAM to enhance its applicability. Notably, FastSAM [62] replaces SAM's extensive ViT-based architecture with the efficient CNN-based YOLOv8-seg [23] model, while MobileSAM [60] adopts the lightweight Tinyvit [49] to replace the image encoder and employs knowledge distillation from the original encoder. EdgeSAM [63] introduces the prompt-in-the-loop knowledge distillation to accurately capture the intricate dynamics between user input and mask generation. EfficientSAM [50] innovatively adapts MAE [15] framework to obtain efficient image encoders for segment anything model but requires extensive training data even more than the SA-1B dataset. However, the above approaches all inevitably suffer from scratch training, resulting in unsatisfactory performance when training data is limited.

Remark. The application of common pruning and KD methods falls short in achieving superior performance due to the unique challenges presented by limited training data and SAM's coupled structure. To enhance performance, we propose an alternate slimming framework to minimize divergence from the original model and enable the intermediate feature alignment by consistent dimensionality. We also propose disturbed Taylor pruning to address the misalignment between pruning objectives and training targets. In contrast to other SAM compression methods, our SlimSAM achieves superior compression performance while significantly incurring lower training data requirements.

3 Methods

Our paramount objective is to achieve substantial compression of the large image encoder while minimizing performance degradation in scenarios characterized by severe data limitations. To navigate the challenging trade-off between maintaining remarkable performance and the necessity for copious training data, we adopt a strategy of directly inheriting the core weights from the original SAM. This approach capitalizes on SAM's robust prior knowledge, derived from 11 million images. Adhering to this foundational principle, we begin with a standard workflow: initial pruning of the model followed by refinement through post-distillation.

3.1 Identifying SAM Redundancy

The initial phase is dedicated to the estimation of the importance of each parameter, determining the non-essential and redundant parameters of the image encoder to be pruned. To fulfill this objective, we endeavor to estimate the importance of a parameter through the quantification of prediction errors engendered by its removal [38]. Given a labeled dataset with N image pairs $\{x_i, y_i\}_{i=1}^{N}$ and a model $\mathcal{F}$ with M parameters $W = \{wi\}_{i=1}^{M}$. The output of the original model can be defined as $t_i = \mathcal{F}_W(x_i)$. Our objective is to identify the parameters that yield the minimum deviation in the loss. Specifically, the importance of a parameter w_i, can be defined as:

$$I_{w_i} = |\Delta\mathcal{L}(x_i, y_i)| = |\mathcal{L}_{w_i}(x_i, y_i) - \mathcal{L}_{w_i=0}(x_i, y_i)|\,, \tag{1}$$

where $\mathcal{L}(x_i, y_i)$ is the loss between the model output and the label y_i when input data is x_i. We can approximate $\mathcal{L}_{w_i=0}$ in the vicinity of w_i by its first-order Taylor expansion:

$$\mathcal{L}_{w_i=0}(x_i, y_i) = \mathcal{L}_{w_i}(x_i, y_i) - \frac{\partial\mathcal{L}(x_i, y_i)}{\partial w_i} w_i + \mathcal{R}_1(w_i = 0). \tag{2}$$

Substituting equation 2 into equation 1, we can approximate the parameter importance as:

$$I_{w_i} \approx \left|\mathcal{L}_{w_i=0}(x_i, y_i) - \mathcal{L}_{w_i=0}(x_i, y_i) + \frac{\partial\mathcal{L}(x_i, y_i)}{\partial w_i} w_i\right| = \left|\frac{\partial\mathcal{L}(x_i, y_i)}{\partial w_i} w_i\right|. \tag{3}$$

However, there exist two distinct limitations associated with the above Taylor importance estimation when pruning the image encoder of SAM. Firstly, the accuracy of Taylor importance relies heavily on the availability of sufficiently accurate hard labels y_i. Unfortunately, due to the intricate nature of jointly optimizing the image encoder and combined decoder [60], the post-distillation process necessitates performing on the image embedding t_i, resulting in the utilization of soft labels exclusively. Secondly, a concern arises regarding the consistency of loss functions when employing Taylor importance estimation for SAM pruning. The importance estimation strategy's primary objective is to identify parameters w_i that minimize the hard label discrepancy $|\Delta\mathcal{L}(x_i, y_i)|$. In contrast, the goal of the distillation-based recovery process is to minimize the soft label loss $|\Delta\mathcal{L}(x_i, t_i)|$. This misalignment in optimization objectives potentially impedes the efficacy of the distillation process. The experimental results in Section 5 also strongly prove our conclusion.

Disturbed Taylor importance. To address the unique limitations associated with Taylor importance estimation, we introduce an extremely simple yet effective solution known as disturbed Taylor importance. Given the absence of hard labels and the incongruity of loss functions, a logical approach is to identify parameters w_i that minimize the soft label divergence $|\Delta\mathcal{L}(x_i, t_i)|$. However, the gradients $\frac{\partial \mathcal{L}(x_i,t_i)}{\partial w_i}$ resulting from applying the loss between encoder's outputs t_i are consistently zero. Consequently, we calculate gradients based on the loss function between the original image embedding t_i and disturbed image embedding $t_i + \mathcal{N}(\mu, \sigma^2)$, where $\mathcal{N}$ is Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 0.01$. As the expectation $E(t_i + \mathcal{N}) = t_i$, when the batch size is large enough, the importance of a parameter w_i can be approximated as:

$$\begin{aligned} I_{w_i} &= |\Delta\mathcal{L}(x_i, t_i)| \approx |\Delta\mathcal{L}(x_i, t_i + \mathcal{N})| \\ &= |\mathcal{L}_{w_i}(x_i, t_i + \mathcal{N}) - \mathcal{L}_{w_i=0}(x_i, t_i + \mathcal{N})| \\ &\approx \left| \frac{\partial \mathcal{L}(x_i, t_i + \mathcal{N})}{\partial w_i} w_i \right| . \end{aligned} \tag{4}$$

As the generated gradients $\frac{\partial \mathcal{L}(x_i,t_i+\mathcal{N})}{\partial w_i} \neq 0$, the importance can be estimated.

Remark. Leveraging our disturbed Taylor importance, the pruning objective is seamlessly aligned with the optimization target of subsequent distillation. Compared to previous pruning techniques, it results in a 0.85% MIoU enhancement when the pruning ratio reaches 77% and a 0.60% MIoU improvement when the pruning ratio is set at 50%. Moreover, the adoption of disturbed Taylor importance transforms the entire compression workflow into a convenient label-free framework without incurring additional computational costs.

3.2 Alternate Slimming.

After estimating the weights' importance, our approach advances to implementing channel-wise structural pruning on the extensive image encoder, followed by distillation-based model finetuning. To attain an unprecedentedly high compression rate, the pruning ratio in this study is necessitated to be set significantly higher than in typical scenarios. With the pruning ratio exceeding 75%, we observe a marked performance degradation between the pruned model and its original counterpart, a consequence of employing the conventional single-step pruning technique. Additionally, the extremely constrained data availability also poses unique challenges to distillation efficacy. Employing merely 0.1% of the SA-1B dataset (10k images) for post-distillation underscores a significant challenge in recuperating satisfactory performance for the pruned model.

To address identified challenges, we introduce an innovative alternate slimming framework, anchored by two principles: reducing the divergence between the original and pruned models, and enhancing post-distillation efficacy.

Our framework decomposes the model into two separate sub-structures: embedding (output dimensions of each block) and bottleneck (intermediate features of each block). By sequentially pruning and restoring either sub-structure, we achieve a smoother compression loss, preventing the steep performance degradation typically associated with extreme pruning ratios. To improve post-distillation, we exploit the hidden state information of the original model. Due to the structural resemblance between the pruned and original models, using intermediate hidden states for supervision facilitates superior knowledge transfer. Traditional pruning workflow struggles with dimensionality inconsistency, complicating hidden state supervision. Our method, by partitioning the model into

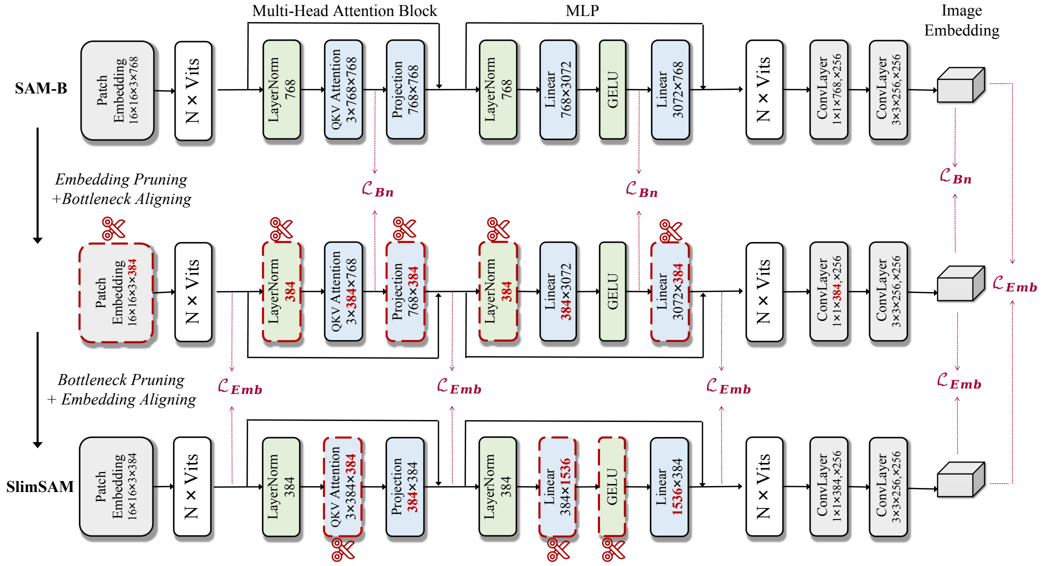


Figure 2: The provided figure depicts our alternate slimming process with a 50% pruning ratio on SAM-B. We utilize structural pruning at the channel-wise group level to compress SAM's image encoder, coupled with knowledge distillation from intermediate layers to restore the pruned encoder. The red numbers highlight the pruned dimensions at each pruning step.

sub-structures, circumvents this issue. Whether pruning embedding or bottleneck dimensions, the intact remaining dimensions enable alignment through loss backpropagation. The effectiveness of this feature alignment, especially in data-scarce scenarios, highlights our framework's efficacy.

An overview of the alternate slimming framework is detailed in Figure 2. Given the Vit-based image encoder with k blocks, the output and intermediate features of each block within the encoder are denoted as $E = \{e_i\}_{i=1}^{k}$ and $H = \{h_i\}_{i=1}^{k}$. Specifically, for *Multi-Head Attention Blocks* (MHABs), the intermediate feature refers to the concatenated QKV features, while for the MLPs, it refers to the hidden features between two linear layers. The final output image embedding is represented as t. The original encoder is referred to as v_0, while the pruned encoders after embedding pruning and bottleneck pruning are denoted as v_1 and v_2, respectively. The alternate slimming process can be described as the following progressive procedure: embedding pruning, bottleneck aligning, bottleneck pruning, and embedding aligning.

Embedding Pruning. The embedding dimension significantly impacts the encoder's performance as it determines the width of features extracted within the encoder. To begin with, we prune the embedding dimensions $\mathcal{D}(E)$ while keeping the bottleneck dimensions $\mathcal{D}(H)$ constant. The presence of residual connections necessitates the preservation of uniformity in the pruned embedding dimensions $\mathcal{D}(\{e_i\}_{i=1}^{K})$ across all blocks. Consequently, we employed uniform local pruning.

Bottleneck Aligning. In the context of incremental knowledge recovery, the pruned encoder learns from the original encoder's output t_{v_0} and aligns with its dimensionality-consistent bottleneck features H_{v_0} in each block. The distillation loss function for bottleneck aligning is a combination of bottleneck feature loss and final image embedding loss:

$$\mathcal{L}_{Bn} = \alpha \cdot \mathcal{L}_{MSE}(H_{v_0}, H_{v_1}) + (1-\alpha) \cdot \mathcal{L}_{MSE}(t_{v_0}, t_{v_1}), \tag{5}$$

where $\mathcal{L}_{MSE}(.,.)$ is mean-squared error, the dynamic weight α of nth epoch is defined as:

$$\alpha = \left\{ \begin{array}{ll} 0.5 & n < N \\ 0 & n >= N \end{array} \right. . \tag{6}$$

We set $N = 10$ for bottleneck aligning.

Bottleneck Pruning. Following the pruning of the embedding dimension $\mathcal{D}(E)$ and its coupled structures, we exclusively focus on pruning the bottleneck dimension. As the dimension of intermediate features $\mathcal{D}(\{h_i\}_{i=1}^{K})$ in each block are entirely decoupled, we can systematically apply dimension pruning at various ratios for each block while maintaining the predetermined overall pruning ratio. This approach involves utilizing a global ranking of importance scores to conduct global structural pruning.

Embedding Aligning. The pruned encoder v_2 will learn from the embeddings E_{v_1} and final image embedding T_{v_1} from the pruned encoder v_1 to expedite knowledge recovery. Simultaneously, it also computes loss functions based on the final image embedding t_{v_0} from the original encoder v_0 to enhance the precision of knowledge recovery. The total loss function for embedding aligning is defined as:

$$\begin{aligned}\mathcal{L}_{Emb} = \alpha \cdot (\mathcal{L}_{MSE}(E_{v_1}, E_{v_2}) + \mathcal{L}_{MSE}(t_{v_1}, t_{v_2})) \\ +(1-\alpha) \cdot \mathcal{L}_{MSE}(t_{v_0}, t_{v_2}),\end{aligned} \tag{7}$$

where the dynamic weight α of nth epoch is defined as:

$$\alpha = \begin{cases} \frac{N-n-1}{N} & n < N \\ 0 & n >= N \end{cases}. \tag{8}$$

The dynamic weight α will progressively diminish to zero as the distillation process unfolds. This transition in the learning objective of distillation gradually shifts from v_1 to v_0 contributing to a smoother knowledge recovery. We also set $N = 10$ for embedding aligning.

Remark. The implementation of alternate slimming on decoupled sub-structures significantly reduces the disruption to the original model, particularly when the pruning ratio is quite high. This strategy also preserves consistent dimensionality, enabling effective intermediate feature distillation, which is especially beneficial in data-scarce conditions. Consequently, in comparison to the previous pruning and distillation methods, our alternate slimming achieves a 3.40% and 0.92% increase in MIoU when the pruning ratios achieve 77% and 50%.

4 Experiments

4.1 Experimental Settings

Implementation Details. Our SlimSAM has been implemented in PyTorch [41] and trained on a single Nvidia Titan RTX GPU using only 0.1% (10,000 images) of the SA-1B [25] dataset. The base model of our framework is SAM-B [25]. The model's parameters were optimized through the ADAM [24] algorithm with a batch size of 4. Training settings for both bottleneck aligning and embedding aligning are identical. The pruned models undergo distillation with an initial learning rate of $1e^{-4}$, which will be reduced by half if validation performance does not improve for 4 consecutive epochs. The total training duration is 40 epochs for SlimSAM-50 (with a 50% pruning ratio) and 80 epochs for SlimSAM-77 (with a 77% pruning ratio). We exclusively compressed the image encoder while retaining SAM's original prompt encoder and mask decoder.

Evaluation Details. To ensure a fair quantitative evaluation of the compressed SAM models, we compute MIoU between the masks predicted by the model and the ground truth masks of the SA-1B dataset. We use the most challenging single-point prompts given in annotations for experiments. The results using box prompts are also reported in our Appendix. For efficiency evaluation, we provide information on parameter counts and MACs. Additionally, we present details about training data, training iteration and training GPUs for evaluating the training cost. Qualitative comparison of results obtained using point prompts, box prompts, and segment-everything prompts are also shown in the following section.

4.2 Comparision and Analysis

Comparing with existing SAM compression methods. As depicted in Table 1, we conducted a comprehensive comparison encompassing performance, efficiency, and training costs with other SOTA methods. Our SlimSAM-50 and SlimSAM-77 models achieve a remarkable parameter reduction to only 4.0% (26M) and 1.4% (9.1M)of the original count, while also significantly lowering computational demands to just 3.5% (98G) and 0.8% (23G) MACs, all while maintaining performance levels comparable to the original SAM-H. In contrast to other compressed models, our approach yields substantial performance enhancements while simultaneously achieving greater lightweight

Table 1: Comparing with other existing SAM compression methods on SA-1B dataset. We report parameter counts, MACs, training costs, and *Mean Intersection over Union* (MIoU) for a comprehensive and fair comparison.

Method	Params↓	MACs↓	TrainSet	BatchSize	GPUs	Iters	MIoU↑
SAM-H [25]	641M	2736G	11M(100%)	256	256	90k	78.30%
SAM-L [25]	312M	1315G	11M(100%)	128	128	180k	77.67%
SAM-B [25]	93M	372G	11M(100%)	128	128	180k	73.37%
FastSAM-s [62]	11M	37G	220k(2%)	32	8	625K	30.72%
FastSAM-x [62]	68M	330G	220k(2%)	32	8	625K	35.41%
MobileSAM [60]	9.8M	40G	100k(1%)	8	1	100k	62.73%
EfficientSAM-t [50]	10M	28G	12.2M(110%)	128	64	450k	69.42%
EfficientSAM-s [50]	26M	94G	12.2M(110%)	128	64	450k	71.19%
EdgeSAM [63]	9.6M	23G	100k(1%)	64	8	50k	65.96%
SlimSAM-50(Ours)	26M	98G	**10k(0.1%)**	4	1	100k	**72.33%**
SlimSAM-77(Ours)	**9.1M**	**23G**	**10k(0.1%)**	4	1	200k	67.40%

Table 2: Comparing with other structural pruning methods. 'Ratio' signifies the pruning ratio applied to channel-wise groups. Training costs remain consistent for the same pruning ratio.

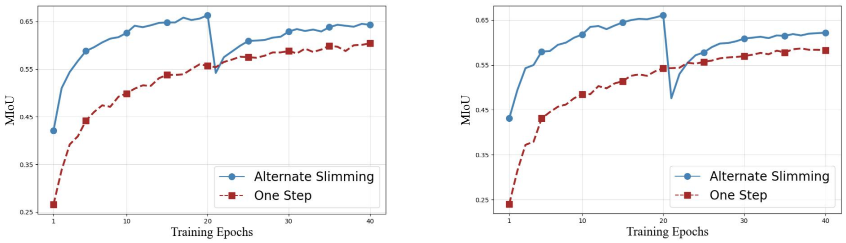
Ratio	Method	Labels	Params↓	MACs↓	MIoU↑
Ratio=0%	SAM-H [25]	✔	641M	2736G	78.30%
	SAM-L [25]	✔	312M	1315G	77.67%
	SAM-B [25]	✔	93M	372G	73.37%
Ratio=50%	Scratch Distillation	✘	26M	98G	1.63%
	Random Pruning	✘			71.03%
	Magnitude Pruning [14]	✘			69.96%
	Hessian Pruning [30]	✔			71.01%
	Taylor Pruning [38]	✔			71.15%
	SlimSAM-50(Ours)	✘			**72.33%**
Ratio=77%	Scratch Distillation	✘	9.1M	23G	1.34%
	Random Pruning	✘			62.58%
	Magnitude Pruning [14]	✘			61.60%
	Hessian Pruning [30]	✔			63.56%
	Taylor Pruning [38]	✔			64.26%
	SlimSAM-77(Ours)	✘			**67.40%**

and efficiency. SlimSAM consistently delivers more accurate and detailed segmentation results across various prompts, preserving SAM's robust segmentation capabilities to the greatest extent. This qualitative superiority over other models is visually evident in Figure 5 and 6. Our approach demonstrates outstanding levels of accuracy and correctness. Most notably, SlimSAM achieves these remarkable outcomes with exceptionally low training data requirements, utilizing merely 0.1% (10k) images of the SA-1B dataset. This represents a significant reduction in data dependency, requiring 10 times less data than both EdgeSAM and MobileSAM, and 1,100 times less data than EfficientSAM.

Comparing with other structural pruning methods. Having demonstrated structural pruning's efficacy for SAM compression, we established a benchmark for evaluating various pruning methods. SlimSAM is compared with four commonly used pruning methods: random pruning, magnitude pruning, Taylor pruning, and Hessian pruning, each employing different criteria for pruning. Additionally, we conducted comparisons with scratch-distilled models, which are randomly initialized networks sharing the same architecture as the pruned models. To ensure a completely equitable comparison, models with the same pruning ratios were subjected to identical training settings. Table 2 showcases our method's consistent superiority over other structural pruning techniques, particularly at higher pruning ratios. SlimSAM-50 and SlimSAM-77 outperform existing methods, achieving a minimum 1% and 3% MIoU improvement while incurring the same training cost. It is noteworthy that the performance of scratch distillation is extremely low at such a limited training cost. This further proves the effectiveness of our workflow in preserving knowledge from the original model.

5 Ablation Study and Analysis

We conducted a series of ablation experiments on the SlimSAM-77 model, which features an ambitious 77% pruning ratio. To ensure a fair comparison in the ablation experiments, all evaluated



Disturbed Taylor Importance

Random Importance

Figure 3: Training results on SA-1B with the common one-step method and our alternate slimming framework. Left and right are results with disturbed Taylor importance and random importance.

Table 3: Comparison between disturbed Taylor pruning and original Taylor pruning.

Method	MIoU↑
Taylor Pruning	62.04%
Disturbed Taylor Pruning	**62.31%**
SlimSAM-77 + Taylor	63.63%
SlimSAM-77 + Disturbed Taylor	**64.48%**

Table 4: Effect of distillation from intermediate layers and final output image embeddings.

Step	Distillation Objective	MIoU↑
Step 1	Final Image Embeddings	65.10%
Step 1	+ Bottleneck Features	**66.32%**
Step 2	Final Image Embeddings	63.91%
Step 2	+ Embedding Features	**64.48%**

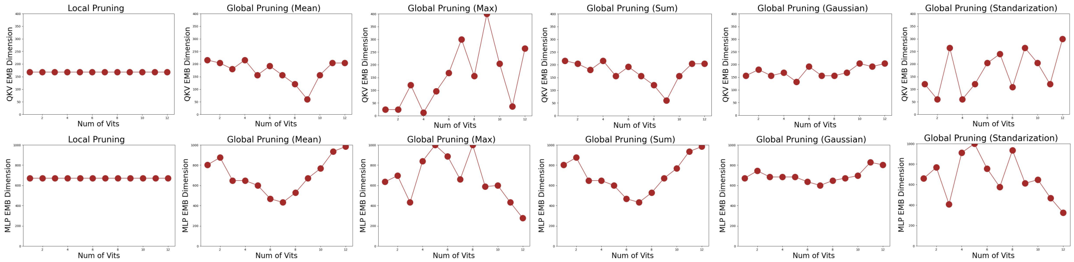


Figure 4: The intermediate dimensions of QVK Attention (top row) and MLP (bottom row) within each ViT after pruning. We present the outcomes of local pruning and global pruning under five distinct normalization methods.

models were trained for 40 epochs on the same 10k images from the SA-1B dataset. We also conduct additional experiments to evaluate the performance of SlimSAM with even less training data.

Disturbed Taylor Pruning. First, we conducted an ablation study to assess the impact of our proposed disturbed Taylor pruning on distillation. This innovative approach aligns the pruning criteria with the optimization objectives of subsequent distillation, resulting in improved performance recovery. As depicted in Table 3, our disturbed Taylor pruning consistently achieves significantly superior performance at the same training cost. For both the common one-step pruning strategy and our alternate slimming strategy, our method demonstrates MIoU improvements of 0.3% and 0.85% over the original Taylor pruning, respectively.

Intermediate Aligning. We also evaluate the effect of incorporating aligning with intermediate layers into the distillation process. As depicted in Table 4, distilling knowledge from intermediate layers leads to significant improvements in training results. Specifically, learning from bottleneck features and final image embeddings results in a 1.22% MIoU improvement for step 1 distillation, compared to learning solely from image embeddings. Similarly, for step 2 distillation, learning from embedding features and final image embeddings achieves a 0.57% MIoU improvement over the case where learning is based solely on image embeddings.

Alternate Slimming. In addition, we conducted experiments to investigate the impact of our alternate slimming framework. Unlike the common one-step pruning strategy, we partition the structural pruning process into two decoupled and progressive steps. In the first step, only the dimensions related to the embedding features are pruned, while in the second stage, only the dimensions related to the bottleneck features are pruned. Following both embedding and bottleneck pruning,

Table 5: Effect of global pruning evaluated under five different normalization approaches.

Method	Normalization	MIoU↑
Local Pruning	—	64.38%
Global Pruning	Mean	63.64%
	Max	64.35%
	Sum	63.55%
	Gaussian	**64.48%**
	Standardization	64.14%

Table 6: Comparision of training results using varied amounts of training data.

Pruning Ratio	Data	Iters	MIoU↑
Ratio=50%	10k	100k	**72.33%**
	5k	100k	71.89%
	2k	100k	69.79%
Ratio=77%	10k	200k	**67.40%**
	5k	200k	64.47%
	2k	200k	61.72%

knowledge distillation with intermediate layer aligning is employed on the pruned model to recover its performance. For a more exhaustive analysis, we present the results obtained using different pruning criteria to assess whether the effectiveness of our method is influenced by importance estimation. As illustrated in Figure 3, our alternative slimming framework yields substantial improvements in MIoU, with gains of 3.9% and 3.5% observed under disturbed Taylor importance estimation and random importance estimation.

Global Pruning vs Local Pruning. Finally, we conducted experiments to evaluate the performance of local pruning and global pruning in bottleneck pruning. Given that the bottleneck dimensions in each block are entirely decoupled, we systematically applied channel-wise group pruning at various ratios for each block while preserving the predefined overall pruning ratio in this step. To obtain a consistent global ranking, we normalized the group importance scores $I_{\mathcal{G}}$ of each layer in five ways: (i) Sum: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{\sum_{i=1}^{K} I_{\mathcal{G}i}}$, (ii) Mean: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{\sum_{i=1}^{K} I_{\mathcal{G}i}/K}$, (iii) Max: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{Max_{i=1}^{K}(I_{\mathcal{G}i})}$, (iv) Standarization: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i} - Max_{i=1}^{K}(I_{\mathcal{G}i})}{Max_{i=1}^{K}(I_{\mathcal{G}i}) - Min_{i=1}^{K}(I_{\mathcal{G}i}) + 1e-8}$, (v) Gaussian: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i} - \sum_{i=1}^{K} I_{\mathcal{G}i}/K}{\sigma_{i=1}^{K}(I_{\mathcal{G}i}) + 1e-8}$. As indicated in Table 5, local pruning ensures consistent performance, whereas global pruning raises the model's upper-performance limit. Global pruning's efficacy is highly dependent on the chosen importance normalization method. For our model, we opted for global pruning with Gaussian normalization, which yielded the best training results. Following global pruning, Figure 4 illustrates the dimensions of bottleneck features (QKV embeddings and MLP hidden embeddings) within each ViT in the image encoder. When applying mean, sum, or Gaussian normalization, the ViTs in the middle exhibit more group redundancy compared to those at the beginning and end. However, the pruned dimensions do not display distinct patterns when utilizing max or standardization normalization. The impact of global pruning becomes more pronounced with an increased number of training iterations. Specifically, when training extends to 80 epochs, the MIoU for global pruning exceeds that of local pruning by approximately 2%.

Even less data. As shown in Table 6, with a pruning ratio of 50%, a reduction in the volume of training data only marginally impacts the model's performance. Notably, even when trained with a limited dataset of just 2,000 images, our SlimSAM-50 model remarkably attains an MIoU of nearly 70%. However, as the pruning ratio is elevated to 77%, a decrease in training data more significantly affects performance. This leads to the inference that although our methodology, which integrates pruning and distillation techniques, mitigates the need for extensive training datasets, the availability of more training data can still enhance model performance, particularly at higher pruning rates.

6 Conclusion

In this paper, we present a novel data-efficient SAM compression method, SlimSAM, which achieves superior performance with minimal training data. The essence of our approach lies in the efficient reuse of pre-trained SAM, avoiding the need for extensive retraining. We introduce key designs to the compression method for enhancing knowledge retention from the original model in data-limited situations. Specifically, our alternate slimming framework carefully prunes and distills decoupled model structures in an alternating fashion, minimizing disruptions to the original model and enabling the intermediate feature alignment by consistent dimensionality. Furthermore, the proposed disturbed Taylor importance estimation rectifies the misalignment between pruning objectives and training targets, thus boosting post-distillation after pruning. SlimSAM convincingly demonstrates its superiority while imposing significantly lower training costs compared to any other existing methods.

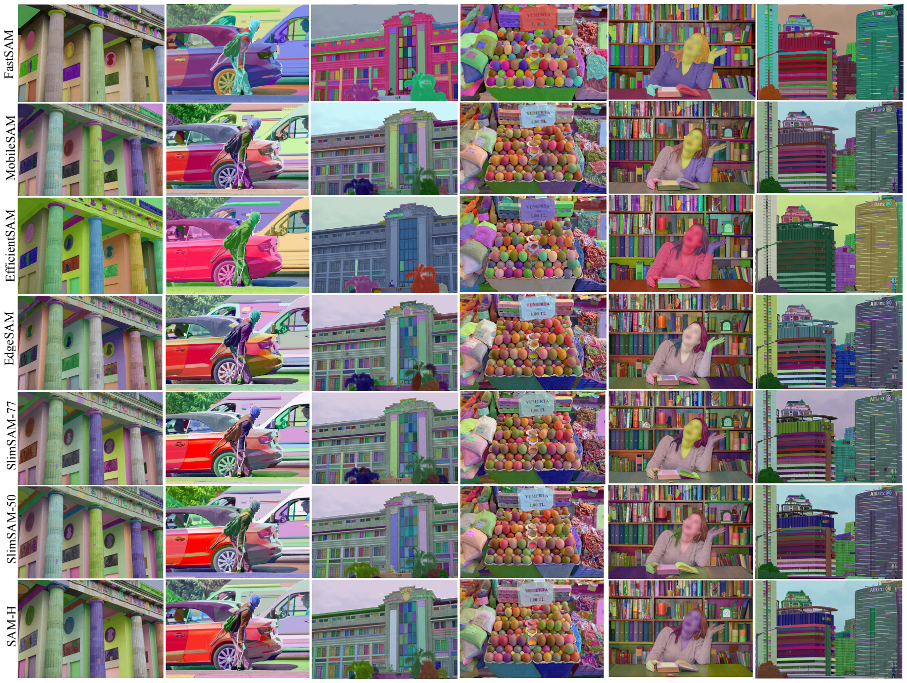


Figure 5: Comparison of segmentation results using segment everything prompts

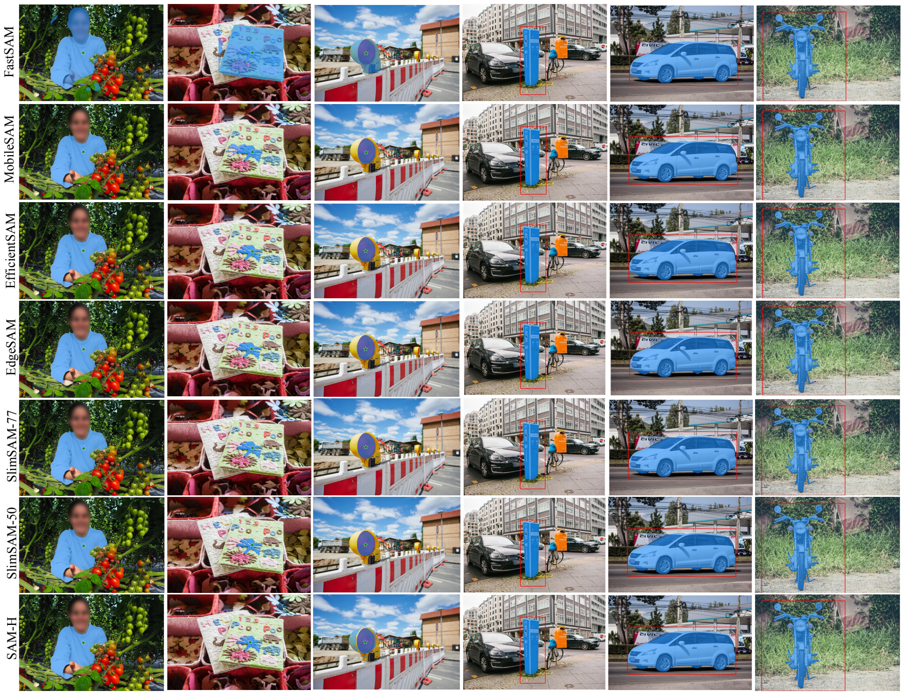


Figure 6: Left 3 columns: segmentation results obtained using point prompts; right 3 columns: segmentation results achieved with box prompts.

Acknowledgement

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award Number: MOE-T2EP20122-0006).

References

[1] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[2] Keyan Chen, Chenyang Liu, Hao Chen, Haotian Zhang, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Rsprompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model. *arXiv preprint arXiv:2306.16269*, 2023.

[3] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1518–1528, 2020.

[4] Xiaohan Ding, Guiguang Ding, Yuchen Guo, and Jungong Han. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4943–4953, 2019.

[5] Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in neural information processing systems*, 30, 2017.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. *arXiv preprint arXiv:2407.04616*, 2024.

[8] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16091–16101, 2023.

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

[10] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. *Advances in neural information processing systems*, 36, 2024.

[11] Gongfan Fang, Kanya Mo, Xinchao Wang, Jie Song, Shitao Bei, Haofei Zhang, and Mingli Song. Up to 100x faster data-free knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6597–6604, 2022.

[12] Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. Maskllm: Learnable semi-structured sparsity for large language models. *arXiv preprint arXiv:2409.17481*, 2024.

[13] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[14] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[16] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1389–1397, 2017.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023.

[19] Yongcheng Jing, Xinchao Wang, and Dacheng Tao. Segment anything in non-euclidean domains: Challenges and opportunities. *arXiv preprint arXiv:2304.11595*, 2023.

[20] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *CVPR*, 2021.

[21] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Meta-aggregator: Learning to aggregate for 1-bit graph neural networks. In *ICCV*, 2021.

[22] Yongcheng Jing, Chongbin Yuan, Li Ju, Yiding Yang, Xinchao Wang, and Dacheng Tao. Deep graph reprogramming. In *CVPR*, 2023.

[23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Yolo by ultralytics, 2023.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[26] Namhoon Lee, Thalaiyasingam Ajanthan, Stephen Gould, and Philip HS Torr. A signal propagation perspective for pruning neural networks at initialization. *arXiv preprint arXiv:1906.06307*, 2019.

[27] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[28] Qi Li, Runpeng Yu, and Xinchao Wang. Encapsulating knowledge in one prompt. *arXiv preprint arXiv:2407.11902*, 2024.

[29] Weicong Liang, Yuhui Yuan, Henghui Ding, Xiao Luo, Weihong Lin, Ding Jia, Zheng Zhang, Chao Zhang, and Han Hu. Expediting large-scale vision transformer for dense prediction without fine-tuning. *Advances in Neural Information Processing Systems*, 35:35462–35477, 2022.

[30] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group fisher pruning for practical network compression. In *International Conference on Machine Learning*, pages 7021–7032. PMLR, 2021.

[31] Songhua Liu, Jingwen Ye, and Xinchao Wang. Any-to-any style transfer: Making picasso and da vinci collaborate. *arXiv e-prints*, pages arXiv–2304, 2023.

[32] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2604–2613, 2019.

[33] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.

[34] Zhihe Lu, Zeyu Xiao, Jiawang Bai, Zhiwei Xiong, and Xinchao Wang. Can sam boost video super-resolution? *arXiv preprint arXiv:2305.06524*, 2023.

[35] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*, 2023.

[36] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36, 2024.

[37] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017.

[38] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019.

[39] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019.

[40] Sejun Park, Jaeho Lee, Sangwoo Mo, and Jinwoo Shin. Lookahead: A far-sighted alternative of magnitude-based pruning. *arXiv preprint arXiv:2002.04809*, 2020.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[42] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389, 2020.

[43] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Anything-3d: Towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*, 2023.

[44] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

[45] Xiaorui Sun, Jun Liu, Heng Tao Shen, Xiaofeng Zhu, and Ping Hu. On efficient variants of segment anything model: A survey. *arXiv preprint arXiv:2410.04960*, 2024.

[46] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*, 2019.

[47] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Kdgan: Knowledge distillation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[48] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.

[49] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European Conference on Computer Vision*, pages 68–85. Springer, 2022.

[50] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv preprint arXiv:2312.00863*, 2023.

[51] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020.

[52] Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18547–18557, 2023.

[53] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

[54] 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*, pages 22552–22562, 2023.

[55] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023.

[56] Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[57] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[58] Ruonan Yu, Songhua Liu, Jingwen Ye, and Xinchao Wang. Teddy: Efficient large-scale dataset distillation via taylor-approximated matching. In *European Conference on Computer Vision*, pages 1–17. Springer, 2025.

[59] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

[60] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[61] 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*, pages 11953–11962, 2022.

[62] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[63] Chong Zhou, Xiangtai Li, Chen Change Loy, and Bo Dai. Edgesam: Prompt-in-the-loop distillation for on-device deployment of sam. *arXiv preprint arXiv:2312.06660*, 2023.

A Appendix

❊ In this document, we provide supplementary materials that extend beyond the scope of the main manuscript, constrained by space limitations. These additional materials include in-depth information about the ablation study of SlimSAM-50, further experiments assessing model efficiency, additional evaluations of training costs, analysis of dynamic loss, limitation discussion, and supplementary qualitative results.

A.1 Ablation Study on SlimSAM-50

We performed an extensive series of ablation studies on the SA-1B [25] dataset utilizing the SlimSAM-50 model, characterized by its significant 50% pruning ratio. To guarantee a fair and consistent comparison across these ablation studies, each model under evaluation was uniformly trained over a span of 20 epochs, employing a training dataset comprising 10,000 images.

Disturbed Taylor Pruning. Initially, we executed an ablation study to evaluate the effects of our innovative disturbed Taylor pruning technique on distillation processes. This approach strategically aligns pruning criteria with the optimization goals of the ensuing distillation, thereby facilitating enhanced performance recovery. As illustrated in Table 7, our disturbed Taylor pruning method consistently outperforms, achieving markedly better results at equivalent training expenditures. In comparison to the conventional one-step pruning strategy and our alternative slimming approach, our methodology registers MIoU enhancements of 0.31% and 0.39% over the standard Taylor pruning method [38], respectively.

Intermediate Aligning. We further investigated the impact of integrating alignment with intermediate layers in the distillation process. As Table 8 illustrates, leveraging knowledge from these intermediate layers substantially enhances the training outcomes. Specifically, when distillation in step 1 incorporates learning from both bottleneck features and final image embeddings, there is a notable 1.19% improvement in MIoU compared to a methodology reliant solely on image embeddings. Similarly, in step 2 of the distillation process, a strategy that utilizes both embedding features and final image embeddings demonstrates a 0.29% MIoU improvement over approaches exclusively based on image embeddings.

Table 7: Comparison between disturbed Taylor pruning and original Taylor pruning.

Method	MIoU↑
Taylor Pruning	70.02%
Disturbed Taylor Pruning	**70.33%**
SlimSAM-50 + Taylor	70.42%
SlimSAM-50 + Disturbed Taylor	**70.81%**

Table 8: Effect of distillation from intermediate layers and final output image embeddings.

Step	Distillation Objective	MIoU↑
Step 1	Final Image Embeddings	70.86%
Step 1	+ Bottleneck Features	**72.07%**
Step 2	Final Image Embeddings	70.52%
Step 2	+ Embedding Features	**70.81%**

Alternate Slimming. Moreover, we conducted a series of experiments to examine the efficacy of our novel alternate slimming strategy. Diverging from the traditional one-step pruning approach, our method divides structural pruning into two distinct and progressive phases. In the initial phase, pruning is exclusively focused on the dimensions pertaining to embedding features. The subsequent stage then targets dimensions associated with bottleneck features. After completing both embedding and bottleneck pruning, we employ knowledge distillation with intermediate layer alignment on

Table 9: Effect of global pruning evaluated under five different normalization approaches.

Pruning Method	Normlization	MIoU↑
Local Pruning	—	70.81%
Global Pruning	Mean	70.76%
	Max	70.77%
	Sum	70.80%
	Gaussian	**70.83%**
	Standardization	70.78%

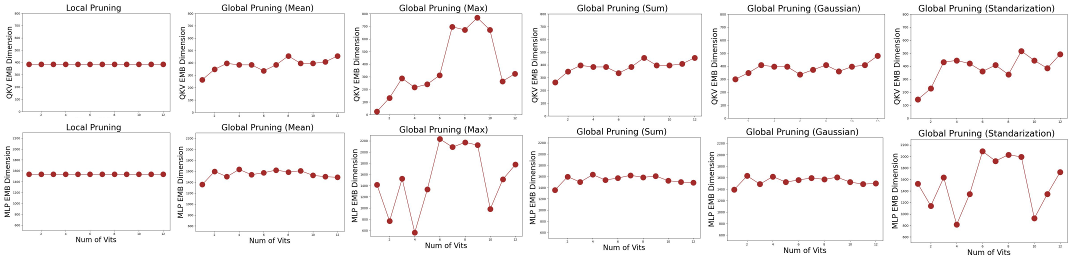


Figure 7: The intermediate dimensions of QKV Attention (top row) and MLP (bottom row) within each ViT after pruning. We present the outcomes of local pruning and global pruning under five distinct normalization methods.

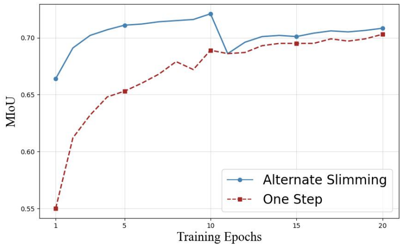


Figure 8: Training results on SA-1B with common one-step strategy and our alternate slimming strategy.

the pruned model to facilitate performance restoration. The outcomes derived from our proposed disturbed Taylor importance estimation are displayed in Figure 8. This figure demonstrates that our alternative slimming strategy significantly boosts MIoU, achieving an increase of 0.5%. When juxtaposed with the ablation study results of SlimSAM-77, it becomes evident that our strategy exhibits a more pronounced improvement, particularly when applied to models with higher pruning ratios.

Global Pruning vs Local Pruning. In our final set of experiments, we assessed the effectiveness of both local and global pruning approaches in the context of bottleneck pruning. Considering the complete decoupling of bottleneck dimensions in each block, we meticulously implemented channel-wise group pruning at varying ratios across different blocks. This was done while maintaining the predetermined overall pruning ratio for this phase of the study. To obtain a consistent global ranking, we normalized the group importance scores $I_{\mathcal{G}}$ of each layer in five ways: (i) Sum: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{\sum_{i=1}^{K} I_{\mathcal{G}i}}$, (ii) Mean: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{\sum_{i=1}^{K} I_{\mathcal{G}i}/K}$, (iii) Max: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i}}{Max_{i=1}^{K}(I_{\mathcal{G}i})}$, (iv) Standarization: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i} - Max_{i=1}^{K}(I_{\mathcal{G}i})}{Max_{i=1}^{K}(I_{\mathcal{G}i}) - Min_{i=1}^{K}(I_{\mathcal{G}i}) + 1e-8}$, (v) Gaussian: $I_{\mathcal{G}i} = \frac{I_{\mathcal{G}i} - \sum_{i=1}^{K} I_{\mathcal{G}i}/K}{\sigma_{i=1}^{K}(I_{\mathcal{G}i}) + 1e-8}$. Table 9 reveals that while local pruning maintains consistent performance, global pruning enhances the upper limit of the model's capabilities. In a departure from the findings observed with SlimSAM-77, various normalization methods do not markedly influence post-pruning performance. This suggests that the necessity of selecting an optimal normalization technique increases with the pruning ratio. For our model, we chose global pruning combined with Gaussian normalization, which led to the most favorable training outcomes. Figure 7 delineates the distribution of bottleneck feature dimensions (including QKV and MLP hidden embeddings) across each *Vision Transformer* (ViT) [6] in the image encoder. When mean, sum or Gaussian normalization is applied, the dimensions within the ViTs tend to distribute more evenly. However, employing max or standardization normalization often results in significant variances in the intermediate dimensions of each ViT.

A.2 More Analysis on Efficiency

In the principal manuscript, the high efficiency of our SlimSAM model is objectively substantiated through the disclosure of parameter counts and Multiply-Accumulate Operations (MACs). This section extends the evaluation by reporting on actual acceleration in inference, further affirming the model's efficiency. As delineated in Table 10, SlimSAM-50 outperforms the original SAM-H

Table 10: Inference acceleration was empirically tested on an NVIDIA TITAN RTX GPU, revealing that higher pruning rates significantly improve inference speed.

Pruning Ratio	Method	Speed Up↑
Ratio=0%	SAM-H [25]	Faster×1.0
	SAM-L [25]	Faster×1.7
	SAM-B [25]	Faster×4.3
Ratio=50%	SlimSAM-50(Ours)	Faster×6.9
Ratio=77%	SlimSAM-77(Ours)	Faster×8.6

Table 11: The training results were compared using the varied amounts of training data but maintaining the same training iterations.

Pruning Ratio	Training Data	Training Iters	MIoU↑
Ratio=50%	10k	100k	72.33%
	5k	100k	71.89%
	2k	100k	69.79%
Ratio=77%	10k	200k	67.40%
	5k	200k	64.47%
	2k	200k	61.72%

model by achieving a 6.9-fold increase in inference speed, while SlimSAM-77 attains an 8.3-fold acceleration. Our compression methodology markedly diminishes the actual inference time, concurrently effecting substantial reductions in both model size and MACs. The inference acceleration metrics were tested using an NVIDIA TITAN RTX GPU.

A.3 More Analysis on Training Costs

In our foundational manuscripts, we demonstrate that SlimSAM exhibits exceptional compression performance with minimal training cost. A pertinent inquiry emerges: can SlimSAM maintain its competitive performance with reduced training costs? To address this, we have undertaken supplementary experiments focusing on the interplay between training cost and performance.

In Table 11, we present the results of additional experiments conducted with varying quantities of training data, while keeping the number of training iterations constant. We observe that with a pruning ratio of 50%, a reduction in the volume of training data only marginally impacts the model's performance. Notably, even when trained with a limited dataset of just 2,000 images, our SlimSAM-50 model remarkably attains an MIoU of nearly 70%. However, as the pruning ratio is elevated to 77%, a decrease in training data more significantly affects performance. This leads to the inference that although our methodology, which integrates pruning and distillation techniques, mitigates the need for extensive training datasets, the availability of more training data can still enhance model performance, particularly at higher pruning rates. It can be anticipated that with an increase in the volume of training data, our model may potentially achieve lossless compression of SAM.

Table 12 showcases the outcomes of experiments conducted by modifying the parameters for training iterations, while maintaining a constant training dataset size. The results clearly illustrate a direct relationship between the quantity of training iterations and the effectiveness of the model compression. It is evident that more extensive training significantly improves the performance of our compressed models. Remarkably, SlimSAM maintains its superiority over other methods even when the training iterations are halved, demonstrating its robustness and efficiency in achieving high-performance compression.

A.4 More Analysis on Dynamic Loss

Further experiments were undertaken to assess the efficacy of employing dynamic loss within our intermediate feature alignment procedure. The outcomes of these ablation studies are detailed in Table 13. It was observed that a constant weight mechanism is more apt for scenarios involving a robust

Table 12: Training outcomes were evaluated using the same amount of training data across different numbers of training iterations.

Pruning Ratio	**Training Data**	**Training Iters**	**MIoU**$\uparrow$
Ratio=50%	10k	50k	70.83%
	10k	100k	72.33%
Ratio=77%	10k	100k	64.43%
	10k	200k	67.40%

Table 13: Ablation study on dynamic loss weights for distillation

Step	**Model**	**Constant** ($\alpha = 0.5$)	**Dynamic** ($\alpha = \frac{N-n-1}{N}$)
1	SlimSAM-50	**MIoU:72.07%**	MIoU:71.60%
	SlimSAM-77	**MIoU:66.32%**	MIoU:65.79%
2	SlimSAM-50	MIoU:70.42%	**MIoU:70.83%**
	SlimSAM-77	MIoU:63.63%	**MIoU:64.48%**

teacher model, whereas the implementation of a dynamic weight strategy enhances performance in instances where the teacher model exhibits lesser strength.

A.5 Limitations

In this analysis, we critically examine the constraints of our methodology.

First, our approach demonstrates robust compression performance with minimal training data. Nonetheless, an expanded training dataset could further enhance the model's capabilities. Our current pre-trained SlimSAMs, limited by hardware constraints, are trained on a dataset of only 10,000 images from the SA-1B dataset. Utilizing a more comprehensive training dataset could potentially enable our method to achieve lossless compression.

Second, the essence of our method lies in employing structural pruning and knowledge distillation to preserve the knowledge of original pre-trained SAMs. This strategy inherently sets the performance ceiling of our model at the level of the original SAM. We found it challenging to surpass the performance of the original SAM, which acted as both the target for pruning and the target for optimization. A key area for future research will be exploring how to surpass the performance of the original SAM with limited parameter counts and reduced training costs.

A.6 More Qualitative Results

We present more visual comparisons with other existing compressed models and the original SAM-H. Figure 9 provide a detailed visual comparison using the segment-everything prompt, while Figures 10 and 11 showcase additional qualitative results obtained with box prompts and point prompts, respectively. Relative to established compression models such as MobileSAM [60], FastSAM [62], EdgeSAM [63] and EfficientSAM [50], our model distinctly outperforms in achieving more precise segmentation, particularly noticeable at the object edges. Notably, even when benchmarked against SAM-H, our model demonstrates commensurate segmentation capabilities.

A.7 Societal impacts

In this paper, we introduce SlimSAM, a novel data-efficient SAM compression method that delivers superior performance using minimal training data. SlimSAM achieves an outstanding compression ratio while preserving robust segmentation capabilities. This advancement enables the deployment of SAM on resource-constrained edge devices, underscoring its significant practical applications.

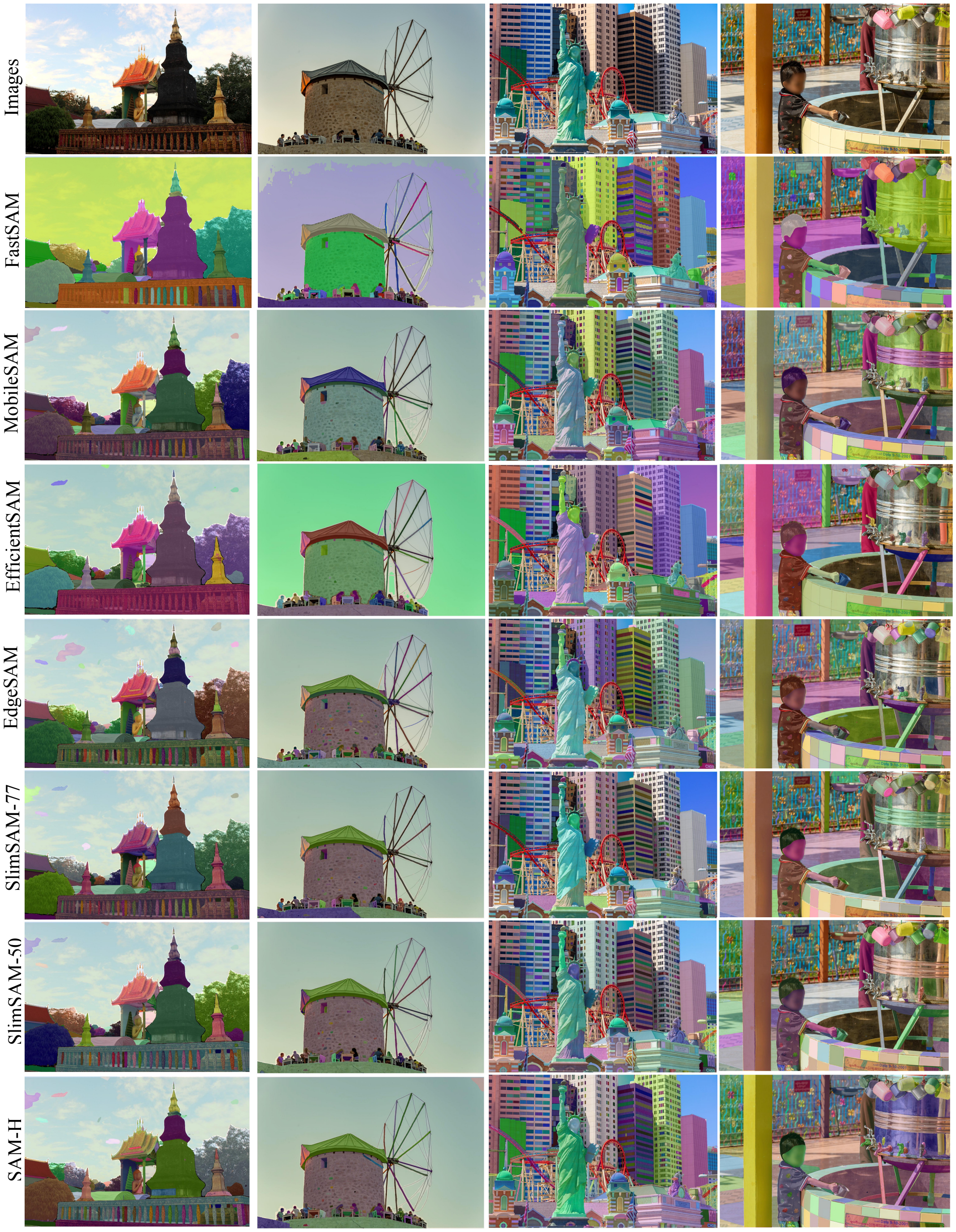


Figure 9: Comparison of segmentation results using segment everything prompts

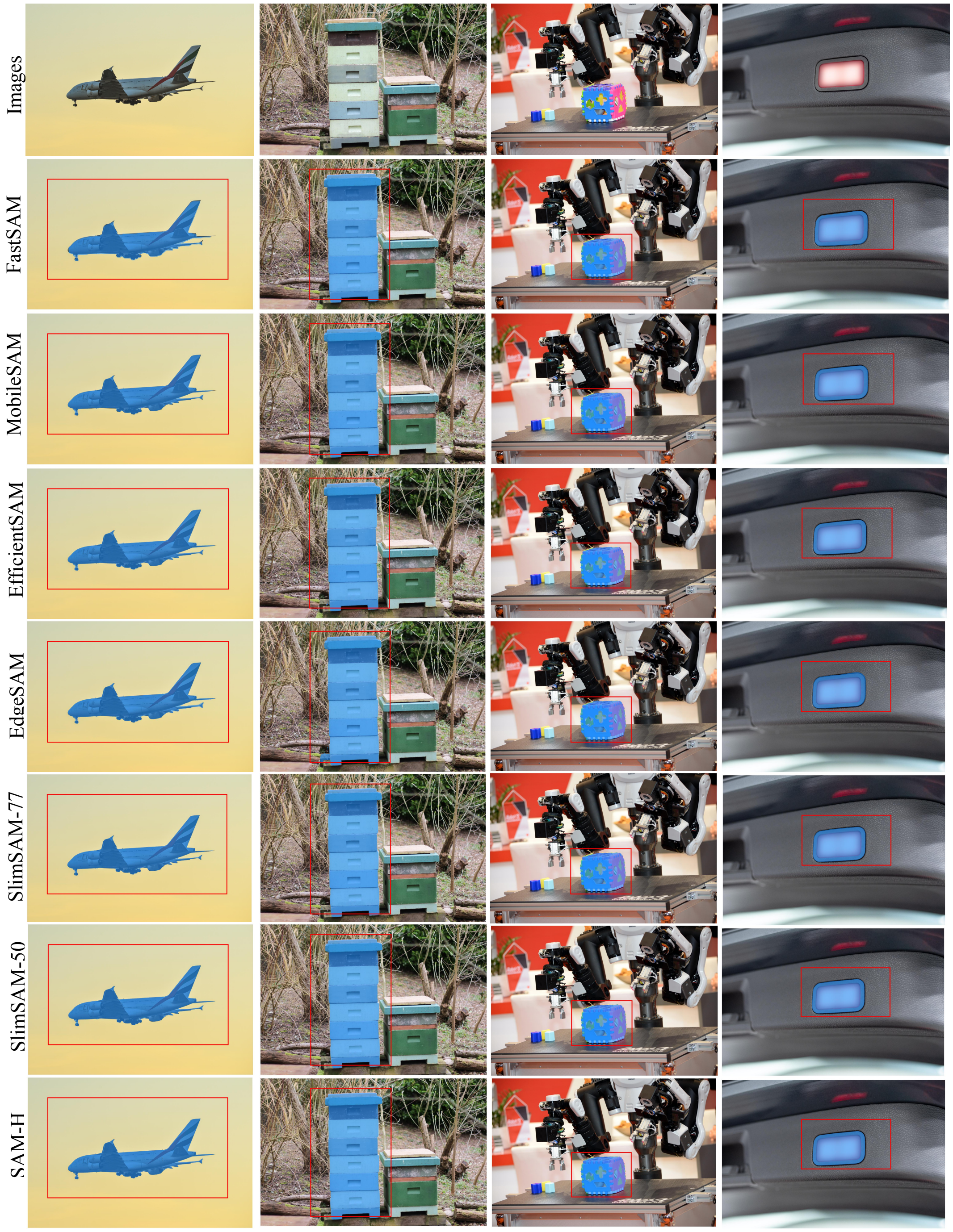


Figure 10: Comparison of segmentation results using box prompts

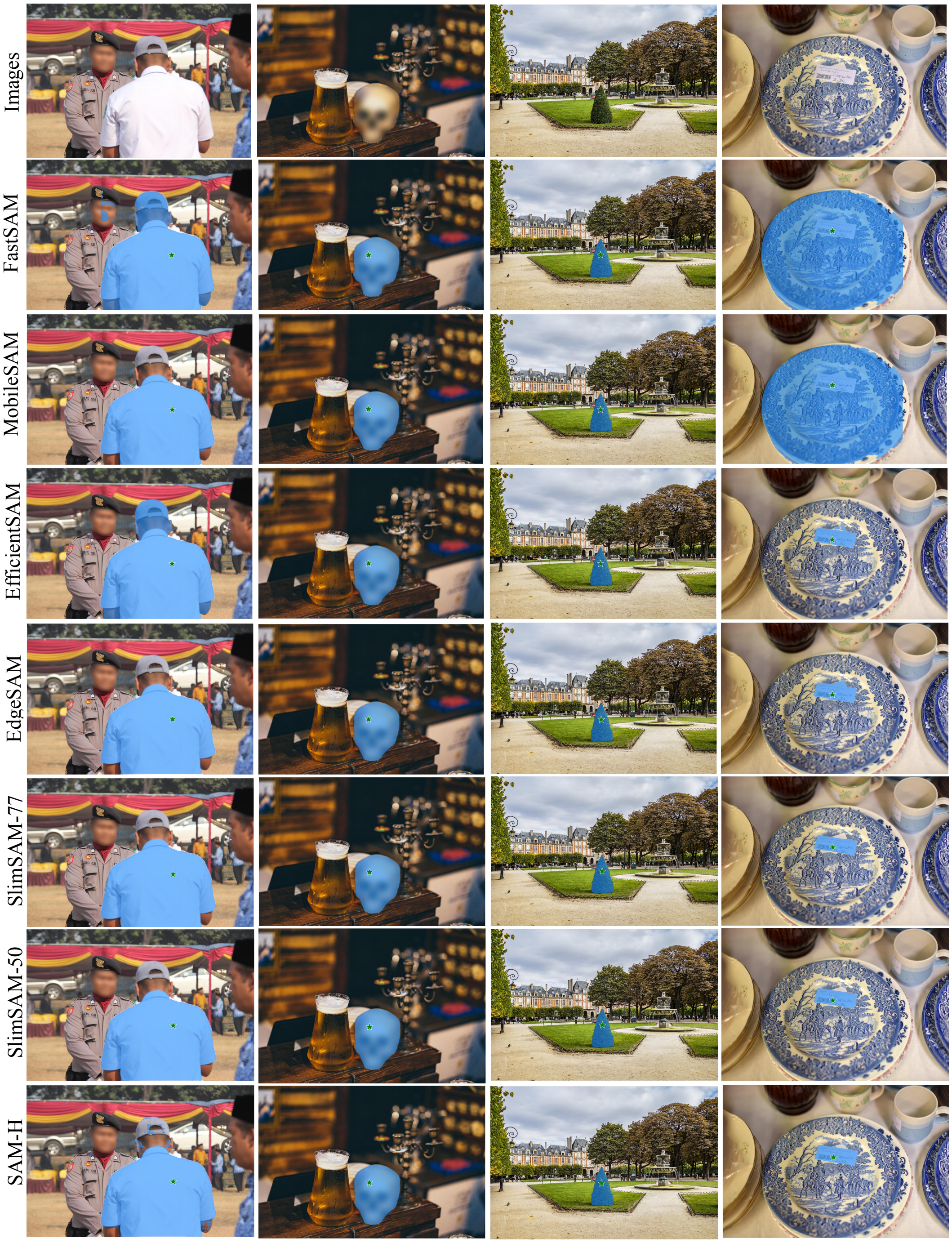


Figure 11: Comparison of segmentation results using point prompts

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in our abstract and introduction accurately reflect the paper’s contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We discuss the limitation of our work in the Appendix.A.5.

Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

   Justification: The paper provide the full set of assumptions.

   Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We provide a detailed description of our method along with extensive experimental results.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We offer the full code along with relevant instructions.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We provide all the details about the experiment in our paper.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: We provide the details about initialization and dataset split.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We provide the details about the computation resources we used in the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We strictly adhere to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We discuss the societal impacts in the Appendix.A.7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models) used in the paper are properly credited, and the license and terms of use are explicitly mentioned and properly adhered to.

    Guidelines:

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

13. **New Assets**

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

    Answer: [Yes]

    Justification: New assets introduced in the paper are well documented, and the documentation is provided alongside the assets.

    Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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