ENTROPY AS A LENS: REVEALING STRUCTURAL IMPORTANCE IN MASKED SELF-SUPERVISED VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked self-supervised Vision Transformers (ViT) such as MAE and VideoMAE have demonstrated strong transferability but often exhibit substantial architectural redundancy. In this work, we ask a simple yet underexplored question: Are all Transformer blocks equally important for representation learning? We present a data-free analysis framework that estimates block importance using the entropy of pretrained weight-number, without relying on downstream data or task-specific sensitivity measurements. Through mathematical derivation and empirical analysis, we show that block-wise weight entropy strongly correlates with performance degradation when blocks are removed, revealing a clear hierarchy of importance across Transformer depth. Leveraging this observation, we demonstrate that a large portion of blocks in masked self-supervised ViT can be removed with minimal impact on downstream performance, suggesting significant structural redundancy in pretrained models. Beyond compression, our results provide new insights into how representation capacity is distributed across Transformer blocks, and position entropy as a lightweight diagnostic tool for understanding and analyzing large pretrained vision models.

1 INTRODUCTION

Masked self-supervised learning has emerged as a dominant paradigm for pretraining large-scale vision transformers Dosovitskiy et al. (2021), enabling strong transfer performance across a wide range of downstream tasks. Models such as MAE He et al. (2022) and VideoMAE Tong et al. (2022) learn rich visual representations from unlabeled data and have become foundational components in modern vision systems. However, the substantial size of these pretrained models poses significant challenges for deployment in resource-constrained environments, including edge devices, mobile platforms, and memory-limited local systems.

A natural approach to address this challenge is model compression Han et al. (2016), among which pruning is particularly attractive due to its potential to reduce model size and computational cost without modifying training objectives or inference pipelines. Nevertheless, existing pruning methods Cheng et al. (2024) face fundamental limitations when applied to masked self-supervised vision transformers. Sensitivity-based pruning methods Molchanov et al. (2016), while effective, require iterative block removal and full finetuning, incurring prohibitive computational overhead. Data-driven pruning approaches Theis et al. (2018) rely on downstream data or activation statistics, introducing dataset bias and limiting applicability in scenarios where data is unavailable or restricted. Moreover, unstructured pruning Molchanov et al. (2017) often leads to irregular sparsity patterns that are difficult to deploy efficiently on modern hardware.

These limitations raise a fundamental question: are all transformer Vaswani et al. (2017) blocks equally important for transfer learning in masked self-supervised vision transformers, and can block importance be estimated efficiently without access to data? To illustrate this phenomenon, Figure 1 provides a block-level pruning analysis on VideoMAE-B Tong et al. (2022). As shown in Figure 1(a), removing different transformer Vaswani et al. (2017) blocks leads to drastically different downstream performance, revealing substantial block-level heterogeneity. Figure 1(b) further shows that entropy-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

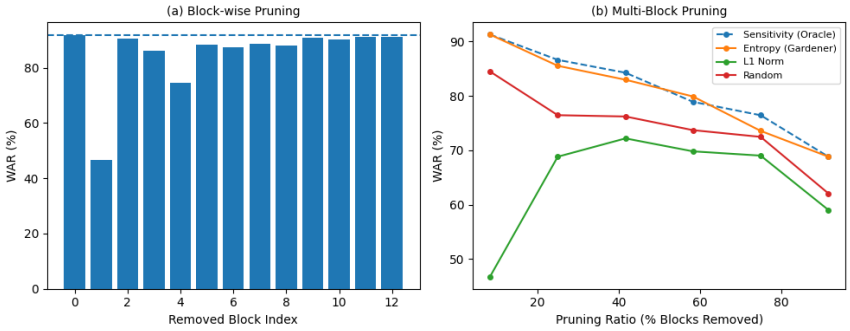


Figure 1: Pruning behavior of masked self-supervised vision transformers on VideoMAE-B finetuned on UCF101. (a) Block-wise pruning results, where index 0 denotes the original model and indices 1–12 correspond to removing individual transformer blocks. The dashed line indicates the unpruned baseline. The results reveal substantial block-level heterogeneity. (b) Multi-block pruning performance under increasing pruning ratios. Entropy-based pruning consistently tracks sensitivity-based (oracle) pruning and outperforms other data-free criteria across a wide range of pruning ratios.

based block ranking closely tracks sensitivity-based (oracle) pruning under increasing pruning ratios, motivating a principled and data-free approach to block importance estimation.

In this work, we provide a negative answer to the first question and a positive answer to the second. Through extensive block-wise analysis, we show that transformer Vaswani et al. (2017) blocks contribute unequally to downstream performance and that removing different blocks leads to drastically different accuracy degradation. While sensitivity-based pruning Molchanov et al. (2016) can reveal this heterogeneity, its high computational cost makes it impractical as a general solution. Our key insight is that block importance is encoded in the pretrained parameters themselves. Specifically, we find that the weight-number entropy of a transformer Vaswani et al. (2017) block—an information-theoretic measure of how parameter counts are distributed across value intervals—strongly correlates with oracle sensitivity obtained via iterative block removal and finetuning. This observation enables a principled, data-free surrogate for block importance estimation that operates solely on pretrained weights.

Based on this insight, we propose a data-free, one-shot, block-level pruning method for masked self-supervised vision transformers. Ranks blocks using weight-number entropy and removes redundant blocks in a single pass, without access to data, gradients, or iterative finetuning. Despite its negligible computational overhead, consistently matches or closely approaches sensitivity-based pruning and outperforms existing data-free baselines across both block-wise and multi-block pruning settings. Beyond practical performance, our analysis reveals systematic structural heterogeneity across transformer blocks in masked self-supervised models. Weight-number entropy captures intrinsic representational properties learned during pretraining, providing insight into why certain blocks are critical for transfer learning while others are redundant. We summarize our contributions as follows:

- We formulate the problem of data-free block importance estimation for masked self-supervised vision transformers under resource-constrained deployment settings.
- We introduce weight-number entropy, an information-theoretic criterion computed solely from pretrained parameters and demonstrate its strong correlation with oracle sensitivity.
- We propose a data-free, one-shot, block-level pruning algorithm with linear-time complexity and negligible memory overhead.
- We conduct extensive experiments on VideoMAE-B Tong et al. (2022), showing that consistently outperforms existing data-free pruning methods and achieves near-oracle performance under both block-wise and multi-block pruning.
- We provide empirical and conceptual analysis that sheds light on the internal organization and redundancy of masked self-supervised vision transformers.

2 Related Work

Model pruning Cheng et al. (2024) has been extensively studied to reduce the computational and memory footprint of deep neural networks. Early work primarily focused on unstructured pruning Molchanov et al. (2017) at the weight level, which achieves high compression ratios but often leads to irregular sparsity patterns that are difficult to exploit efficiently on modern hardware. More recent efforts have shifted toward structured pruning Lagunas et al. (2021), where entire filters, layers, or blocks are removed to preserve architectural regularity and deployment efficiency.

2.1 Structured Pruning

Structured pruning Lagunas et al. (2021) methods aim to remove coherent components of a model, such as channels, layers, or blocks, while maintaining performance. In convolutional networks, filter- and channel-level pruning has been widely explored. In contrast, Vision Transformers Dosovitskiy et al. (2021) introduce a natural modular structure composed of sequential transformer Vaswani et al. (2017) blocks, making block-level pruning a particularly suitable strategy. Several works have investigated layer-wise or block-wise pruning Michel et al. (2019); Voita et al. (2019) for transformers, often relying on training-time signals or activation statistics to assess importance. However, these methods typically require access to data and repeated evaluation, limiting their applicability in data-restricted or resource-constrained settings.

2.2 Data-Driven Pruning

Sensitivity-based pruning evaluates the importance of model components by measuring the performance impact of their removal, often using gradients or second-order approximations Molchanov et al. (2016). While such methods provide strong performance guarantees and can be regarded as oracle criteria, they require iterative pruning and full retraining or finetuning, resulting in substantial computational overhead. Data-driven pruning approaches Theis et al. (2018) estimate importance based on activation statistics, feature saliency, or downstream loss signals. Although effective in many settings, these approaches depend on representative data samples and may suffer from dataset bias or privacy constraints, particularly when downstream data is unavailable at pruning time.

2.3 Data-Free Pruning

Data-free pruning methods Bai et al. (2023) seek to eliminate the reliance on training or validation data by operating directly on pretrained model parameters. Existing approaches often use magnitude-based criteria Chen et al. (2022), such as weight norms, variance, or higher-order moments, to estimate importance. While computationally efficient, these statistics provide limited insight into the structural role of model components and often exhibit unstable behavior across architectures or pruning ratios Tanaka et al. (2020); Wang & Fritz (2024). As a result, their ability to approximate oracle sensitivity remains limited.

2.4 Information-Theoretic Analysis

Information-theoretic measures, including entropy and mutual information Tishby et al. (2000), have been used to analyze representation learning and model behavior in neural networks. Prior work has explored entropy-based criteria for feature selection, compression, and interpretability Shwartz-Ziv & Tishby (2017); Goldfeld et al. (2019), often in conjunction with data-dependent signals such as activations or learned representations Achille & Soatto (2018). In contrast, the use of information-theoretic analysis directly on pretrained parameters for structured pruning remains underexplored, particularly in the context of masked self-supervised vision transformers.

Our work bridges these lines of research by introducing a data-free, block-level pruning method grounded in information-theoretic analysis of pretrained parameters. Unlike sensitivity-based approaches Molchanov et al. (2016), our method avoids iterative evaluation and finetuning. Unlike existing data-free criteria Cheng et al. (2024); Bai et al. (2023), weight-number entropy captures the structural distribution of parameters within transformer blocks, enabling a closer approximation to oracle block importance.

3 METHOD

3.1 PROBLEM FORMULATION

We consider a masked self-supervised vision transformer pretrained on unlabeled data, consisting of a sequence of L transformer blocks, denoted as $\{B_1, ..., B_L\}$, with Dataset $D = \{(x_i, y_i)\}_{i=1}^{C}$. Given the pretrained model parameters, our goal is to identify a subset of blocks to be removed under a target resource budget, such that the resulting pruned model preserves maximal transfer performance after downstream finetuning. Constrained optimization problem:

$$\min_{M} \mathcal{L}(B_l \odot M; D) = \min_{M} \frac{1}{C} \sum_{i=1}^{C} \ell(B_l \odot M; D) \quad \text{s.t. } \|M\|_0 \leq L \tag{1}$$

Where $M \in \{0, 1\}$ is a binary mask matrix.

Crucially, we focus on a data-free pruning setting, where no training or validation data from downstream tasks is accessible during the pruning decision process. This setting reflects practical deployment scenarios in which pretrained models must be adapted prior to task specification or data availability and avoids dataset-induced bias commonly introduced by data-driven pruning criteria.

Formally, let $M \subseteq \{1, \ldots, L\}$ denote the set of removed blocks under a resource constraint. The objective is to select M such that the resulting model preserves maximal downstream performance after fine-tuning. Under this constraint, requires a lightweight surrogate function $f(B_l)$ that approximates oracle sensitivity while relying solely on pretrained weights.

$$f(B_l) \approx \text{Sens}(B_l) = \frac{\Delta L}{\Delta B_l} \tag{2}$$

The central challenge lies in estimating the importance of each transformer block using solely pretrained weights, without iterative finetuning, gradient-based sensitivity analysis, or data-dependent evaluation. Addressing this challenge requires a computationally efficient surrogate for block importance that can operate entirely in a data-free and one-shot manner. We seek a block importance scoring function $f(B_l)$ that ranks blocks according to their contribution to downstream transfer performance, under the constraint that f operates solely on pretrained parameters. In this work, we treat sensitivity-based pruning as an oracle criterion and seek a low-cost surrogate that approximates such oracle behavior under zero-data constraints.

3.2 BLOCK-LEVEL WEIGHT-NUMBER ENTROPY

To estimate block importance under the data-free constraint, we introduce block-level weight-number entropy, an information-theoretic measure computed solely from pretrained model parameters. Unlike conventional entropy definitions that operate on weight magnitudes or activations Tishby et al. (2000), our formulation characterizes the distribution of parameter counts within a transformer block.

Consider a transformer block B_l with parameter set W_l, containing $N_l = \|W_l\|_0$ scalar weights. We partition the range of weight values into a finite set of disjoint intervals K. Let $n_l(i)$ denote the number of weights in block B_l whose values fall into interval i. We define empirical probability mass function as

$$p_l(i) = \frac{n_l(i)}{N_l} \tag{3}$$

where the denominator corresponds to the total number of parameters in the block, rather than the sum of weight magnitudes. The weight-number entropy of block B_l is then defined as

$$H'(B_l) = -\sum_{i} p_l(i) \log p_l(i) \tag{4}$$

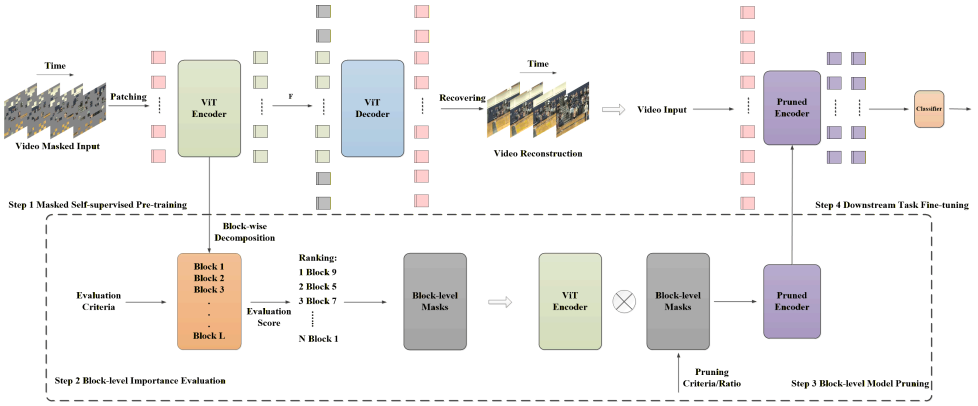


Figure 2: Architectural diagram of the information entropy-based Block-level pruning algorithm. Step 1: Visual self-supervised learning; Step 2: Calculating pruning criteria at the Block-level; Step 3: Pruning process of the VideoMAE Encoder Pretrained Model; Step 4: Fine-tuning the Pruned VideoMAE Encoder.

This formulation measures how uniformly parameter values are distributed across intervals, capturing the structural diversity of weights within each block while remaining invariant to weight scaling. Intuitively: **(1) Higher entropy indicates a more diverse parameter distribution; (2) Lower entropy indicates more concentrated parameter allocation.** By relying on parameter counts rather than magnitudes, weight-number entropy avoids biases introduced by a small number of large-magnitude weights and provides a stable, architecture-agnostic block importance score A.1.

In the data-free pruning setting, entropy values are computed independently for each block and used to rank block importance. Blocks with lower weight-number entropy are considered less critical and are preferentially removed under a given pruning budget. Weight-number entropy can be computed in a single pass over pretrained parameters, incurs negligible computational overhead, and requires neither forward or backward propagation nor access to input data. We observe that block rankings induced by weight-number entropy are robust to reasonable choices of binning strategies and interval resolutions. The mathematical formulation for important critical ranking is as follows:

$$Rank(B_l) = \text{Sort}\{H'(B_1), ..., H'(B_l)\} \tag{5}$$

3.3 Pruning Algorithm

Building upon the block-level weight-number entropy defined in Section 3.2, we propose as shown in Algorithm 1, a data-free and one-shot pruning algorithm for masked self-supervised vision transformers. It identifies and removes redundant transformer blocks solely based on pretrained model parameters, without requiring any data, gradient computation, or iterative finetuning.

Given a pretrained model consisting of L transformer blocks $\{B_1, ..., B_L\}$, it first computes the weight-number entropy $H(B_l)$ for each block independently. These entropy values serve as block importance scores. Blocks are then ranked in ascending order of entropy, where lower-entropy blocks are considered less critical to downstream transfer performance.

Let M denote the set of blocks to be removed under a target pruning budget. In contrast to iterative pruning schemes Molchanov et al. (2016), it performs one-shot selection by directly choosing the lowest-ranked blocks according to entropy scores. This design avoids repeated model modification and evaluation, enabling efficient pruning even under strict computational constraints.

It operates at the block level, removing entire transformer blocks while preserving the remaining model structure. This structured pruning strategy maintains architectural integrity and is naturally compatible with modern hardware accelerators and deployment frameworks as shown in Figure 2. After block removal, the pruned model is fine-tuned on downstream tasks using standard training procedures. Importantly, it does not alter the downstream optimization process and can be seamlessly integrated with existing transfer learning pipelines Raffel et al. (2020).

Algorithm 1 Pruning Algorithm

Input: Pretrained transformer blocks $\{B_1, \ldots, B_L\}$
pruning ratio $r \in (0, 1)$
Parameter: l
Output: Pruned transformer model M_{pruned}
for $l = 1$ to L **do**
  Extract all scalar weights $N_l = \|W_l\|_0$ from block B_l
  Partition weight values into K disjoint intervals
  Count number of weights $n_l(i)$ in each interval i
  Compute empirical probability $p_l(i) = n_l(i)/N_l$
  Compute block entropy:
  $H'(B_l) = -\sum_i p_l(i) \log p_l(i)$
end for
Rank blocks in ascending order of $H'(B_l)$
Select the lowest-ranked $\lfloor rL \rfloor$ blocks as pruning set S
Remove blocks in M from the pretrained model
Return M_{pruned}

It requires a single pass over pretrained parameters to compute block entropy scores, resulting in a time complexity $O(N_l)$ linear in the number of model parameters. No forward or backward propagation is required during pruning, making it orders of magnitude more efficient than sensitivity-based pruning methods Molchanov et al. (2016). In contrast to sensitivity-based pruning methods Molchanov et al. (2016) that require iterative finetuning and gradient computation, it provides an efficient surrogate for block importance estimation under strict resource constraints. Overall, it provides a simple yet effective mechanism for data-free block pruning, enabling rapid model adaptation and resource-efficient deployment of masked self-supervised vision transformers. We treat sensitivity-based block pruning as an oracle criterion and view it as a low-cost surrogate that approximates such oracle behavior under data-free constraints.

4 Experiments

4.1 Experimental Setup

We conduct all experiments using a VideoMAE-Base Tong et al. (2022) model pretrained on the Kinetics-400 dataset Kay et al. (2017). The pretrained model follows a Vision Transformer (ViT) Dosovitskiy et al. (2021) backbone and consists of 12 transformer Vaswani et al. (2017) blocks, with a total of 86.6M parameters (approximately 330 MB). The input to the model is a sequence of 16 RGB video frames with spatial resolution $224 * 224$.

We evaluate transfer performance on the human action recognition task using the UCF101 dataset Soomro et al. (2012). UCF101 contains 101 action categories and 10,100 videos, split into 7,070 training videos, 2,020 validation videos, and 1,010 test videos Soomro et al. (2012). Following standard practice, no additional data augmentation is applied during finetuning, and Label Ranker Xiang & Bai (2025) is used to eliminate the bias introduced by label position encoding.

After pruning, the resulting models are finetuned on UCF101 Soomro et al. (2012) using the same training configuration across all methods. We adopt the AdamW Loshchilov & Hutter (2017) optimizer with a weight decay of 0.05. The initial learning rate is set to 1×10^{-3} and scheduled using a Warm-up followed by Cosine Decay Loshchilov & Hutter (2016). Models are trained for 100 epochs, with the first 5 epochs used for learning rate Warm-up. The batch size is fixed to 64 for all experiments.

We report performance using Weighted Average Recall (WAR), which accounts for class imbalance and is commonly used in action recognition benchmarks. All reported results are evaluated on the test split of UCF101 Soomro et al. (2012). All experiments are conducted on a V2-8 TPU with high-memory configuration. Importantly, pruning decisions are made entirely without access to downstream data or training signals, and computational resources are only used during the standard finetuning stage.

Table 1: Block-wise pruning performance of VideoMAE-B finetuned on UCF101. RB: Remove Block. See A.2 for details.

Model	Params (M)	GFLOPs	WAR (%)
Original	86.6	180	91.78
RB-1	79.5	166	46.73
RB-2	79.5	166	90.58
RB-3	79.5	166	86.14
RB-4	79.5	166	74.65
RB-5	79.5	166	88.32
RB-6	79.5	166	87.43
RB-7	79.5	166	88.51
RB-8	79.5	166	88.02
RB-9	79.5	166	90.79
RB-10	79.5	166	90.30
RB-11	79.5	166	91.09
RB-12	79.5	166	**91.29**

Table 2: Comparison between entropy-based vs. sensitivity-based oracle ranking. See A.3 for details.

Block	WAR (%)	Entropy	Sensitivity
-	-	-	-
1	46.73	1	1
4	74.65	3	2
3	86.14	5	3
6	87.43	4	4
5	88.32	6	6
7	88.51	8	7
8	88.02	7	5
2	90.58	2	9
9	90.79	9	10
10	90.30	11	8
11	91.09	10	11
12	91.29	12	12

4.2 Oracle Analysis

We evaluate whether weight-number entropy provides a reliable surrogate for block importance by comparing it against sensitivity-based pruning Molchanov et al. (2016), which we treat as an oracle criterion. Sensitivity-based analysis measures the impact of removing transformer Vaswani et al. (2017) blocks on downstream performance but requires iterative block removal and full finetuning, resulting in high computational cost.

We first conduct block-wise pruning experiments by removing a single transformer Vaswani et al. (2017) block at a time from a pretrained VideoMAE-B Tong et al. (2022) model and finetuning the resulting model on UCF101 Soomro et al. (2012). The performance degradation induced by removing each block serves as a direct estimate of block sensitivity. As shown in Table 1, removing different blocks leads to substantially different downstream accuracies, confirming that transformer Vaswani et al. (2017) blocks are not equally important for transfer learning.

We then rank all blocks using weight-number entropy computed from pretrained parameters and compare this ranking with the oracle sensitivity ranking derived from block-wise pruning as shown in Table 2. We observe a strong alignment between the two rankings: blocks with higher weight-number entropy consistently correspond to larger performance drops when removed, whereas blocks with lower entropy exhibit smaller impact on downstream accuracy. This result indicates that weight-number entropy captures intrinsic properties of block importance that are otherwise revealed only through expensive oracle evaluation.

To further validate the correlation between entropy-based importance estimation and oracle sensitivity, we evaluate multi-block pruning under identical pruning ratios using different data-free criteria. As summarized in Table 3, entropy-based pruning consistently outperforms alternative data-free baselines, including random selection Chen et al. (2022), mean-based, variance-based, norm-based, and kurtosis-based pruning Cheng et al. (2024) etc. Across pruning ratios, entropy-based pruning achieves downstream performance closest to sensitivity-based pruning Molchanov et al. (2016), despite operating in a one-shot and data-free manner.

Taken together, the block-wise and multi-block results provide strong empirical evidence that weight-number entropy serves as a reliable surrogate for sensitivity-based block importance. Importantly, this alignment holds across both fine-grained (single-block) and coarse-grained (multi-block) pruning settings, validating the effectiveness of entropy-based importance estimation beyond isolated cases. These findings justify the use of entropy as the core criterion and explain its ability to achieve near-oracle pruning performance with negligible computational overhead.

Table 3: Multi-block pruning performance under different pruning ratios on UCF101.PR: Pruning Ratio. See A.4 for details.

PR (%)	WAR (%)				GLOPs
	Sensitivity	Entropy	L1 Norm	Random	
8.3	**91.29**	**91.29**	46.73	84.49	166
25.0	**86.63**	85.54	68.81	76.45	139
41.7	**84.26**	82.97	72.18	76.21	112
58.3	78.91	**79.90**	69.80	73.70	84
75.0	**76.44**	73.56	69.01	72.46	57
91.7	**68.81**	**68.81**	59.01	62.05	30

4.3 RESULTS

We now evaluate the effectiveness of it under both block-wise and multi-block pruning settings and analyze how different data-free criteria align with oracle sensitivity. All experiments are conducted on VideoMAE-B Tong et al. (2022) pretrained on Kinetics-400 Kay et al. (2017) and finetuned on UCF101 Soomro et al. (2012).

We first remove a single transformer Vaswani et al. (2017) block at a time and finetune the resulting model. The downstream accuracies are reported in Table 1. Removing different blocks leads to substantial performance variation, indicating that transformer Vaswani et al. (2017) blocks contribute unequally to transfer performance. Removing certain blocks results in dramatic accuracy degradation, while removing others has a relatively minor impact. These results define block-wise sensitivity, which we treat as an oracle measure of block importance.

To better understand how different data-free criteria approximate oracle sensitivity, we summarize block-wise statistics and ranking results in Table 2. The table reports block-wise accuracy, entropy-based scores, mutual information scores, efficiency-related metrics, and their corresponding rankings. We observe that entropy-based ranking exhibits the strongest alignment with sensitivity ranking, consistently assigning higher importance to blocks whose removal leads to larger performance drops. In contrast, magnitude-based and efficiency-based criteria show weaker or less stable correspondence with oracle sensitivity.

We further evaluate it under multi-block pruning by removing multiple blocks simultaneously under identical pruning ratios. The results are summarized in Table 3. Across all pruning settings, it consistently outperforms alternative data-free pruning baselines and remains closest to sensitivity-based pruning Molchanov et al. (2016). These findings demonstrate that the superiority of entropy-based importance estimation extends beyond single-block analysis and generalizes to more aggressive multi-block pruning scenarios.

Overall, the block-wise, statistical, and multi-block results jointly confirm that weight-number entropy provides a robust and scalable surrogate for oracle sensitivity, enabling effective one-shot pruning without access to data or gradients.

5 DISCUSSION

This section discusses why weight-number entropy is an effective block importance indicator, what it reveals about masked self-supervised vision transformers, and the limitations of the proposed approach. Our results show that block-level weight-number entropy strongly correlates with sensitivity-based block importance, despite being computed solely from pretrained parameters. This effectiveness stems from the fact that weight-number entropy captures how parameters are structurally allocated within a transformer Vaswani et al. (2017) block. Unlike magnitude-based statistics, which focus on the scale of weights, entropy measures the distribution of parameter counts across value intervals, reflecting the diversity of representational pathways encoded by a block.

Empirically, we observe that blocks with higher weight-number entropy consistently induce larger performance degradation when removed, indicating higher importance for downstream transfer.

Notably, this relationship holds even when visual inspection of weight distributions suggests sharp concentration as shown in A.5 Figure 3. This highlights a key distinction between apparent value concentration and entropy: a block may exhibit sharp peaks while still maintaining a broadly distributed allocation of parameters across intervals, resulting in high entropy and strong representational capacity. Sensitivity-based pruning Molchanov et al. (2016) evaluates block importance by measuring the downstream performance drop caused by block removal and can be regarded as an oracle criterion. However, it requires iterative block removal and full finetuning, making it computationally prohibitive. Our experiments demonstrate that weight-number entropy provides a low-cost surrogate that closely approximates sensitivity-based rankings at the block level. This alignment suggests that entropy captures intrinsic properties of pretrained representations that govern their contribution to transfer learning, even before any downstream data is observed.

The observed entropy patterns offer insights into the internal organization of masked self-supervised vision transformers. Early and intermediate blocks tend to exhibit higher weight-number entropy, reflecting their role in capturing diverse low-level and mid-level visual patterns under reconstruction objectives. Deeper blocks show progressively lower entropy, indicating increased specialization toward semantic abstraction. This structural heterogeneity explains why blocks are not equally important for transfer learning and motivates block-level pruning as a natural and effective compression strategy for ViT-based Dosovitskiy et al. (2021) masked autoencoders.

Despite its effectiveness, it has several limitations. First, weight-number entropy is a coarse-grained, static measure that does not account for input-dependent behavior or task-specific activation patterns. While this design is intended to enable data-free pruning, it may limit adaptability to highly specialized downstream tasks. Second, the current study focuses on block-level pruning in ViT-based Dosovitskiy et al. (2021) masked self-supervised models; extending entropy-based criteria to finer-grained structures such as attention heads or MLP Michel et al. (2019); Voita et al. (2019) sublayers remains an open question. Finally, although entropy correlates well with sensitivity in our experiments, it is not guaranteed to perfectly approximate oracle importance in all architectures or training regimes. These limitations suggest several promising directions for future work, including combining entropy-based criteria with lightweight data-aware signals, extending the approach to other self-supervised modalities such as audio or multimodal transformers, and exploring entropy-aware model design that explicitly encourages redundancy where compression is desirable. We believe that information-theoretic analysis of pretrained parameters offers a principled pathway toward more efficient and interpretable foundation models.

6 Conclusion

In this work, we studied the problem of block importance estimation in masked self-supervised vision transformers under strict data-free and resource-constrained settings. We showed that transformer Vaswani et al. (2017) blocks contribute unequally to downstream transfer performance and that identifying redundant blocks is critical for efficient deployment.

To address this challenge, we introduced a data-free, one-shot, block-level pruning method based on weight-number entropy. By analyzing the distribution of pretrained parameters, it provides a computationally efficient surrogate for sensitivity-based pruning without requiring access to data, gradients, or iterative finetuning. Extensive experiments on VideoMAE-B Tong et al. (2022) demonstrate that entropy-based block ranking strongly correlates with oracle sensitivity and consistently outperforms existing data-free pruning criteria across both block-wise and multi-block pruning settings.

Beyond empirical performance, our analysis reveals systematic structural heterogeneity across transformer blocks in masked self-supervised models. Weight-number entropy captures intrinsic representational properties encoded during pretraining, offering insight into why certain blocks are more critical for transfer learning than others.

Overall, this work highlights the potential of information-theoretic analysis of pretrained parameters as a principled approach to model compression. We hope that our findings will inspire further research on data-free and resource-efficient adaptation of large-scale self-supervised models for real-world deployment.

REFERENCES

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Shipeng Bai, Jun Chen, Xintian Shen, Yixuan Qian, and Yong Liu. Unified data-free compression: Pruning and quantization without fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5876–5885, 2023.

Tianlong Chen, Zhenyu Zhang, Jun Wu, Randy Huang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Can you win everything with a lottery ticket? *Transactions on Machine Learning Research*, 2022.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Ziv Goldfeld et al. Estimating information flow in deep neural networks. *Proceedings of the ICML Workshop*, 2019.

Song Han et al. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

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 (CVPR)*, 2022.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

François Lagunas et al. Block pruning for faster transformers. In *Proceedings of EMNLP*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Dmitry Molchanov et al. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Colin Raffel et al. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*, 2020.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. In *arXiv preprint arXiv:1703.00810*, 2017.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Hidenori Tanaka et al. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. Faster gaze prediction with dense networks and fisher pruning. *arXiv preprint arXiv:1801.05787*, 2018.

Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. *arXiv preprint arXiv:physics/0004057*, 2000.

Zizheng Tong, Yifan Song, Jue Wang, Li Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ashish Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Elena Voita et al. Analyzing multi-head self-attention: Specialized heads do the heavy lifting. In *Proceedings of the ACL*, 2019.

Hui-Po Wang and Mario Fritz. Language models as zero-shot lossless gradient compressors: Towards general neural parameter prior models. *Advances in Neural Information Processing Systems*, 37: 63634–63653, 2024.

Peihao Xiang and Ou Bai. Label ranker: Self-aware preference for classification label position in visual masked self-supervised pre-trained model. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pp. 1535–1541, 2025.

A Appendix

A.1 Why Entropy Reflects Importance

Fisher Information Matrix-based Sensitivity Theis et al. (2018) measures $S(B_l)$ the amount of information that the weights carry about the model's parameters. Compute the Fisher Information Matrix $I(B_l) = \mathrm{Var}[S(B_l)]$ for each weight, which measures how sensitive the likelihood of the data is to that weight. Time complexity as $T = \mathcal{O}(N^2 \cdot C(\nabla)) = \mathcal{O}(N^3)$.

$$S(B_l) = \nabla_{B_l} \log \mathcal{L}(B_l; D) = \frac{\partial \log \mathcal{L}}{\partial B_l} \tag{6}$$

However, the resource cost of calculating the Fisher information matrix is too high, and its likelihood function itself has limitations, that is, it only considers the change of each weight relative to the model performance, while the statistical relationship between weights is directly ignored.

In response to the above problems, we began to consider starting from the Fisher likelihood function $S(B_l)$, directly modifying the partial derivative of the weight to the partial derivative of the weight entropy to evaluate the information contained in this type of weight. Time complexity as $T = \mathcal{O}(C(\nabla)) = \mathcal{O}(N)$.

$$S(B_l) = \nabla_{H(B_l)} \log \mathcal{L}(B_l; D) = \frac{\partial \log \mathcal{L}}{\partial H(B_l)} \tag{7}$$

The information $H'(B_l)$ of this type of weight to the model can be approximately used as a measure of sensitivity $S(B_l)$. A source set B_l with a non-uniform distribution will have less entropy than the same set with a uniform distribution (i.e. the "optimized alphabet"). However, there are differences in the information entropy of weights $H(B_l)$ because the distribution of weights is not uniform but normal (Gaussian distribution). Therefore, we propose to use weight-number entropy $H'(B_l)$ to solve the problem.

$$S(B_l) = \nabla_{H(B_l)} \log \mathcal{L}(B_l; D) \approx H'(B_l) \tag{8}$$

A.1.1 UNIFORM WEIGHT DISTRIBUTION CASE

If the block B_l is uniformly distributed, K is the total number of weights, and there are $i(i \ll K)$ types of weights, the number of weights in each type is $n(n = K/i)$. Then,

$$\begin{aligned} H(B_l) &= -\frac{|B_l|}{\sum_{i=1}^{k}|B_i|}\log\frac{|B_l|}{\sum_{i=1}^{k}|B_i|} \\ &= -\frac{|B_l|}{n\sum_{j=1}^{k}|B_j|}\log\frac{|B_l|}{n\sum_{j=1}^{k}|B_j|} \end{aligned} \tag{9}$$

So, there are n numbers of each type of weight entropy $H(B_j)$.

$$H(B_l) \subseteq \bigcup_{j=1}^{k} nH(B_j) \tag{10}$$

Sensitivity:

$$S(B_l) = \nabla_{H(B_l)} \log \mathcal{L}(B_l; D) \approx H'(B_l) \tag{11}$$

Corollary: The smaller the weight, the greater the weight entropy; the greater the weight entropy, the lower the sensitivity; and vice versa.

A.1.2 GAUSSIAN WEIGHT DISTRIBUTION CASE

If the block B_l is normal distribution (gaussian distribution), K is the total number of weights. Then,

$$H(B_l) = -\frac{|B_l|}{\sum_{i=1}^{k}|B_i|}\log\frac{|B_l|}{\sum_{i=1}^{k}|B_i|} \tag{12}$$

Therefore, the number of weight entropy is the same as the number of weights of this type. Sensitivity:

$$S(B_l) = \nabla_{H(B_l)} \log \mathcal{L}(B_l; D) \approx H'(B_l) \tag{13}$$

Corollary: The smaller the weight, the greater the weight entropy; the greater the weight entropy, the lower the sensitivity; and vice versa.

A.2 FULL BLOCK-WISE PRUNING RESULTS

Table 4: Full block-wise pruning results of VideoMAE-B finetuned on UCF101, including detailed training, validation, and test metrics for each block removal.

Model State	Size (MB)	#Params (M)	Train Loss	Train Acc. (%)	Val Loss	Val Acc. (%)	Test Loss	Test Acc. (%)	GFLOPs	WAR (%)
Original	330.26	86.6	0.2223	95.20	0.2808	94.91	0.3608	91.67	180	91.78
Remove Block-1	303.22	79.5	1.9419	47.00	1.9590	47.73	1.9471	47.40	166	46.73
Remove Block-2	303.22	79.5	0.0716	97.78	0.3205	91.33	0.3009	90.77	166	90.58
Remove Block-3	303.22	79.5	0.1481	95.57	0.4028	88.91	0.4556	86.04	166	86.14
Remove Block-4	303.22	79.5	0.7273	78.84	0.8952	75.50	0.8930	74.79	166	74.65
Remove Block-5	303.22	79.5	0.1647	94.94	0.3836	88.86	0.3981	88.54	166	88.32
Remove Block-6	303.22	79.5	0.2022	93.72	0.4436	87.70	0.4581	87.50	166	87.43
Remove Block-7	303.22	79.5	0.0997	96.95	0.3459	90.02	0.4151	88.65	166	88.51
Remove Block-8	303.22	79.5	0.1119	96.95	0.3991	89.06	0.4002	87.81	166	88.02
Remove Block-9	303.22	79.5	0.0588	98.20	0.2989	91.89	0.3278	90.42	166	90.79
Remove Block-10	303.22	79.5	0.0792	97.68	0.2927	92.49	0.3468	90.52	166	90.30
Remove Block-11	303.22	79.5	0.0718	97.90	0.3131	90.78	0.3046	91.25	166	91.09
Remove Block-12	303.22	79.5	0.0183	**99.76**	0.3110	**91.78**	0.3297	**91.46**	166	**91.29**

Due to space constraints, we provide the full block-wise pruning results and detailed training statistics in Table 4. These results include training, validation, and test metrics for each block removal setting.

A.3 Full Block-wise Statistical Analysis

Table 5 and Table 6 reports detailed block-wise statistics, including entropy, mutual information, magnitude-based metrics, and their corresponding rankings.

Table 5: Statistical comparison of block-wise weights for VideoMAE-B (K400 training). This table reports performance, information-theoretic metrics, and efficiency-related statistics.

Model	Acc. (%)	Δ (%)	Rank	Entropy_N	Rank	MI_N	Rank	Efficiency	Rank	Entropy_V	Rank	MI_V	Rank	Sensitive	Rank
Original	**91.78**	–	–	–	–	–	–	0.5150	–	–	–	–	–	–	–
Block-1	46.73	45.05	1	1	1	1	**1**	0.6054	1	0	12	0	12	12.8957	1
Block-2	90.58	1.20	9	0.3420	2	0.4653	2	0.5971	2	0.7617	11	0.7095	11	0.3252	9
Block-3	86.14	5.64	3	0.1845	5	0.3161	5	0.5951	5	0.8746	8	0.8114	8	1.5121	3
Block-4	74.65	17.13	2	0.2297	3	0.3559	3	0.5956	3	0.8296	10	0.7731	10	4.6835	2
Block-5	88.32	3.46	6	0.1715	6	0.3027	6	0.5949	6	0.8782	7	0.8151	7	0.9307	6
Block-6	87.43	4.35	4	0.2038	4	0.3322	4	0.5953	4	0.8533	9	0.7942	9	1.1842	4
Block-7	88.51	3.27	7	0.1443	8	0.2710	8	0.5946	8	0.9018	5	0.8412	5	0.9003	7
Block-8	88.02	3.76	5	0.1499	7	0.2794	7	0.5946	7	0.8946	6	0.8321	6	1.0233	5
Block-9	90.79	0.99	10	0.0627	9	0.1963	9	0.5935	9	0.9626	4	0.8896	4	0.2622	10
Block-10	90.30	1.48	8	0.0134	11	0.1436	10	0.5929	11	1	**1**	0.9202	3	0.3676	8
Block-11	91.09	0.69	11	0.0262	10	0.1176	11	0.5931	10	0.9911	3	0.9341	2	0.1554	11
Block-12	91.29	**0.49**	**12**	0	**12**	0	12	0.5927	**12**	0.9991	2	1	**1**	0.0991	**12**

Table 6: Distribution statistics of block-wise weights, including norm-based and higher-order statistical measures.

Model	Cot. zero	Max	Rank	Mean	Rank	L1 Norm	Rank	L2 Norm	Rank	Var	Rank	Std	Rank	Kurtosis	Rank
Original	46	1.7090	–	0.0387	–	–	–	–	–	0.001074	–	0.0328	–	3.1024	–
Block-1	9	1.7090	1	0.0349	**12**	0	**12**	0.3549	3	0.001544	1	0.0393	1	16.2497	**1**
Block-2	6	1.1221	8	0.0369	7	0.1354	7	0.1058	5	0.000996	4	0.0316	4	1.9431	5
Block-3	2	0.9580	12	0.0373	5	0.1628	5	0.0887	6	0.000940	7	0.0307	7	1.3087	9
Block-4	5	1.2158	6	0.0366	10	0.1130	10	0.0572	9	0.000945	6	0.0307	6	2.1245	3
Block-5	4	0.9691	11	0.0372	6	0.1544	6	0.0811	8	0.000937	8	0.0306	8	1.5999	8
Block-6	1	1.3604	3	0.0367	9	0.1240	9	0.0566	10	0.000932	9	0.0305	9	2.0342	4
Block-7	4	1.2480	5	0.0363	11	0.0956	11	0	**12**	0.000876	**12**	0.0296	**12**	1.8137	6
Block-8	1	1.1133	9	0.0367	8	0.1246	8	0.0386	11	0.000904	10	0.0301	10	1.6516	7
Block-9	4	1.2988	4	0.0378	4	0.1940	4	0.0846	7	0.000900	11	0.0300	11	0.9015	10
Block-10	5	1.1523	7	0.0403	3	0.3666	3	0.2633	4	0.000990	5	0.0315	5	0.5767	11
Block-11	4	1.0361	10	0.0444	2	0.6527	2	0.5957	2	0.001214	3	0.0348	3	0.5308	12
Block-12	1	1.4404	2	0.0495	1	1	1	1	1	0.001518	2	0.0390	2	2.1493	2

A.4 Full Multi-block Pruning Strategies

Table 7: Pruning strategies when removing 1 block (8.30%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
8.30%	Random	–	166	84.49	7.29
	Mean	1	166	46.73	45.05
	Variance	7	166	88.51	3.27
	L1 Norm	1	166	46.73	45.05
	Sensitivity	12	166	91.29	0.49
	L2 Norm	7	166	88.51	3.27
	Standard Deviation	7	166	88.51	3.27
	Kurtosis	1	166	46.73	45.05
	Information Entropy	12	166	91.29	0.49
	Mutual Information	1	166	46.73	45.05

Table 8: Pruning strategies when removing 3 blocks (25%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
25%	Random	–	139	76.45	15.33
	Mean	1,4,7	139	68.81	22.97
	Variance	7,8,9	139	85.25	6.53
	L1 Norm	1,4,7	139	68.81	22.97
	Sensitivity	9,11,12	139	86.63	5.15
	L2 Norm	6,7,8	139	83.56	8.22
	Standard Deviation	7,8,9	139	85.25	6.53
	Kurtosis	1,4,12	139	64.36	27.42
	Information Entropy	10,11,12	139	85.54	6.24
	Mutual Information	1,2,4	139	60.99	30.79

Table 9: Pruning strategies when removing 5 blocks (41.70%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
41.70%	Random	–	112	76.21	15.57
	Mean	1,4,6,7,8	112	72.18	19.60
	Variance	5,6,7,8,9	112	79.70	12.08
	L1 Norm	1,4,6,7,8	112	72.18	19.60
	Sensitivity	2,9,10,11,12	112	84.26	7.52
	L2 Norm	4,5,6,7,8	112	81.58	10.20
	Standard Deviation	5,6,7,8,9	112	79.70	12.08
	Kurtosis	1,2,4,6,12	112	70.30	21.48
	Information Entropy	7,9,10,11,12	112	82.97	8.81
	Mutual Information	1,2,3,4,6	112	62.48	29.30

Table 10: Pruning strategies when removing 7 blocks (58.30%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
58.30%	Random	–	84	73.70	18.08
	Mean	1,2,4,5,6,7,8	84	69.80	21.98
	Variance	3,4,5,6,7,8,9	84	78.02	13.76
	L1 Norm	1,2,4,5,6,7,8	84	69.80	21.98
	Sensitivity	2,5,7,9,10,11,12	84	78.91	12.87
	L2 Norm	3,4,5,6,7,8,9	84	78.02	13.76
	Standard Deviation	3,4,5,6,7,8,9	84	78.02	13.76
	Kurtosis	1,2,4,6,7,8,12	84	70.40	21.38
	Information Entropy	5,7,8,9,10,11,12	84	79.90	11.88
	Mutual Information	1,2,3,4,5,6,8	84	65.15	26.63

Table 11: Pruning strategies when removing 9 blocks (75%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
75%	Random	–	57	72.46	19.32
	Mean	1,2,3,4,5,6,7,8,9	57	69.01	22.77
	Variance	2,3,4,5,6,7,8,9,10	57	72.97	18.81
	L1 Norm	1,2,3,4,5,6,7,8,9	57	69.01	22.77
	Sensitivity	2,5,6,7,8,9,10,11,12	57	76.44	15.34
	L2 Norm	2,3,4,5,6,7,8,9,10	57	72.97	18.81
	Standard Deviation	2,3,4,5,6,7,8,9,10	57	72.97	18.81
	Kurtosis	1,2,3,4,5,6,7,8,12	57	70.30	21.48
	Information Entropy	3,5,6,7,8,9,10,11,12	57	73.56	18.22
	Mutual Information	1,2,3,4,5,6,7,8,9	57	69.01	22.77

Table 12: Pruning strategies when removing 11 blocks (91.70%).

Pruning Ratio	Method	Removed Block	GFLOPs	WAR (%)	Δ (%)
91.70%	Random	–	30	62.05	29.73
	Mean	1,2,3,4,5,6,7,8,9,10,11	30	59.01	32.77
	Variance	2,3,4,5,6,7,8,9,10,11,12	30	68.81	22.97
	L1 Norm	1,2,3,4,5,6,7,8,9,10,11	30	59.01	32.77
	Sensitivity	2,3,4,5,6,7,8,9,10,11,12	30	68.81	22.97
	L2 Norm	1,2,3,4,5,6,7,8,9,10,11	30	59.01	32.77
	Standard Deviation	2,3,4,5,6,7,8,9,10,11,12	30	68.81	22.97
	Kurtosis	1,2,3,4,5,6,7,8,9,10,12	30	58.32	33.46
	Information Entropy	2,3,4,5,6,7,8,9,10,11,12	30	68.81	22.97
	Mutual Information	1,2,3,4,5,6,7,8,9,10,11	30	59.01	32.77

We provide the complete multi-block pruning results under different pruning ratios in Table 7-12. Highlighted rows indicate the best-performing pruning criteria.

A.5 Block-level Weight Distribution

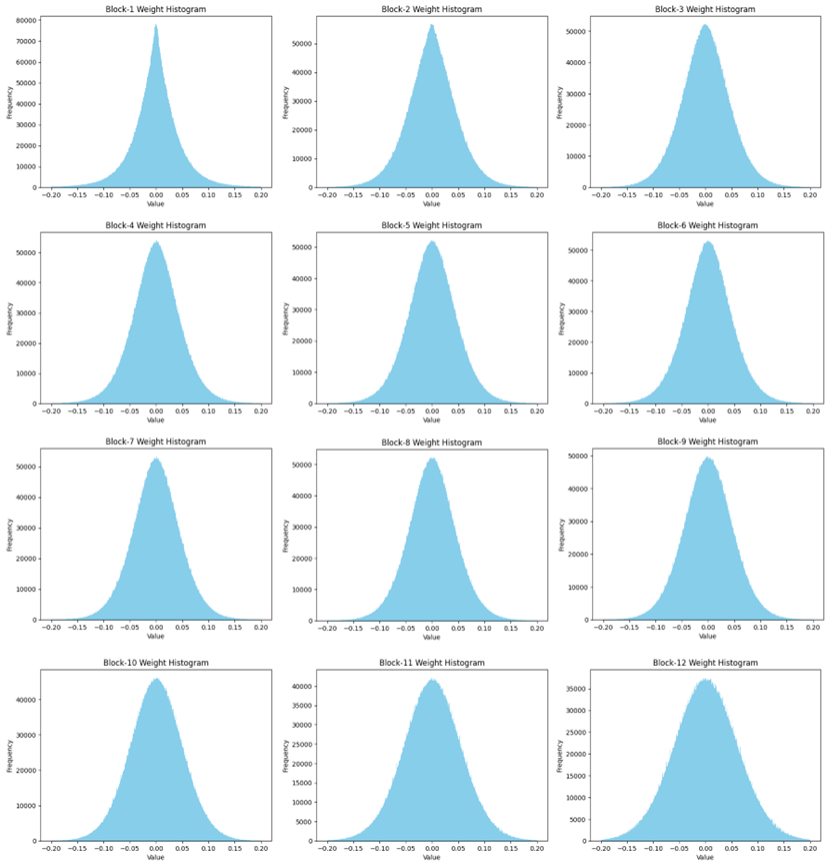


Figure 3: Weight distributions of transformer blocks at different depths in a pretrained VideoMAE encoder. Early blocks exhibit sharply peaked distributions, while deeper blocks show increasingly flatter and more dispersed parameter distributions.