

SlimSAM: 0.1% Data Makes Segment Anything Slim

Zigeng Chen, Gongfan Fang, Xinyin Ma, and Xinchao Wang^{*}

National University of Singapore
 {zigeng99,gongfan,maxinyin}@u.nus.edu, xinchao@nus.edu.sg
<https://github.com/czg1225/SlimSAM>

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.

Keywords: Segment Anything · Model Compression · Data-Efficient

1 Introduction

Segment Anything Model (SAM) [18] has attracted considerable attention from the community since its inception. A plethora of studies [2, 14, 15, 23, 26, 27, 35, 39, 44, 45, 47] 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 [21, 41, 48, 50, 51] have been made to effectively compress SAM. Without exception, these endeavors opt to replace

^{*} Corresponding author

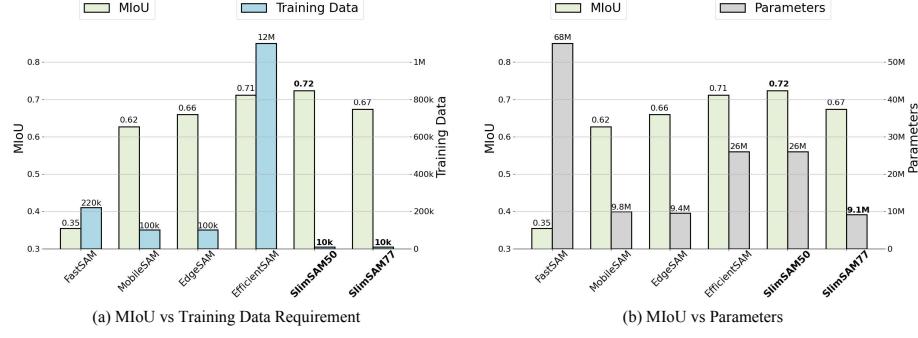


Fig. 1: (a) In contrast to scratch training or distilling from randomly initialized networks, SlimSAM achieves superior performance with significantly reduced training data requirements. (b) Maintaining a similar compression ratio, SlimSAM also outperforms other competing methodologies.

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. As depicted in Figure 1, 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 [3, 7, 10, 30, 43] to directly compress the sizable SAM by removing redundant parameters from the network and fine-tuning the streamlined model with a minimal dataset [8, 12, 22, 28]. 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 2, 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

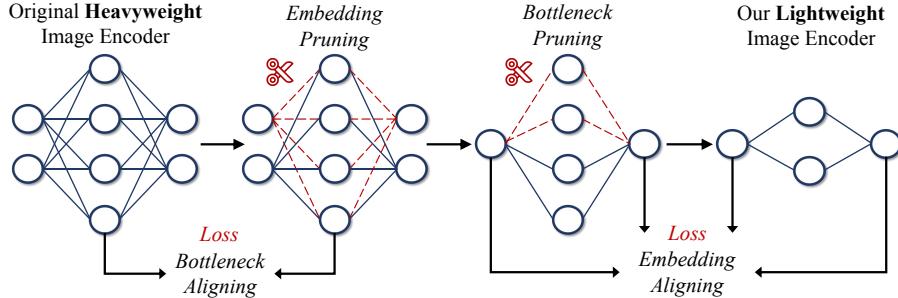


Fig. 2: A simple overall diagram of the proposed alternate slimming process.

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.

As depicted in Figure 1, 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 within a span of merely 1 to 2 days.

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 [9], model pruning [3, 8, 10, 12, 20, 22, 25, 28, 43] has proved to be an effective approach for accelerating and compressing models. Pruning techniques can be generally classified into two main categories: structural pruning [4, 7, 20, 22, 46, 46] and unstructured pruning [5, 19, 32, 34]. 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.

Knowledge Distillation. *Knowledge Distillation* (KD) [13] aims to transfer knowledge from a larger, powerful teacher model to a lighter, efficient student model. This process typically involves soft target functions and a temperature parameter to facilitate the learning. KD [1, 24, 29, 31, 36–38, 42, 49] has gained substantial prominence as an effective technique for model compression across various research domains.

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 [50] replaces SAM’s extensive ViT-based architecture with the efficient CNN-based YOLOv8-seg [16] model, while MobileSAM [48] adopts the lightweight Tinyvit [40] to replace the image encoder and employs knowledge distillation from the original encoder. EdgeSAM [51] introduces the prompt-in-the-loop knowledge distillation to accurately capture the intricate dynamics between user input and mask generation. EfficientSAM [41] innovatively adapts MAE [11] 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 [30]. Given a labeled dataset with N image pairs $\{x_i, y_i\}_{i=1}^N$ and a model \mathcal{F} with M parameters $W = \{w_i\}_{i=1}^M$. 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:

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

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:

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

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

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

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 [48], 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 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:

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

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

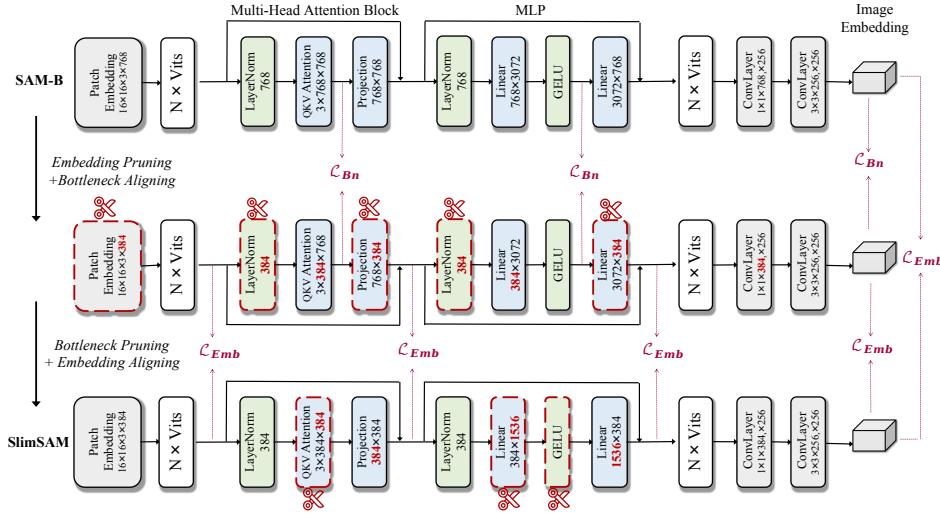


Fig. 3: The provided figure depicts our alternate slimming process with a 50% pruning ratio on SAM-B. We utilize structural pruning at the channel-wise group level to compress SAM’s image encoder, coupled with knowledge distillation from intermediate layers to restore the pruned encoder. The red numbers 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 3. 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:

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

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

$$\alpha = \begin{cases} 0.5 & n < N \\ 0 & n \geq N \end{cases}. \quad (6)$$

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:

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

where the dynamic weight α of n th epoch is defined as:

$$\alpha = \begin{cases} \frac{N-n-1}{N} & n < N \\ 0 & n \geq N \end{cases}. \quad (8)$$

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, 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%.

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 [18]	641M	2736G	11M(100%)	256	256	90k	78.30%
SAM-L [18]	312M	1315G	11M(100%)	128	128	180k	77.67%
SAM-B [18]	93M	372G	11M(100%)	128	128	180k	73.37%
FastSAM-s [50]	11M	37G	220k(2%)	32	8	625K	30.72%
FastSAM-x [50]	68M	330G	220k(2%)	32	8	625K	35.41%
MobileSAM [48]	9.8M	40G	100k(1%)	8	1	100k	62.73%
EfficientSAM-t [41]	10M	28G	12.2M(110%)	128	64	450k	69.42%
EfficientSAM-s [41]	26M	94G	12.2M(110%)	128	64	450k	71.19%
EdgeSAM [51]	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 [18]	✓	641M	2736G	78.30%
	SAM-L [18]	✓	312M	1315G	77.67%
	SAM-B [18]	✓	93M	372G	73.37%
Ratio=50%	Scratch Distillation	✗			1.63%
	Random Pruning	✗			71.03%
	Magnitude Pruning [10]	✗	26M	98G	69.96%
	Hessian Pruning [22]	✓			71.01%
	Taylor Pruning [30]	✓			71.15%
	SlimSAM-50(Ours)	✗			72.33%
Ratio=77%	Scratch Distillation	✗			1.34%
	Random Pruning	✗			62.58%
	Magnitude Pruning [10]	✗	9.1M	23G	61.60%
	Hessian Pruning [22]	✓			63.56%
	Taylor Pruning [30]	✓			64.26%
	SlimSAM-77(Ours)	✗			67.40%

4 Experiments

4.1 Experimental Settings

Implementation Details. Our SlimSAM has been implemented in PyTorch [33] and trained on a single Nvidia Titan RTX GPU using only 0.1% (10,000 images) of the SA-1B [18] dataset. The base model of our framework is SAM-B [18]. The model’s parameters were optimized through the ADAM [17] 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

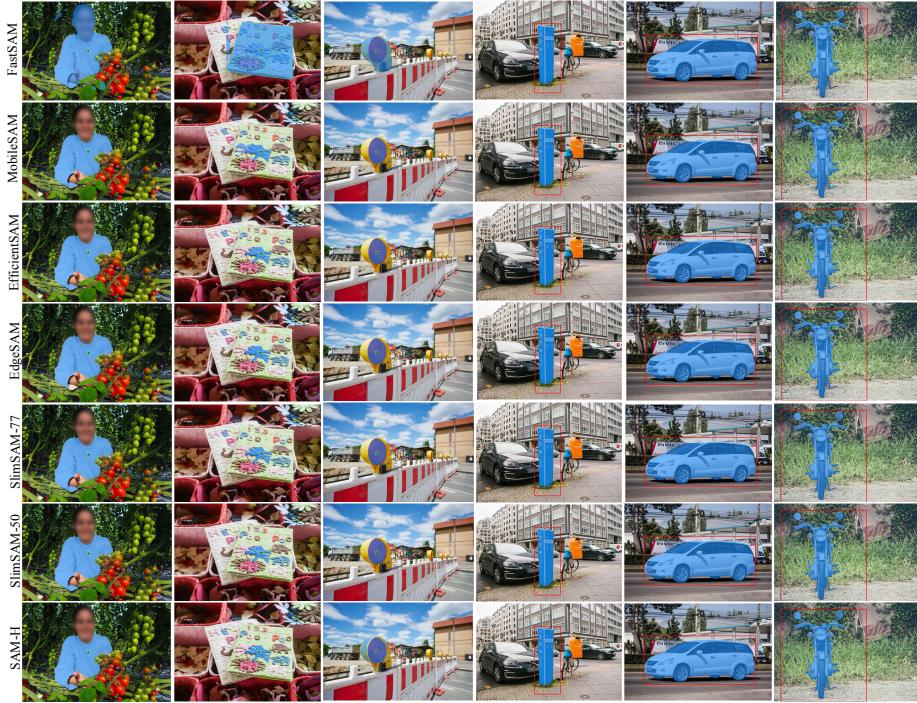


Fig. 4: Left 3 columns: segmentation results obtained using point prompts; right 3 columns: segmentation results achieved with box prompts.

while retaining SAM’s original prompt encoder and mask decoder. No additional training tricks are employed.

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%



Fig. 5: Comparison of segment everything results. In comparison to other models, our segmentation results exhibit greater proximity to the original SAM-H in terms of both accuracy and comprehensiveness.

(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 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 4 and 5. 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 prun-

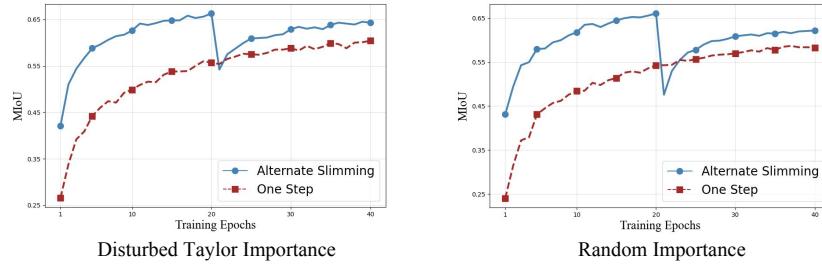


Fig. 6: Training results on SA-1B with the common one-step method and our alternate slimming framework. Left and right are results with disturbed Taylor importance and random importance.

Table 3: Comparison between proposed 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%

ing, 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 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

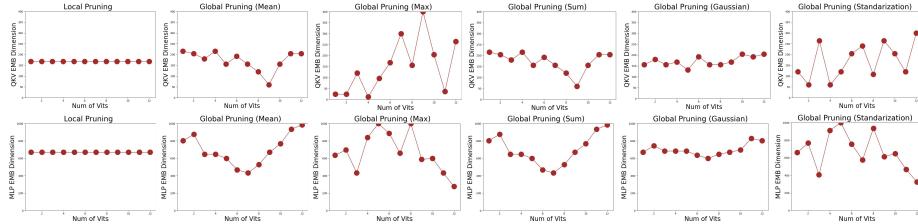


Fig. 7: The intermediate dimensions of QVK Attention (top row) and MLP (bottom row) within each ViT after pruning. We present the outcomes of local pruning and global pruning under five distinct normalization methods.

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

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

Table 6: Comparision of training results using varied amounts of training data but maintaining the same iterations.

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%

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, 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 6, 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_{G_i} of each layer in five ways: (i) Sum: $I_{G_i} = \frac{I_{G_i}}{\sum_{i=1}^K I_{G_i}}$, (ii) Mean: $I_{G_i} = \frac{I_{G_i}}{\sum_{i=1}^K I_{G_i}/K}$, (iii) Max: $I_{G_i} = \frac{I_{G_i}}{\text{Max}_{i=1}^K (I_{G_i})}$, (iv) Standardization: $I_{G_i} = \frac{I_{G_i} - \text{Max}_{i=1}^K (I_{G_i})}{\text{Max}_{i=1}^K (I_{G_i}) - \text{Min}_{i=1}^K (I_{G_i}) + 1e-8}$, (v) Gaussian: $I_{G_i} = \frac{I_{G_i} - \sum_{i=1}^K I_{G_i}/K}{\sigma_{i=1}^K (I_{G_i}) + 1e-8}$. 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 7 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.

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. Slim-SAM convincingly demonstrates its superiority while imposing significantly lower training costs compared to any other existing methods.

References

1. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems* **30** (2017)
2. Chen, K., Liu, C., Chen, H., Zhang, H., Li, W., Zou, Z., Shi, Z.: Rsprompt: Learning to prompt for remote sensing instance segmentation based on visual foundation model. arXiv preprint arXiv:2306.16269 (2023)
3. Chin, T.W., Ding, R., Zhang, C., Marculescu, D.: Towards efficient model compression via learned global ranking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1518–1528 (2020)
4. Ding, X., Ding, G., Guo, Y., Han, J.: Centripetal sgd for pruning very deep convolutional networks with complicated structure. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4943–4953 (2019)
5. Dong, X., Chen, S., Pan, S.: Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in neural information processing systems* **30** (2017)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X.: Depgraph: Towards any structural pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16091–16101 (2023)
8. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. *Advances in neural information processing systems* **36** (2024)
9. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635 (2018)
10. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. *Advances in neural information processing systems* **28** (2015)
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
12. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE international conference on computer vision. pp. 1389–1397 (2017)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. arXiv preprint arXiv:2307.12981 (2023)

15. Jing, Y., Wang, X., Tao, D.: Segment anything in non-euclidean domains: Challenges and opportunities. arXiv preprint arXiv:2304.11595 (2023)
16. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics (2023), <https://github.com/ultralytics/ultralytics>
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
19. Lee, N., Ajanthan, T., Gould, S., Torr, P.H.: A signal propagation perspective for pruning neural networks at initialization. arXiv preprint arXiv:1906.06307 (2019)
20. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
21. Liang, W., Yuan, Y., Ding, H., Luo, X., Lin, W., Jia, D., Zhang, Z., Zhang, C., Hu, H.: Expediting large-scale vision transformer for dense prediction without fine-tuning. Advances in Neural Information Processing Systems **35**, 35462–35477 (2022)
22. Liu, L., Zhang, S., Kuang, Z., Zhou, A., Xue, J.H., Wang, X., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Group fisher pruning for practical network compression. In: International Conference on Machine Learning. pp. 7021–7032. PMLR (2021)
23. Liu, S., Ye, J., Wang, X.: Any-to-any style transfer: Making picasso and da vinci collaborate. arXiv e-prints pp. arXiv–2304 (2023)
24. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2604–2613 (2019)
25. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision. pp. 2736–2744 (2017)
26. Lu, Z., Xiao, Z., Bai, J., Xiong, Z., Wang, X.: Can sam boost video super-resolution? arXiv preprint arXiv:2305.06524 (2023)
27. Ma, X., Fang, G., Wang, X.: Llm-pruner: On the structural pruning of large language models. arXiv preprint arXiv:2305.11627 (2023)
28. Ma, X., Fang, G., Wang, X.: Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems **36** (2024)
29. Mishra, A., Marr, D.: Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. arXiv preprint arXiv:1711.05852 (2017)
30. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J.: Importance estimation for neural network pruning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11264–11272 (2019)
31. Nayak, G.K., Mopuri, K.R., Shaj, V., Radhakrishnan, V.B., Chakraborty, A.: Zero-shot knowledge distillation in deep networks. In: International Conference on Machine Learning. pp. 4743–4751. PMLR (2019)
32. Park, S., Lee, J., Mo, S., Shin, J.: Lookahead: A far-sighted alternative of magnitude-based pruning. arXiv preprint arXiv:2002.04809 (2020)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
34. Sanh, V., Wolf, T., Rush, A.: Movement pruning: Adaptive sparsity by fine-tuning. Advances in Neural Information Processing Systems **33**, 20378–20389 (2020)

35. Shen, Q., Yang, X., Wang, X.: Anything-3d: Towards single-view anything reconstruction in the wild. arXiv preprint arXiv:2304.10261 (2023)
36. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for bert model compression. arXiv preprint arXiv:1908.09355 (2019)
37. Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., Liu, T.Y.: Multilingual neural machine translation with knowledge distillation. arXiv preprint arXiv:1902.10461 (2019)
38. Wang, X., Zhang, R., Sun, Y., Qi, J.: Kdgan: Knowledge distillation with generative adversarial networks. Advances in neural information processing systems **31** (2018)
39. Wu, J., Fu, R., Fang, H., Liu, Y., Wang, Z., Xu, Y., Jin, Y., Arbel, T.: Medical sam adapter: Adapting segment anything model for medical image segmentation. arXiv preprint arXiv:2304.12620 (2023)
40. Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., Yuan, L.: Tinyvit: Fast pretraining distillation for small vision transformers. In: European Conference on Computer Vision. pp. 68–85. Springer (2022)
41. Xiong, Y., Varadarajan, B., Wu, L., Xiang, X., Xiao, F., Zhu, C., Dai, X., Wang, D., Sun, F., Iandola, F., et al.: Efficientsam: Leveraged masked image pretraining for efficient segment anything. arXiv preprint arXiv:2312.00863 (2023)
42. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge distillation meets self-supervision. In: European Conference on Computer Vision. pp. 588–604. Springer (2020)
43. Yang, H., Yin, H., Shen, M., Molchanov, P., Li, H., Kautz, J.: Global vision transformer pruning with hessian-aware saliency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18547–18557 (2023)
44. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
45. Yang, Y., Wu, X., He, T., Zhao, H., Liu, X.: Sam3d: Segment anything in 3d scenes. arXiv preprint arXiv:2306.03908 (2023)
46. You, Z., Yan, K., Ye, J., Ma, M., Wang, P.: Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. Advances in neural information processing systems **32** (2019)
47. Yu, T., Feng, R., Feng, R., Liu, J., Jin, X., Zeng, W., Chen, Z.: Inpaint anything: Segment anything meets image inpainting. arXiv preprint arXiv:2304.06790 (2023)
48. Zhang, C., Han, D., Qiao, Y., Kim, J.U., Bae, S.H., Lee, S., Hong, C.S.: Faster segment anything: Towards lightweight sam for mobile applications. arXiv preprint arXiv:2306.14289 (2023)
49. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 11953–11962 (2022)
50. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., Wang, J.: Fast segment anything. arXiv preprint arXiv:2306.12156 (2023)
51. Zhou, C., Li, X., Loy, C.C., Dai, B.: Edgesam: Prompt-in-the-loop distillation for on-device deployment of sam. arXiv preprint arXiv:2312.06660 (2023)