# MAP: Unleashing Hybrid Mamba-Transformer Vision Backbone's Potential with Masked Autoregressive Pretraining

Yunze Liu[1,3] Li Yi[1,2,3]
[1]IIIS, Tsinghua University [2]Shanghai Artificial Intelligence Laboratory [3]Shanghai Qi Zhi Institute

## Abstract

*Hybrid Mamba-Transformer networks have recently garnered broad attention. These networks can leverage the scalability of Transformers while capitalizing on Mamba's strengths in long-context modeling and computational efficiency. However, the challenge of effectively pretraining such hybrid networks remains an open question. Existing methods, such as Masked Autoencoders (MAE) or autoregressive (AR) pretraining, primarily focus on single-type network architectures. In contrast, pretraining strategies for hybrid architectures must be effective for both Mamba and Transformer components. Based on this, we propose Masked Autoregressive Pretraining (MAP) to pretrain a hybrid Mamba-Transformer vision backbone network. This strategy combines the strengths of both MAE and Autoregressive pretraining, improving the performance of Mamba and Transformer modules within a unified paradigm. Experimental results show that the hybrid Mamba-Transformer vision backbone network pretrained with MAP significantly outperforms other pretraining strategies, achieving state-of-the-art performance. We validate the method's effectiveness on both 2D and 3D datasets and provide detailed ablation studies to support the design choices for each component. Code and checkpoints are available at https://github.com/yunzeliu/MAP.*

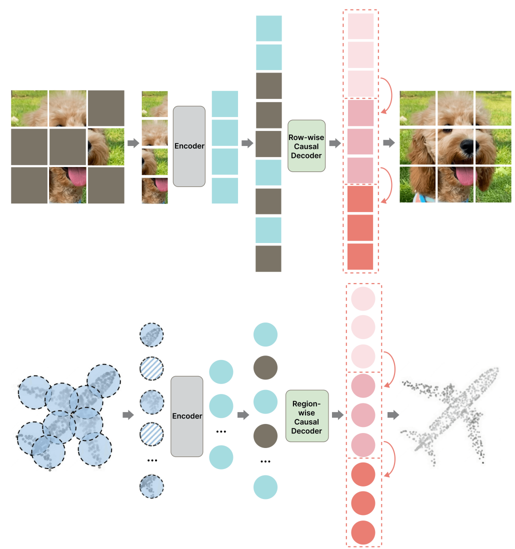


Figure 1. We propose **M**asked **A**utoregressive **P**retraining to pretrain the hybrid Mamba-Transformer backbone. It demonstrates significant performance improvements on both 2D and 3D tasks.

## 1. Introduction

The Mamba-Transformer backbone[4, 15, 22, 42] has recently attracted widespread attention. It leverages the scalability advantages of Transformers and utilizes Mamba's[10, 51] strong capabilities in long-context language modeling. However, to scale up Mamba-Transformer vision backbones, an effective pretraining strategy is essential for maximizing the combined capabilities of Mamba and Transformer. Our work aims to take the first step in this direction.

Developing an effective pretraining strategy for Mamba-Transformer vision backbones is challenging. Although the capability of MAE in pretraining Transformers has been well validated, pretraining methods for Mamba are still underexplored, and the optimal approach remains unclear. Additionally, the hybrid structure requires a pretraining strategy compatible with both computation blocks. This is particularly challenging because the State Space Model captures visual features very differently from Transformers.

To address these challenges, we conducted pilot studies and identified three key observations. Firstly, existing popular pretraining strategies for Transformers, such as MAE[18] and Contrastive Learning[17], do not yield satisfactory results for Mambas. Secondly, Autoregressive Pretraining[37] can be effective for Mamba-based vision backbones with an appropriate scanning pattern and token masking ratio. Thirdly, pretraining strategies suitable for either Mamba or Transformers do not effectively benefit the other.

Based on the above observations, we develop a novel pretraining strategy suitable for the Mamba-Transformer vision backbone named Masked Autoregressive pretraining, or MAP for short. The key is a hierarchical pretraining objective where local MAE is leveraged to learn good local attention for the Transformer blocks while global autoregressive pretraining enables the Mamba blocks to learn meaningful

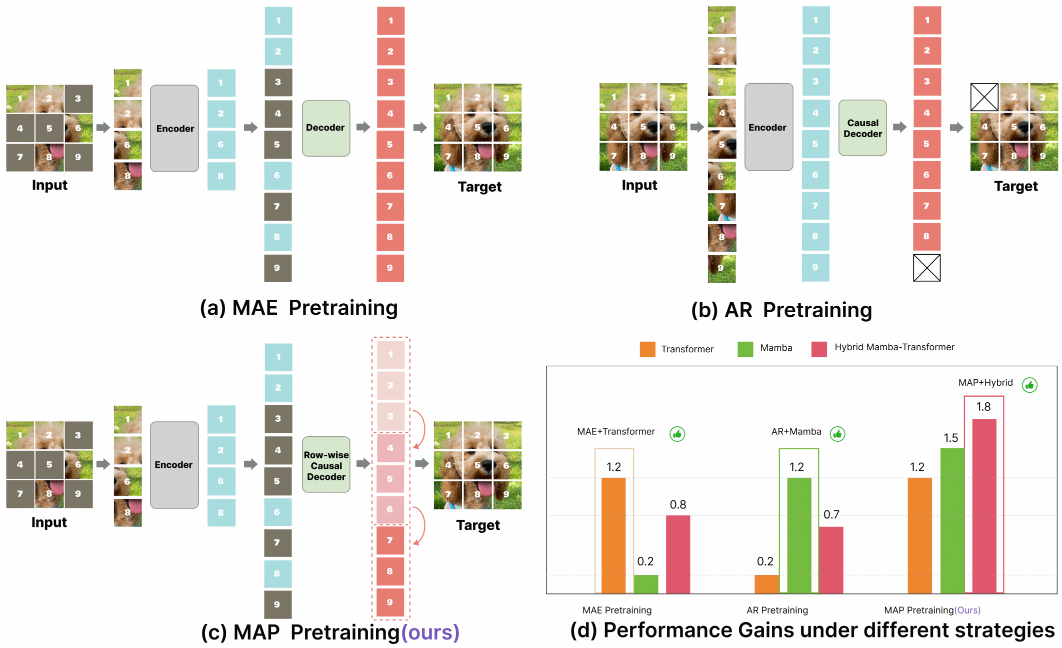


Figure 2. **(a) MAE Pretraining.** Its core lies in reconstructing the masked tokens based on the unmasked tokens to build a global bidirectional contextual understanding. **(b) AR Pretraining.** It focuses on building correlations between contexts, and its scalability has been thoroughly validated in the field of large language models. **(c) MAP Pretraining(ours).** Our method first randomly masks the input image, and then reconstructs the original image in a row-by-row autoregressive manner. This pretraining approach demonstrates significant advantages in modeling contextual features of local characteristics and the correlations between local features, making it highly compatible with the Mamba-Transformer hybrid architecture. **(d) Performance Gains under different pretraining strategies on ImageNet-1K.** We found MAE pretraining is better suited for Transformers, while AR is more compatible with Mamba. MAP, on the other hand, is more suited for the Mamba-Transformer backbone. Additionally, MAP also demonstrates impressive performance when pretraining with pure Mamba or pure Transformer backbones, showcasing the effectiveness and broad applicability of our method.

contextual information. Specifically, the pretraining method is supported by two key designs. First, we leverage local MAE to enable the hybrid framework, particularly the Transformer module, to learn local bidirectional connectivity. This requires the hybrid network to predict all tokens within a local region after perceiving local bidirectional information. Second, we autoregressively generate tokens for each local region to allow the hybrid framework, especially the Mamba module, to learn rich contextual information. This requires the network to autoregressively generate subsequent local regions based on the previously decoded tokens.

Notably, since our focus in this paper is on pretraining strategies for hybrid frameworks rather than the hybrid frameworks themselves, we empirically choose a densely mixed hybrid scheme as the default hybrid framework and refer to it as HybridNet. Our experiments demonstrate that HybridNet pretrained with MAP outperforms other pretraining strategies by a significant margin on the 2D and 3D vision tasks. Extensive ablation studies confirm the rationale and effectiveness of each design choice.

Our contributions are threefold as follows:
**Firstly,** we propose a novel method for pretraining the Hybrid Mamba-Transformer Vision Backbone for the first time, enhancing the performance of hybrid backbones as well as pure Mamba and pure Transformer backbones within a unified paradigm. **Secondly,** For diagnostic purposes, we also conduct an in-depth analysis of the key components of pure-Mamba with autoregressive pretraining, revealing that the effectiveness hinges on maintaining consistency between the pretraining order and the Mamba scanning order, along with an appropriate token masking ratio. **Thirdly,** we demonstrate that our proposed method, MAP, significantly improves the performance of both Mamba-Transformer and pure Mamba backbones across various 2D and 3D datasets. Extensive ablation studies confirm the rationale and effectiveness of each design choice.

## 2. Related Work

**Vision Mambas and Vision Transformers.** Vision Mamba(Vim)[52] is an efficient model for visual representation learning, leveraging bidirectional state space blocks to outperform traditional vision transformers like DeiT in both performance and computational efficiency. The VMamba[26] architecture, built using Visual State-Space blocks and 2D Selective Scanning, excels in visual perception tasks by balancing efficiency and accuracy. Autoregressive pretraining(ARM)[37] further boosts Vision Mamba's performance, enabling it to achieve superior accuracy and faster training compared to conventional supervised models. Nevertheless, why autoregression is effective for Vision Mamba and what the key factors are remains an unresolved question. In this paper, we explore the critical design elements behind the success of Mamba's autoregressive pretraining for the first time. The key difference between our proposed MAP and ARM[37] lies in our use of random masking in the AR process, where the network decodes the next complete local information based on the masked local information. The masking mechanism enhances the modeling capability of local information and the relevance of context. Vision Transformers(ViT)[7] adapt transformer architectures to image classification by treating image patches as sequential tokens. Swin Transformer[27] introduces a hierarchical design with shifted windows, effectively capturing both local and global information for image recognition. MAE [18] enhances vision transformers through self-supervised learning, where the model reconstructs masked image patches using an encoder-decoder structure, enabling efficient and powerful pretraining for vision tasks. However, the MAE pretraining strategy is not effective for Mamba, which hinders our ability to pretrain the hybrid Mamba-Transformer backbones. Although the hybrid architecture using MAP still struggles to surpass the Transformer using MAE under the same settings, it provides a balance between computational cost and performance, as well as the potential to introduce longer contexts while maintaining scalability. This property is particularly important in the video domain and large language models, and we leave it for future work.

**Self-Supervised Visual Representation Learning.** Self-Supervised Visual Representation Learning is a machine learning approach that enables the extraction of meaningful visual features from large amounts of unlabeled data. This methodology relies on pretext tasks, which serve as a means to learn representations without the need for explicit labels. GPT-style AR[13] models predict the next part of an image or sequence given the previous parts, encouraging the model to understand the spatial or temporal dependencies within the data. MAE[18] methods mask out random patches of an input image and train the model to reconstruct these masked regions. This technique encourages the model to learn contextual information and global representations. Contrastive Learning(CL)[17] techniques involve contrasting positive and negative samples to learn discriminative features. It typically involves creating pairs of positive and negative examples and training the model to distinguish between them. However, we found that existing pretraining strategies fail to fully unlock the potential of the hybrid framework, which motivated us to explore a new pretraining paradigm for hybrid Mamba-Transformer backbones. In this paper, we propose MAP to pretrain hybrid architectures. On the pure Mamba architecture, our method outperforms AR, due to MAP's ability to effectively model local features and the associations between local regions. On the pure Transformer architecture, although MAP does not surpass the performance of MAE, it still achieves comparable results. This is because, while local MAE reduces the receptive field, the autoregressive modeling between local regions enhances the ability to capture local relationships, ensuring that MAP maintains strong performance on Transformers.

## 3. Polit Study

Before exploring the strategies for pretraining hybrid frameworks, we first analyze how to properly pretrain Transformers and Mamba respectively. Analyzing the pretraining strategies for them helps us gain a deeper understanding of the key aspects of training hybrid frameworks. In this Section, we first conduct experiments to investigate the differences in pretraining strategies for ViT and Vim, which are representative of visual Transformers and visual Mamba. The success of the MAE strategy on the ViT architecture is well acknowledged, while the Vim pretraining strategy remains in its early stages. We are interested in determining whether the MAE strategy is equally applicable to Vim or if the AR strategy is more suitable. To explore this, we conduct experiments on the classification task using the ImageNet-1K dataset. The results are shown in Table 1.

| Method | ViT | ViT+MAE | ViT+AR | ViT+CL |
|---|---|---|---|---|
| Accuracy | 82.3 | **83.6(+1.4)** | 82.5(+0.2) | 82.5(+0.2) |
| Method | Vim | Vim+MAE | Vim+AR | Vim+CL |
| Accuracy | 81.2 | 81.4(+0.2) | **82.6(+1.4)** | 81.1(-0.1) |

Table 1. Pilot Study. We use ViT-B and Vim-B as the default configurations. The AR strategy processes the image tokens in a row-first order, while the MAE operates according to the default settings. For contrastive learning, we only used crop and scale data augmentation and used the MoCov2 for pretraining. All experiments are conducted at a resolution of 224x224. The number of mask tokens for AR is set to 40 tokens (20%).

Experiments show that MAE is more suitable for Transformer pretraining, while AR is better suited for Mamba pretraining. We observe that the MAE strategy significantly enhances the performance of ViT. However, for Vim, the MAE strategy does not yield the expected improvements, while the AR strategy substantially boosts its performance. This indicates that for the ViT, applying the MAE strategy is essential to establish bidirectional associations between tokens, thereby improving performance. In contrast, for Vim, it is more important to model the continuity between preceding and succeeding tokens. Therefore, for the hybrid framework, we intuitively need a strategy that can leverage the advantages of MAE in bidirectional modeling while retaining the strengths of AR in context modeling. Since MAE's pretraining on Transformers is already well-established, benefiting from its asymmetric network design and high masking ratio, the key to AR's success on Mamba remains unclear. Therefore, in the following sections, we analyze the critical factors behind AR's success on Mamba, which will help us design strategies for the hybrid framework later.

We conducted an in-depth analysis and discovered that consistent autoregression pretraining with scanning order and proper masking ratio is the key to pretraining Mamba. This is crucial for diagnosing how to pretrain Mamba and for designing pretraining strategies for the hybrid backbone.

**Relationship between AR and Scanning Order.** Different prediction orders can significantly impact how the model captures image features and the effectiveness of sequence generation. The analysis of the role of prediction order will help optimize AR pretraining for Vim, exploring how the model can better capture the continuity and relationships of image information under different contextual conditions. We conduct ablation studies on Vim by allowing it to perform both row-first and column-first scanning. We then pretrain it with row-first and column-first AR orders, respectively, to compare their performance. Figure 3 shows different orders for AR pretraining and Mamba scanning.

| Method | Vim(R) | Vim(R) + AR(C) | Vim(R) + AR(R) |
|---|---|---|---|
| Accuracy | 79.7 | 79.9(+0.2) | **82.6(+2.9)** |
| Method | Vim(C) | Vim(C) + AR(C) | Vim(C) + AR(R) |
| Accuracy | 79.5 | **82.5(+3.0)** | 79.9(+0.4) |

Table 2. The impact of AR order on downstream tasks. Vim(R) refers to Vim with row-first scanning. Vim(C) refers to Vim with column-first scanning. AR(R) refers to row-first autoregressive pretraining. AR(C) refers to column-first autoregressive pretraining. The results indicate that the best performance is achieved when the AR pretraining design aligns with Mamba's scanning order.

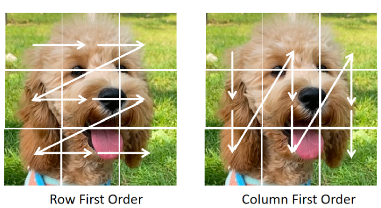


Figure 3. Different orders for AR pretraining and Mamba scanning. The row-first and column-first orders allow the network to perceive local information in different ways and sequences.

The results are shown in Table 2. We observe that employing a pretraining strategy consistent with the scanning order significantly enhances Vim's performance. This suggests that when designing pretraining strategies, they should be aligned with the downstream scanning order. This conclusion is also implicitly confirmed in ARM[37], as it uses cluster-based AR to pretrain cross-scanned Mamba, which is essentially a combination of row-first and column-first approaches. Unlike ARM, we experimentally verified this and provided comprehensive experimental results.

**Masking Ratio of Autoregression Pretraining.** Masking a single token follows the traditional AR paradigm, while masking $n$ tokens transforms the task into an inpainting problem, as the input and output sequence lengths remain equal. In this context, varying the auto-regressive masking ratios effectively adjusts the inpainting ratio, influencing the model's predictions beyond just the sequence length.

| Masked tokens | 1 (0.5%) | 20 (10%) | 40 (20%) |
|---|---|---|---|
| Accuracy | 81.7 | 82.0 | **82.6** |
| Masked tokens | 60 (30%) | 100 (50%) | 140 (70%) |
| Accuracy | 82.5 | 82.2 | 81.9 |

Table 3. The impact of Masking Ratio on AR pretraining. We masked 1 token (0.5%), 20 tokens (10%), 40 tokens (20%), 60 tokens (30%), 100 tokens (50%), and 140 tokens (70%), respectively, while also recording the results of fine-tuning on downstream tasks. The experiment shows that an appropriate masking ratio is important for autoregressive pretraining.

The results are shown in Table 3. In auto-regressive pretraining, as the Masking Ratio increases, the performance of the Mamba improves. This is because a higher Masking Ratio encourages the model to learn more complex and rich feature representations, thereby enhancing its generative ability and adaptability. However, an excessively high Masking Ratio may lead to instability during the training process. In such cases, the model may struggle to make accurate predictions due to a lack of sufficient contextual information, negatively impacting its pretraining effectiveness. Therefore, when designing auto-regressive pretraining tasks, finding an appropriate masking ratio is crucial to strike a balance between performance improvement and training stability.

Now we have the following three conclusions to serve as a reference for designing the pretraining of hybrid backbone:

- MAE is more suitable for Transformers, while AR is better suited for Mamba.
- For MAE pretraining of Transformers, an asymmetric structure, and an appropriate masking ratio are important.
- For AR pretraining of Mamba, an appropriate AR order and masking ratio are important.

Based on the above conclusions, we propose Masked Autoregressive Pretraining (MAP) to pretrain hybrid frameworks. MAP leverages AR to model the relationships between local regions and local MAE to model the internal features of local regions. It also employs a masking strategy that is crucial for both MAE and AR to enhance the network's representation capabilities. This strategy combines the strengths of MAE in local feature modeling, which is suitable for pretraining Transformers, and the strengths of AR in context modeling, which is suitable for pretraining Mamba. Effectively inheriting the advantages of both strategies, MAP improves the pretraining performance of hybrid architectures within a unified framework. In the following section, we will provide a detailed introduction to the MAP pretraining strategy.

## 4. Method

**Preliminary.**The focus of this paper is on studying how to pretrain hybrid Mamba-Transformer frameworks, rather than on designing the hybrid frameworks themselves. Therefore, we first determine a hybrid network as the default of our study. We tried a series of hybrid Mamba-Transformer vision backbones and compared their performance when trained from scratch. The results indicate that the hybrid approach using MMMTMMMT performs the best. When comparing Mamba-R* with MMMMMMTT, we found that adding a Transformer after Mamba enhances its long-context modeling capabilities, leading to improved performance. However, when comparing MMMMMMTT with TTMMMMMM, we observed that simply appending Transformers after Mamba does not fully leverage the architecture's potential. This suggests that incorporating Transformers at the beginning is crucial for extracting sufficient local features. We believe that the MMMTMMMT approach effectively balances local feature extraction and contextual modeling enhancement, making it our default configuration. We adopt the network shown in Figure 4(d) as our default hybrid network and refer to it as **HybridNet**. It is worth noting that our method can also be applied to other Mamba-Transformer hybrid frameworks, but the design of the hybrid frameworks themselves is not within the scope of this paper.

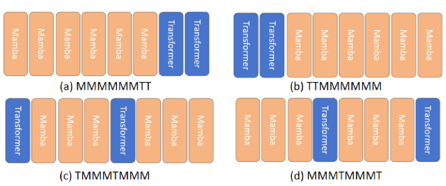


Figure 4. Different Hybrid Model Design. (d) achieves the best results and is set as default and refer to it as **HybridNet**.

| Design | DeiT* | Mamba-R* | MMMMMMTT |
|---|---|---|---|
| Accuracy | 82.80 | 82.70 | 82.88 |
| Design | TTMMMMMM | TMMMTMMM | MMMTMMMT |
| Accuracy | 82.93 | 83.01 | **83.12** |

Table 4. Hybrid Design of Mamba-Transformer backbone. All experiments are trained from scratch. Mamba-R* means 24 Mamba-R[40] Mamba layers plus 8 additional Mamba layers. DeiT* means 24 DeiT[39] Transformer layers plus 8 additional Transformer layers. MMMMMMTT represents 24 Mamba layers followed by 8 Transformer layers. TTMMMMMM represents 8 Transformer layers followed by 24 Mamba layers. TMMMTMMM represents a unit consisting of 1 Transformer layer and 3 Mamba layers, repeated 8 times. MMMTMMMT represents a unit of 3 Mamba layers followed by 1 Transformer layer, repeated 8 times.

**Overview.** Our proposed MAP is a pre-training strategy for the hybrid Mamba-Transformer framework that can significantly improve the performance of the hybrid architecture compared to traditional MAE and AR pre-training. Our MAP pre-training is to perform random masking on a given image and then region-wise autoregressively reconstruct its original image. Here, the tokens of each row of the image are predicted at the same time, and the tokens between rows are predicted by autoregression.

Notably, we choose each row as a sub-region here because, as mentioned in the previous chapter, the AR order of Mamba should match the scanning order. The default scanning order for most Mamba implementations is row-first. To ensure a fairer comparison with existing methods, we treat each row as a prediction unit. After changing the scanning order of Mamba, more sophisticated clustering strategies are theoretically expected to yield better results.

As shown in Figure 2(c), given an image, HybridNet first maps the randomly masked image into the feature space and then uses a Transformer Decoder to decode the original image row-wise. Consider an image $\mathbf{I}$ which is divided into rows:

$$\mathbf{I} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_M\} \tag{1}$$

Each row $\mathbf{r}_i$ consists of tokens:

$$\mathbf{r}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iN}\} \tag{2}$$

Select a subset of tokens to mask within each row. Let $\mathbf{M}_i \subset \{1, 2, \ldots, N\}$ denote the indices of the masked tokens in row $\mathbf{r}_i$.

For a given row $\mathbf{r}_i$, predict all masked tokens simultaneously:

$$p(\mathbf{x}_{ij} \mid \mathbf{x}_{i,j\notin\mathbf{M}_i}, \mathbf{r}_{<i}) \tag{3}$$

where $\mathbf{r}_{<i}$ refers to all rows preceding row $i$.

The prediction of tokens in row $i$ depends on all previous rows and the visible tokens within the row. This can be expressed as:

$$p(\mathbf{r}_i \mid \mathbf{r}_{<i}) = \prod_{j=1}^{N} p(\mathbf{x}_{ij} \mid \mathbf{x}_{i,j\notin\mathbf{M}_i}, \mathbf{r}_{<i}) \tag{4}$$

The overall loss function is the sum of the negative log-likelihoods of the predicted tokens:

$$\mathcal{L} = -\sum_{i=1}^{M} \sum_{j\in\mathbf{M}_i} \log p(\mathbf{x}_{ij} \mid \mathbf{x}_{i,j\notin\mathbf{M}_i}, \mathbf{r}_{<i}) \tag{5}$$

Our proposed MAP represents a general paradigm applicable to data across various domains, with 2D image data as an example. Our method can be easily extended to large language models and the fields of image video and point cloud video. Our method optimizes the synergy between Mamba and Transformer within a unified framework, allowing both models to fully leverage their strengths. In the Mamba-Transformer hybrid architecture, this approach effectively enhances the cooperation between the models, resulting in significant performance improvements.

For HybridNet, the masking mechanism of MAP can significantly enhance the local modeling capabilities of both the Mamba and Transformer layers. The region-wise autoregressive strategy of MAP can significantly improve Mamba's ability to model context. Therefore, overall, MAP can bring about substantial performance improvements for HybridNet. For pure Mamba architecture, our method outperforms AR, due to MAP's ability to effectively model local features and the associations between local regions. On the pure Transformer architecture, although MAP does not surpass the performance of MAE, it still achieves comparable results. This is because, while local MAE reduces the receptive field, the autoregressive modeling between local regions enhances the ability to capture local relationships, ensuring that MAP maintains strong performance on Transformers.

We will introduce the specific design components of the framework. The subsequent experiments in this section are conducted using the base-sized model on the ImageNet-1K dataset.

**Masking.** Since masking is also crucial for MAE pretraining of Transformers and AR pretraining of Mamba, masking is an important design aspect of MAP. We experimented with different masking strategies, including random, sequential, and diagonal masking. Our experiments show that random masking delivers the best results. We attribute this to the fact that sequential and diagonal masking can hinder the Transformer's ability to establish contextual relationships. Random masking not only promotes bidirectional modeling for Transformers but also enhances Mamba's generalization and representation capabilities in sequence modeling. The random masking strategy is actually a balance, as it retains the bidirectional modeling capability of Transformers while leveraging row-wise autoregressive pretraining to train Mamba. The raster and diagonal strategies fail to fully utilize the bidirectional modeling capability of Transformers and do not enhance Mamba's ability to model local features.

Additionally, we explored the effects of different masking ratios and found that a 50% masking ratio yielded the best results. This conclusion aligns with intuition: while MAE performs optimally on Transformers with a 75% masking ratio, previous experiments showed that AR achieves the best results on Mamba with a 20% ratio. Therefore, a 50% ratio serves as a balanced number, leveraging the strengths of both paradigms. This ratio is significantly different from the conclusions drawn in MAE.

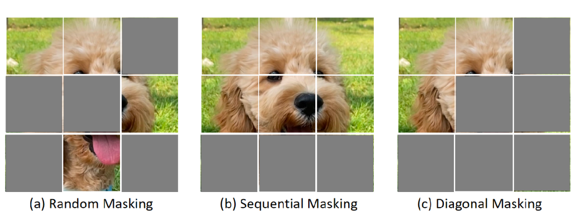


Figure 5. Different Masking Strategies. The random masking strategy produces the best results.

| Strategy | FS | Random | Sequential | Diagonal |
|---|---|---|---|---|
| Accuracy | 83.1 | **84.9** | 84.0 | 83.8 |
| Ratio | **0%** | **25%** | **50%** | **75%** |
| Accuracy | 83.3 | 84.5 | **84.9** | 84.2 |

Table 5. Random masking with a 50% ratio performs the best.

**MAP Transformer Decoder.** To reconstruct the original image, we utilize a masked Transformer for signal recovery. The reason for using a Transformer decoder rather than a Mamba decoder is that the Transformer decoder can reconstruct region-wise based on the encoder's features by applying a decoder mask. In contrast, the Mamba decoder, due to its unidirectional scanning nature, struggles to simultaneously reconstruct an entire local region. Our decoder employs a distinct row-wise decoding strategy that allows autoregressive decoding of one row of tokens at a time, enhancing the network's ability to capture local features and contextual relationships among regions. Experiments show that this method significantly outperforms the original AR, MAE, and local MAE decoding strategies. Local MAR refers to limiting the receptive field of MAE within each row of the image. Notably, in the hybrid framework, local MAE performs comparably to standard MAE, emphasizing the significance of local feature learning. Our MAP method improves local feature modeling while leveraging autoregressive techniques to capture contextual relationships across regions, resulting in superior performance.


Figure 6. Different Decoder Mask. Green represents activation. White represents non-activation.

| Decoder Mask | AR | MAE | local MAE | ours |
|---|---|---|---|---|
| Accuracy | 83.7 | 84.1 | 84.2 | **84.9** |

Table 6. Decoder Mask Design. Our MAP decoder strategy achieves the best results.

**Reconstruction Target.** Consistent with MAE, we reconstructed normalized original pixels as the target and employed MSE loss. Inspired by MAR[19] to use reconstruction output as a conditional signal for diffusion models to improve generation quality, we explored whether pretraining with diffusion loss could enhance performance. However, this approach did not yield significant improvements, suggesting that the quality of reconstructed images is not directly linked to encoder pretraining success.

| Loss | FS | Diffusion Loss | MSE Loss (ours) |
|---|---|---|---|
| Accuracy | 83.1 | 83.3 | **84.9** |

Table 7. Reconstruction Target.

# 5. Experiments

## 5.1. 2D Classification on ImageNet-1k.

**Settings.** We pretrained on the training set of the ImageNet-1K[6] dataset and then fine-tuned on its classification task. We report the top-1 validation accuracy of a single 224x224 crop, and in some settings, we also report the results for a 384x384 crop. During the pretraining phase, we applied a random masking strategy with a 50% masking ratio, using only random cropping as the data augmentation strategy. We utilized AdamW as the optimizer and trained for 1600 epochs across all settings. In the fine-tuning phase, we directly fine-tune for 400 epochs and report the results.

| Model | Img. size | #Params | Throughput | Mem | Acc. (%) |
|---|---|---|---|---|---|
| ***Pure Convolutional networks:*** | | | | | |
| ResNet-50 [16] | $224^2$ | 25M | 2388 | 6.6G | 76.2 |
| ResNet-152 [16] | $224^2$ | 60M | 1169 | 12.5G | 78.3 |
| EfficientNet-B3 [38] | $300^2$ | 12M | 496 | 19.7G | 81.6 |
| ConvNeXt-T [28] | $224^2$ | 29M | 701 | 8.3G | 82.1 |
| ConvNeXt-S [28] | $224^2$ | 50M | 444 | 13.1G | 83.1 |
| ConvNeXt-B [28] | $224^2$ | 89M | 334 | 17.9G | 83.8 |
| ***Pure Vision Transformers:*** | | | | | |
| ViT-B/16 [8] | $224^2$ | 86M | 284 | 63.8G | 77.9 |
| ViT-L/16 [8] | $224^2$ | 307M | 149 | - | 76.5 |
| ***Pretrained Vision Transformers:*** | | | | | |
| ViT-B/16 + AR [8] | $224^2$ | 86M | 284 | 63.8G | 82.5 |
| ViT-B/16 + MAE [8] | $224^2$ | 86M | 284 | 63.8G | 83.6 |
| ViT-L/16 + MAE [8] | $224^2$ | 307M | 149 | - | 85.9 |
| ViT-B/16 + MAP | $224^2$ | 86M | 284 | 63.8G | 83.6 |
| ViT-L/16 + MAP | $224^2$ | 307M | 149 | - | 86.1 |
| ***Pure Mamba architecture:*** | | | | | |
| Vim-T [52] | $224^2$ | 7M | 1165 | 4.8G | 76.1 |
| Vim-S [52] | $224^2$ | 26M | 612 | 9.4G | 80.5 |
| MambaR-T [40] | $224^2$ | 9M | 1160 | 5.1G | 77.4 |
| MambaR-S [40] | $224^2$ | 28M | 608 | 9.9G | 81.1 |
| MambaR-B [40] | $224^2$ | 99M | 315 | 20.3G | 82.9 |
| MambaR-L [40] | $224^2$ | 341M | 92 | 55.5G | 83.2 |
| ***Pretrained Mamba architecture:*** | | | | | |
| ARM-B (Mamba+AR) [37] | $224^2$ | 85M | 325 | 19.7G | 83.2 |
| ARM-L (Mamba+AR) [37] | $224^2$ | 297M | 111 | 53.1G | 84.5 |
| MambaR-B+MAE | $224^2$ | 99M | 315 | 20.3G | 83.1 |
| MambaR-B+AR | $224^2$ | 99M | 315 | 20.3G | 83.7 |
| MambaR-B+MAP | $224^2$ | 99M | 315 | 20.3G | 84.0 |
| MambaR-L+MAP | $224^2$ | 341M | 92 | 55.5G | 84.8 |
| ***Hybrid 2D convolution + Mamba:*** | | | | | |
| VMamba-T [26] | $224^2$ | 31M | 464 | 7.6G | 82.5 |
| VMamba-S [26] | $224^2$ | 50M | 313 | 27.6G | 83.6 |
| VMamba-B [26] | $224^2$ | 89M | 246 | 37.1G | 83.9 |
| ***Hybrid 2Dconvolution + Mamba + Transformer architecture: (with down-sampling)*** | | | | | |
| MambaVision-T [15] | $224^2$ | 35M | 1349 | 10.7G | 82.7 |
| MambaVision-S [15] | $224^2$ | 51M | 1058 | 36.6G | 83.3 |
| MambaVision-B [15] | $224^2$ | 97M | 826 | 50.8G | 84.2 |
| MambaVision-L [15] | $224^2$ | 241M | 229 | 78.6G | 85.3 |
| MambaVision-B+MAP [15] | $224^2$ | 97M | 826 | 50.8G | 84.9 |
| MambaVision-L+MAP [15] | $224^2$ | 241M | 229 | 78.6G | 86.4 |
| ***Hybrid Mamba + Transformer architecture: (without down-sampling)*** | | | | | |
| HybridNet-T | $224^2$ | 12M | 910 | 7.6G | 77.7 |
| HybridNet-S | $224^2$ | 37M | 512 | 14.6G | 81.3 |
| HybridNet-B | $224^2$ | 128M | 244 | 30.0G | 83.1 |
| | $384^2$ | 128M | 244 | 76.1G | 84.5 |
| HybridNet-L | $224^2$ | 443M | 63 | 78.3G | 83.2 |
| | $384^2$ | 443M | 63 | - | 84.6 |
| ***Pretrained Hybrid architecture:*** | | | | | |
| HybridNet-T + MAP | $224^2$ | 12M | 910 | 7.6G | 78.6 |
| HybridNet-S + MAP | $224^2$ | 37M | 512 | 14.6G | 82.5 |
| HybridNet-B + MAE | $224^2$ | 128M | 244 | 30.0G | 83.9 |
| HybridNet-B + AR | $224^2$ | 128M | 244 | 30.0G | 83.8 |
| HybridNet-B + CL | $224^2$ | 128M | 244 | 30.0G | 83.1 |
| HybridNet-B + MAP | $224^2$ | 128M | 244 | 30.0G | 84.9 |
| | $384^2$ | 128M | 244 | 76.1G | 85.5 |
| HybridNet-L + MAP | $224^2$ | 443M | 63 | 78.3G | 85.0 |
| | $384^2$ | 443M | 63 | - | 86.2 |

Table 8. ImageNet-1k classification results. The throughput is computed on an A100 GPU. The memory overhead is measured with a batch size of 128. Our results are highlighted in blue.

**Results.** Results are shown in Table 8. The results indicate that the hybrid framework achieves a balance between performance and computational overhead. The focus of this paper is not on designing the hybrid network but on investigating its pretraining methods. Our proposed pretraining method MAP significantly enhances the performance of the hybrid Mamba-Transformer framework. Comparing the performance of MAP, MAE, AR, and CL on HybridNet-B, it is evident that MAP significantly improves performance. This indicates that compared to MAE, the local MAE in MAP, which enhances local modeling and adopts an autoregressive approach to strengthen the associations between local regions, is crucial for hybrid networks. Compared to AR, the masking mechanism and the local autoregressive strategy in MAP contribute to performance gains. Additionally, we verified that our MAP method can improve the performance of the pure Mamba framework and the pure Transformer backbone. By comparing AR, MAE, and MAP pretraining results on MambaR-B, we can see that MAE offers limited improvement for pure Mamba. AR, benefiting from its autoregressive modeling of context, shows improved performance. Compared to AR, the masking mechanism and local autoregressive decoding strategy in MAP enhance Mamba's ability to model local features, further improving performance. By comparing the performance of AR, MAE, and MAP on ViT, we find that on ViT-B, MAP can achieve the same results as MAE. This is because, although local MAE reduces the receptive field compared to the original MAE, region-wise autoregressive modeling not only enhances local feature modeling but also strengthens the modeling of correlations between local regions, preventing a decline in performance. When comparing MAE and MAP on ViT-L, the advantage of region-wise autoregressive modeling in MAP becomes apparent in larger-scale models, where it outperforms MAE. This also suggests that autoregressive modeling has advantages in larger models, which is consistent with the scaling law phenomenon observed in LLMs. In addition, by comparing the results of HybridNet-B/L at 224 and 384 resolutions, we can see that at the longer sequence length of 384 resolution, our method further achieved improvements of 0.6 and 0.8, respectively. This indicates that the longer context modeling allowed by the Mamba layer can indeed lead to performance gains. Comparing the results of MambaVision-B/L before and after MAP pretraining shows that our MAP method not only benefits the HybridNet used as an example but also brings significant improvements to other existing hybrid Mamba-Transformer network frameworks.

## 5.2. Transfer Learning Experiments

**Semantic Segmentation** We conduct experiments for semantic segmentation on the ADE20K [50] and use UperNet [45] as the segmentation framework followed by Vim. All settings are consistent with Vim. When comparing HybridNet-S with

| Method | Backbone | image size | #param. | *val* mIoU |
|---|---|---|---|---|
| DeepLab v3+ | ResNet-101 | $512^2$ | 63M | 44.1 |
| UperNet | ResNet-50 | $512^2$ | 67M | 41.2 |
| UperNet | ResNet-101 | $512^2$ | 86M | 44.9 |
| UperNet | DeiT-Ti | $512^2$ | 11M | 39.2 |
| UperNet | DeiT-S | $512^2$ | 43M | 44.0 |
| UperNet | Vim-Ti | $512^2$ | 13M | 41.0 |
| UperNet | Vim-S | $512^2$ | 46M | 44.9 |
| UperNet | HybridNet-S | $512^2$ | 69M | 45.6 |
| UperNet | HybridNet-S+MAP | $512^2$ | 69M | 46.9 |

Table 9. Results of semantic segmentation on the ADE20K *val* set.

| Backbone | $AP^{box}$ | $AP^{mask}$ |
|---|---|---|
| DeiT-Ti | 44.4 | 38.1 |
| Vim-Ti | 45.7 | 39.2 |
| HybridNet-Ti | 45.9 | 39.2 |
| HybridNet-Ti+MAP | 46.4 | 39.8 |

Table 10. Results of object detection and instance segmentation on the COCO *val* set using Cascade Mask R-CNN [1] framework.

Vim-S, we can see that introducing a Transformer significantly enhances the performance of dense prediction tasks. This highlights the value of hybrid networks, which maintain satisfactory performance when introducing longer contexts at 512x512 resolution. Comparing the results of HybridNet-S before and after MAP pretraining, we can see that our pretraining method also brings significant performance improvements in segmentation tasks.

**Object Detection and Instance Segmentation** We conduct experiments for object detection and instance segmentation on the COCO 2017 dataset [23] and use ViTDet [45] as the basic framework followed by Vim. All settings are consistent with Vim. The results indicate that, while simply inserting a Transformer into Vim does not bring about significant improvements, the performance of HybridNet-Ti, when fine-tuned after initialization with MAP pretraining, shows a notable improvement.

## 5.3. 3D Classification Experiments

**Settings.** To verify that our method can be applied to other data formats, we transferred the MAP pretraining to 3D data for experimentation. We pretrained using the ShapeNet[2] dataset, employing random rotation and translation scaling as data augmentation techniques. Each point cloud consists of 1024 points and is divided into 64 patches, with each patch containing 32 points. Since point clouds are unordered, the concept of rows does not apply here; instead, we randomly generate 32 patches each time and complete the reconstruction process in an autoregressive manner. Similar to Mamba3D[14], we did not adopt any special sorting strategies but ensured that the order of pretraining matches that of the actual Mamba scans. We conducted pretraining on both the hybrid framework and the original Mamba3D to validate their performance advantages. During pretraining and downstream fine-tuning, we employed the AdamW optimizer with a cosine decay strategy for 300 epochs.

| Method | PT | #P ↓ | #F ↓ | ModelNet40 |
|---|---|---|---|---|
| *Supervised Learning Only: Dedicated Architectures* | | | | |
| PointNet[32] | × | 3.5 | 0.5 | 89.2 |
| PointNet++[33] | × | 1.5 | 1.7 | 90.7 |
| DGCNN[43] | × | 1.8 | 2.4 | 92.9 |
| PointCNN[20] | × | 0.6 | - | 92.2 |
| DRNet [35] | × | - | - | 93.1 |
| SimpleView[9] | × | - | - | 93.9 |
| GBNet[36] | × | 8.8 | - | 93.8 |
| PRA-Ne[5] | × | - | 2.3 | 93.7 |
| MVTN[12] | × | 11.2 | 43.7 | 93.8 |
| PointMLP[29] | × | 12.6 | 31.4 | 94.5 |
| PointNeXt[34] | × | 1.4 | 3.6 | 94.0 |
| P2P-HorNet[44] | ✓ | - | 34.6 | 94.0 |
| DeLA[3] | × | 5.3 | 1.5 | 94.0 |
| *Supervised Learning Only: Transformer or Mamba-based Models* | | | | |
| Transformer | × | 22.1 | 4.8 | 91.4 |
| PCT[11] | × | 2.9 | 2.3 | 93.2 |
| PointMamba[21] | × | 12.3 | 3.6 | - |
| PCM[49] | × | 34.2 | 45.0 | 93.4±0.2 |
| SPoTr[31] | × | 1.7 | 10.8 | - |
| PointConT[25] | × | - | - | 93.5 |
| Mamba3d w/o vot.[14] | × | 16.9 | 3.9 | 93.4 |
| Mamba3d w/ vot.[14] | × | 16.9 | 3.9 | 94.1 |
| HybridNet w/o vot. | × | 19.3 | 4.4 | 93.5 |
| HybridNet w/ vot. | × | 19.3 | 4.4 | 94.3 |
| *With Self-supervised pretraining* | | | | |
| Transformer | *OcCo[41]* | 22.1 | 4.8 | 92.1 |
| Point-BERT | *IDPT[47]* | 22.1+1.7$^{\dagger}$ | 4.8 | 93.4 |
| MaskPoint | *MaskPoint[24]* | 22.1 | 4.8 | 93.8 |
| PointMamba | *Point-MAE[30]* | 12.3 | 3.6 | - |
| Point-MAE | *IDPT[47]* | 22.1+1.7$^{\dagger}$ | 4.8 | 94.4 |
| Point-M2AE | *Point-M2AE[48]* | 15.3 | 3.6 | 94.0 |
| Mamba3d w/o vot. | *Point-BERT[46]* | 16.9 | 3.9 | 94.4 |
| Point-MAE | *Point-MAE[30]* | 22.1 | 4.8 | 93.8 |
| Mamba3d w/o vot. | *Point-MAE[30]* | 16.9 | 3.9 | 94.7 |
| Mamba3d w/ vot. | *Point-MAE[30]* | 16.9 | 3.9 | 95.4 |
| Mamba3d w/o vot. | *MAP* | 16.9 | 3.9 | 95.1 |
| Mamba3d w/ vot. | *MAP* | 16.9 | 3.9 | 95.6 |
| HybridNet w/o vot. | *MAP* | 19.3 | 4.4 | 95.4 |
| HybridNet w/ vot. | *MAP* | 19.3 | 4.4 | 95.9 |

Table 11. Results on 3D ModelNet classification tasks.

**Results.**The experiments demonstrate that our method also can improve the performance on 3D tasks. This suggests that our approach can be easily adapted to other domains. Comparing HybridNet and Mamba3d at the backbone network level, it is evident that the performance improvement solely from inserting a Transformer is limited. By comparing the results of HybridNet before and after MAP pretraining, we can see that our MAP pretraining brings about a significant improvement. This further validates the effectiveness of our proposed pretraining method. More experiments on 3D datasets can be found in the Supplementary Material.

# 6. Conclusion

In this paper, we propose Masked Autoregressive Pretraining, a pretraining framework suitable for hybrid Mamba-Transformer models. It integrates the advantages of MAE and AR, retaining their core designs to model local features while strengthening the modeling of contextual correlations. Extensive experiments have validated the effectiveness of our method and the rationality of its component design.

## References


[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43 (5):1483–1498, 2019.

[2] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

[3] Binjie Chen, Yunzhou Xia, Yu Zang, Cheng Wang, and Jonathan Li. Decoupled local aggregation for point cloud learning. *arXiv preprint arXiv:2308.16532*, 2023.

[4] Wenchao Chen, Liqiang Niu, Ziyao Lu, Fandong Meng, and Jie Zhou. Maskmamba: A hybrid mamba-transformer model for masked image generation. *arXiv preprint arXiv:2409.19937*, 2024.

[5] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Trans. Image Process. (TIP)*, 30:4436–4448, 2021.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[9] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 3809–3820. PMLR, 2021.

[10] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[11] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: point cloud transformer. *Comput. Vis. Media*, 7(2):187–199, 2021.

[12] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. MVTN: multi-view transformation network for 3d shape recognition. In *Int. Conf. Comput. Vis. (ICCV)*, pages 1–11. IEEE, 2021.

[13] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[14] Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. *arXiv preprint arXiv:2404.14966*, 2024.

[15] Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. *arXiv preprint arXiv:2407.08083*, 2024.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

[19] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

[20] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, pages 828–838, 2018.

[21] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[22] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *Proc. of European Conference on Computer Vision*, 2022.

[25] Yahui Liu, Bin Tian, Yisheng Lv, Lingxi Li, and Fei-Yue Wang. Point cloud classification using content-based transformer via clustering in feature space. *IEEE/CAA Journal of Automatica Sinica*, 2023.

[26] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[29] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud:

A simple residual MLP framework. In *Int. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2022.

[30] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proc. of European Conference on Computer Vision*, 2022.

[31] Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J Kim. Self-positioning point-based transformer for point cloud understanding. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 21814–21823, 2023.

[32] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 77–85, 2017.

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, pages 5099–5108, 2017.

[34] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.

[35] Shi Qiu, Saeed Anwar, and Nick Barnes. Dense-resolution network for point cloud classification and segmentation. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, pages 3812–3821, 2021.

[36] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Trans. Multimedia (TMM)*, 24:1943–1955, 2022.

[37] Sucheng Ren, Xianhang Li, Haoqin Tu, Feng Wang, Fangxun Shu, Lei Zhang, Jieru Mei, Linjie Yang, Peng Wang, Heng Wang, et al. Autoregressive pretraining with mamba in vision. *arXiv preprint arXiv:2406.07537*, 2024.

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[40] Feng Wang, Jiahao Wang, Sucheng Ren, Guoyizhe Wei, Jieru Mei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Mamba-r: Vision mamba also needs registers. *arXiv preprint arXiv:2405.14858*, 2024.

[41] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2021.

[42] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019.

[44] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2P: tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.

[45] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[46] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

[47] Yaohua Zha, Jinpeng Wang, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Instance-aware dynamic prompt tuning for pre-trained point cloud models. In *Porc. of IEEE Intl. Conf. on Computer Vision*, 2023.

[48] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Proc. of Advances in Neural Information Processing Systems*, 2022.

[49] Tao Zhang, Xiangtai Li, Haobo Yuan, Shunping Ji, and Shuicheng Yan. Point could mamba: Point cloud learning via state space model. *arXiv preprint arXiv:2403.00762*, 2024.

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[51] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*.

[52] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.