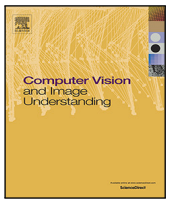
Computer Vision and Image Understanding 252 (2025) 104294

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Learning to mask and permute visual tokens for Vision Transformer pre-training

Lorenzo Baraldi [a,*,1], Roberto Amoroso [b,1], Marcella Cornia [b], Lorenzo Baraldi [b], Andrea Pilzer [c], Rita Cucchiara [b,d]

[a] University of Pisa, Pisa, Italy
[b] University of Modena and Reggio Emilia, Modena, Italy
[c] NVIDIA AI Technology Center, Italy
[d] Institute of Informatics and Telematics, National Research Council (CNR), Pisa, Italy

ARTICLE INFO

Communicated by Nicu Sebe

MSC:
68T45
68T30
68T01
68T05

ABSTRACT

The use of self-supervised pre-training has emerged as a promising approach to enhance the performance of many different visual tasks. In this context, recent approaches have employed the Masked Image Modeling paradigm, which pre-trains a backbone by reconstructing visual tokens associated with randomly masked image patches. This masking approach, however, introduces noise into the input data during pre-training, leading to discrepancies that can impair performance during the fine-tuning phase. Furthermore, input masking neglects the dependencies between corrupted patches, increasing the inconsistencies observed in downstream fine-tuning tasks. To overcome these issues, we propose a new self-supervised pre-training approach, named Masked and Permuted Vision Transformer (MaPeT), that employs autoregressive and permuted predictions to capture intra-patch dependencies. In addition, MaPeT employs auxiliary positional information to reduce the disparity between the pre-training and fine-tuning phases. In our experiments, we employ a fair setting to ensure reliable and meaningful comparisons and conduct investigations on multiple visual tokenizers, including our proposed *k*-CLIP which directly employs discretized CLIP features. Our results demonstrate that MaPeT achieves competitive performance on ImageNet, compared to baselines and competitors under the same model setting. We release an implementation of our code and models at https://github.com/aimagelab/MaPeT.

1. Introduction

Self-supervised pre-training models have achieved remarkable success in boosting the performance of Transformer-based architectures in Computer Vision. Inspired by the BERT model (Devlin et al., 2019), the Masked Image Modeling (MIM) pre-training objective (Bao et al., 2022) has become widely adopted in literature as a powerful self-supervision method for vision tasks. In particular, this pre-training objective involves masking random image patches and reconstructing the corrupted visual input. While several recent studies have refined the MIM approach (Chen et al., 2023; He et al., 2022), there has been little exploration of alternative pre-training objectives in the visual domain. In contrast, in the field of Natural Language Processing (NLP), several methods (Song et al., 2020; Yang et al., 2019) have surpassed the BERT pre-training objective with the introduction of different paradigms that aim to solve the drawbacks of previous methods.

Drawing inspiration from NLP, we investigate a permutation-based pre-training strategy, which we term Permuted Image Modeling (PIM). This approach autoregressively predicts permuted image patches maintaining contextual bi-directionality without corrupting any part of the input. Despite offering an improvement over the standard MIM-based objective, the autoregressive PIM technique reduces the amount of positional information available for each prediction. To tackle this issue, we propose a Masked and Permuted pre-training solution for Vision Transformers (MaPeT) which leverages auxiliary position information as input during pre-training, thus allowing the model to access the positional information of each image patch.

In addition to the pre-training objective, a crucial aspect of self-supervised vision pre-training is the design of visual targets, used as supervisory signal. While some works have employed low-level and hand-crafted visual features (Huang et al., 2023; Liu et al., 2022; Wei et al., 2022a), the dominant approach is to employ discrete visual tokens to reconstruct the corrupted input (Bao et al., 2022; Chen et al., 2023; El-Nouby et al., 2021; Peng et al., 2022). In this context,

* Corresponding author.
E-mail address: lorenzo.baraldi@phd.unipi.it (L. Baraldi).
[1] Both authors contributed equally to this research.

https://doi.org/10.1016/j.cviu.2025.104294
Received 22 January 2024; Received in revised form 13 January 2025; Accepted 13 January 2025
Available online 21 January 2025
1077-3142/© 2025 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

although BEiT (Bao et al., 2022) initially employed DALL-E (Ramesh et al., 2021) visual tokens, its performance has been surpassed by VQ-KD, proposed in BEiT v2 (Peng et al., 2022). In particular, VQ-KD employs an encoder–decoder architecture that reconstructs CLIP features and is directly trained on ImageNet-1k (Deng et al., 2009), requiring retraining to achieve satisfactory results on other datasets. In contrast to previous works, we propose k-CLIP, a novel discrete tokenizer for generating visual tokens that can directly employ CLIP features without requiring training an ad hoc discrete autoencoder.

To evaluate the effectiveness of the proposed pre-training model and tokenizer, we conduct experiments that provide a fair comparison between models, adhering to the same experimental settings. This approach allows to accurately measure the efficacy of each model under consistent conditions and effectively compare their relative strengths and weaknesses. Experimental results demonstrate that our MaPeT model achieves competitive performance and surpasses both mask- and permutation-based image pre-training. Additionally, we show that the visual tokens extracted by our proposed k-CLIP tokenizer exhibit richer semantic information than competitors, outperforming both DALL-E and VQ-KD visual tokens when employed directly for image classification.

2. Related work

Self-supervised learning. Several solutions have been introduced in the last years to effectively pre-train vision-based architectures via self-supervised learning, initially based on different pretext tasks (Doersch et al., 2015; Gidaris et al., 2018; Noroozi and Favaro, 2016; Wang and Gupta, 2015; Zhang et al., 2016) and then exploiting contrastive learning paradigms (Chen et al., 2020a; Chen and He, 2021; Grill et al., 2020; He et al., 2020; Oord et al., 2018). The advent of Vision Transformer models (ViT) (Dosovitskiy et al., 2021) has pushed towards the introduction of new increasingly sophisticated self-supervised pre-training strategies (Bao et al., 2022; Caron et al., 2021; Chen et al., 2020b, 2021). In this context, motivated by the great success in NLP, some attempts have been made to effectively adapt the Masked Language Modeling paradigm (Devlin et al., 2019; Liu et al., 2019) and its auto-regressive variant (Brown et al., 2020) in the Computer Vision domain, either by directly predicting pixels (Chen et al., 2020b), image patches (Dosovitskiy et al., 2021), or discrete visual tokens (Bao et al., 2022). In particular, the recently proposed BEiT approach (Bao et al., 2022) effectively performs pre-training via image patch masking and predicts the discretized labels (Ramesh et al., 2021) of masked patches. Since the introduction of BEiT, many subsequent methods based on similar pre-training strategies have been presented (Chen et al., 2023; Dong et al., 2023; El-Nouby et al., 2021; He et al., 2022; Tian et al., 2022, 2023; Wei et al., 2022a,b; Xie et al., 2022; Zhang et al., 2023). Some of these methods (Chen et al., 2023; El-Nouby et al., 2021; He et al., 2022) have introduced an encoder–decoder architecture to separate uncorrupted encoded information from masked tokens, which are employed directly as input to the decoder. Zhang et al. (2023) introduced HiViT, a simplified hierarchical Vision Transformer that combines the advantages of ViT and Swin Liu et al. (2022) models while being more efficient for Masked Image Modeling. Tian et al. (2023) proposed integrally pre-trained Transformer pyramid networks, which jointly optimize a HiViT-based network backbone and a feature pyramid neck to minimize the transfer gap between pre-training and downstream tasks. Additionally, Tian et al. (2022) investigated alternative learning objectives for token-based pre-training, providing insights into effective design principles beyond traditional masking approaches. Inspired by findings from NLP literature (Song et al., 2020; Yang et al., 2019), our proposed method aims to overcome the limitations of Masked Image Modeling in self-supervised pre-training.

Visual targets. In the domain of self-supervised learning, visual targets are a set of specific objectives that are used as the supervisory signal to pre-train visual models. Based on the type of signal, the targets can be categorized into low-level visual features, hand-crafted features, and visual tokens. Some recent studies (Fang et al., 2023; He et al., 2022; Huang et al., 2023; Liu et al., 2022; Xie et al., 2022) have utilized pixel information as the low-level supervisory signal for self-supervised pre-training. In contrast, Wei et al. (2022a) have used HOG hand-crafted features to reconstruct the masked visual input. Recently, visual targets based on CLIP (Radford et al., 2021) have been used with remarkable success, either by directly employing CLIP features (Hou et al., 2022; Wei et al., 2022b; Zhang et al., 2022) or training a discrete tokenizer (Peng et al., 2022) to reconstruct the semantic features encoded by CLIP. In our method, we propose a discrete visual tokenizer based on CLIP features which offers a novel approach to visual pre-training without the need for specific training over a particular dataset.

3. Preliminaries

In this section, we detail two pre-training strategies that are the starting point of our proposal, *i.e.*, Masked Image Modeling (MIM) and Permuted Image Modeling (PIM), and we introduce the terminology used in the rest of the paper.

Image patches. In this study, we adopt ViT (Dosovitskiy et al., 2021) as the backbone network for our architecture. ViT splits an image into a sequence of 2D image patches, which are linearly projected to the model embedding space and elaborated through multiple attention blocks. Given an input image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$, this is mapped into a sequence of N square patches $\{\boldsymbol{x}_i^p\}_{i=1}^N$, where $\boldsymbol{x}_i^p \in \mathbb{R}^{P\times P\times C}$ is the ith patch of the input image. Subsequently, a linear layer is applied to each flattened patch to project it to the input dimensionality of the model D, outputting the patch embeddings $\{\boldsymbol{x}_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^D$, which are then added to the learnable 1D positional embeddings $\boldsymbol{E}_{\text{pos}} \in \mathbb{R}^{N\times D}$.

The ViT (Dosovitskiy et al., 2021) encoder consists of L identical layers of Transformer blocks, where the output embeddings of the last layer represent the encoded representations of the N input image patches. In our experiments, we consider an image dimension of 224×224 with a patch dimension P of 16×16, constituting an input sequence of $14 \times 14 = 196$ patch embeddings.

Visual tokens. In self-supervised pre-training, learned supervisory signals are typically used to effectively pre-train the visual backbone. In this work, we employ visual tokens as supervisory signals during the pre-training phase. Specifically, we represent targets as a discrete token sequence, accomplished by utilizing a visual tokenizer $\mathcal{T}(\boldsymbol{x})$ on the input image $\boldsymbol{x}$. The visual tokenizer maps the image pixels onto a visual codebook (or vocabulary), generating a sequence of tokens $\boldsymbol{v} = [v_1, \dots, v_N] \in \mathcal{V}^{(H/P)\times(W/P)}$, where $\mathcal{V}$ represents the vocabulary containing discrete token indices. Our approach employs a 14×14 grid of visual tokens to represent each image, while the vocabulary size is set to $|\mathcal{V}| = 8192$.

Masked Image Modeling (MIM). Inspired by the Masked Language Modeling strategy utilized in BERT (Devlin et al., 2019), the MIM paradigm is a pre-training technique for vision tasks that aims to recover visual information from a corrupted input image. This is achieved by randomly masking a portion of the image patches and predicting the visual tokens related to the corrupted region of the input, as depicted in Fig. 1(a). Like Masked Language Modeling, MIM has some inherent disadvantages. Firstly, a mask token M is introduced during pre-training and never used during fine-tuning, leading to a discrepancy in the pre-training and fine-tuning phases. Secondly, given the masked tokens $\bar{\boldsymbol{x}}$ and the uncorrupted context $\tilde{\boldsymbol{x}}$, the probability $p(\bar{\boldsymbol{x}} \mid \tilde{\boldsymbol{x}})$ is typically factorized, assuming the independence of reconstructed patches. To address these issues, the NLP literature has investigated a permutation-based variant (Yang et al., 2019) that can reduce the disadvantages of standard Masked Language Modeling. These results motivate us to investigate the application of this strategy to vision tasks.

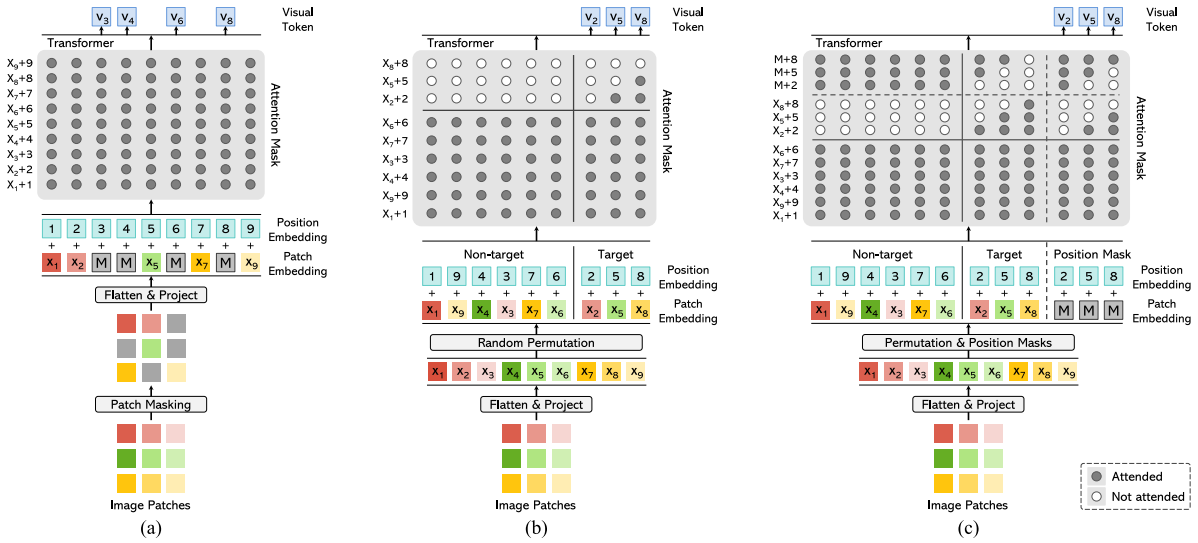


Fig. 1. (a) Masked Image Modeling (MIM). (b) Permuted Image Modeling (PIM). (c) Masked and Permuted pre-training for Vision Transformers (MaPeT). While MIM reconstructs visual tokens from randomly masked image patches, PIM autoregressively predicts tokens associated with permuted image patches. MaPeT uses PIM to capture intra-patch dependency and takes auxiliary position information as input to ensure that the model sees a full sequence of patches at each target position.

Permuted Image Modeling (PIM). The PIM self-supervised pre-training objective differs from MIM in two key components: the use of patch permutations and attention masking to capture bidirectional contexts. Specifically, PIM permutes patch embeddings, splits them into non-target and target patches, and predicts the visual tokens associated with the target patches using an auto-regressive approach. Attention masking is then applied to reduce the visibility of patches in the attention process, allowing a target patch to not access its contextual information (*i.e.*, its content) during prediction while remaining visible to patches that come after it in the permuted order. A visual representation of the process is shown in Fig. 1(b).

Formally, given an input image $\boldsymbol{x}$, we extract the patch embeddings $\{\boldsymbol{x}_i\}_{i=1}^N$, and tokenize it into N visual tokens $\{v_i\}_{i=1}^N$. We define $\mathcal{Z}_t$ as the set of all $N!$ possible permutations of the length-N index sequence $\{1, 2, \ldots, N\}$. Given a permutation $\boldsymbol{z}$, we use z_t and $\boldsymbol{z}_{<t}$ to denote the tth element and the first $t-1$ elements of $\boldsymbol{z}$, respectively. After applying $\boldsymbol{z}$, the permuted patch embeddings are fed into an L-layer ViT backbone to extract the final hidden representations. For each input embedding $\boldsymbol{x}_{z_t}$ at position z_t in the considered permutation $\boldsymbol{z}$, we use a softmax classifier to predict the corresponding visual token. The goal of PIM is to maximize the following log-likelihood objective:

$$\max_{\theta} \sum_{\boldsymbol{x} \in D} \mathbb{E}_{\boldsymbol{z} \in \mathcal{Z}_t} \left[\sum_{t=c+1}^{N} \log p_\theta(v_{z_t} | \boldsymbol{x}_{\boldsymbol{z}_{<t}}) \right], \tag{1}$$

where θ represents the model parameters, D denotes the training dataset, $\boldsymbol{x}_{\boldsymbol{z}_{<t}}$ are the only patch embeddings that are visible at position z_t, and c is a cutting point applied to split the permutation $\boldsymbol{z}$ in a subset of non-target patch embeddings $\boldsymbol{x}_{\boldsymbol{z}_{\leq c}}$ and target patch embeddings $\boldsymbol{x}_{\boldsymbol{z}_{>c}}$. The aim of c is to reduce the number of visual tokens to be predicted, thus mitigating optimization difficulties.

While MIM preserves full positional information of each image patch, during the prediction process, PIM can access the contextual and positional information only of the $t-1$ patches preceding z_t, as shown in Eq. (1). Given this lack of full positional information, PIM introduces an input discrepancy between pre-training and fine-tuning, underscoring the need for a pre-training technique that combines the advantages of both MIM and PIM while mitigating their respective limitations.

4. Proposed method

In this section, we present MaPeT, a novel pre-training paradigm that combines masked and permuted image modeling strategies. In addition, we also introduce the k-CLIP visual tokenizer, which exploits discretized CLIP features to produce visual tokens.

4.1. Masked and permuted pre-training

Drawing inspiration from NLP literature (Song et al., 2020; Yang et al., 2019), we propose a novel pre-training methodology called **Ma**sked and **Pe**rmuted Vision **T**ransformer (MaPeT) which builds on the strengths of both MIM and PIM to enhance performance on vision tasks. In particular, our approach overcomes the independence assumption of reconstructed patches, thus capturing intra-patch dependencies more effectively. Moreover, MaPeT incorporates auxiliary position embeddings during pre-training, enabling the model to access position information for all patches, thereby resolving the pre-training fine-tuning discrepancy introduced by PIM. Fig. 1(c) shows an overview of the MaPeT approach.

Given a permutation $\boldsymbol{z}$ and a cutting point c, MaPeT can predict the visual token associated with an input patch embedding $\boldsymbol{x}_{z_t}$ by leveraging the content and position of the preceding $\boldsymbol{x}_{\boldsymbol{z}_{<t}}$, as well as the position of the subsequent target embeddings $\boldsymbol{x}_{\boldsymbol{z}_{>t}}$. To this end, we introduce the concept of learnable masked token $M \in \mathbb{R}^D$, which is used to express the positional information of $\boldsymbol{x}_{\boldsymbol{z}_{>c}}$. By repeating $N-c$ times the token M, we obtain $\{M_i\}_{i=c+1}^N$ identical masked tokens, which are then summed to the positional embedding $\{E^i_{\text{pos}}\}_{i=z_{c+1}}^{z_N}$ of each target $\boldsymbol{x}_{\boldsymbol{z}_{>c}}$. These resulting position-aware masked tokens $\boldsymbol{M}_{\text{pos}} = \{M_{c+1} + E^{z_{c+1}}_{\text{pos}}, \ldots, M_N + E^{z_N}_{\text{pos}}\}$ are concatenated to the input patch embeddings $\boldsymbol{H}^0$ thus obtaining the augmented input $\boldsymbol{H}^0_M = [\boldsymbol{H}^0, \boldsymbol{M}_{\text{pos}}]$. Note that the positional embeddings are permuted according to the same permutation $\boldsymbol{z}$ applied to the patches. For convenience, we introduce $\boldsymbol{M}_{\boldsymbol{z}_{\geq t}}$ which represents the subset $\{M^i_{\text{pos}}\}_{i=\max(1,\, t-c)}^{N-c}$. Intuitively, when $t \leq c$, we have that $\boldsymbol{M}_{\boldsymbol{z}_{\geq t}}$ comprises all $\boldsymbol{M}_{\text{pos}}$. Conversely, when $t > c$, $\boldsymbol{M}_{\boldsymbol{z}_{\geq t}}$ only includes the position-aware masked tokens related to $\boldsymbol{x}_{\boldsymbol{z}_{\geq t}}$.

The training objective of our MaPeT model is to maximize the log-likelihood of predicting the visual token v_{z_t} associated with the patch embedding $\boldsymbol{x}_{z_t}$ given $\boldsymbol{x}_{\boldsymbol{z}_{<t}}$:

$$\max_{\theta} \sum_{\boldsymbol{x} \in D} \mathbb{E}_{\boldsymbol{z} \in \mathcal{Z}_t} \left[\sum_{t=c+1}^{N} \log p_\theta(v_{z_t} | \boldsymbol{x}_{\boldsymbol{z}_{<t}}, \boldsymbol{M}_{\boldsymbol{z}_{\geq t}}) \right]. \tag{2}$$

By doing so, MaPeT allows the patch embedding $\boldsymbol{x}_{z_t}$ to attend to contextual information of the patch embeddings $\boldsymbol{x}_{\boldsymbol{z}_{<t}}$ as well as the positional information of $\boldsymbol{x}_{\boldsymbol{z}_{>t}}$. This approach compensates for the position discrepancy of PIM and provides the model with information about the position of the target patches whose visual tokens are to be predicted. Fig. 2 shows an illustration of our MaPeT method.

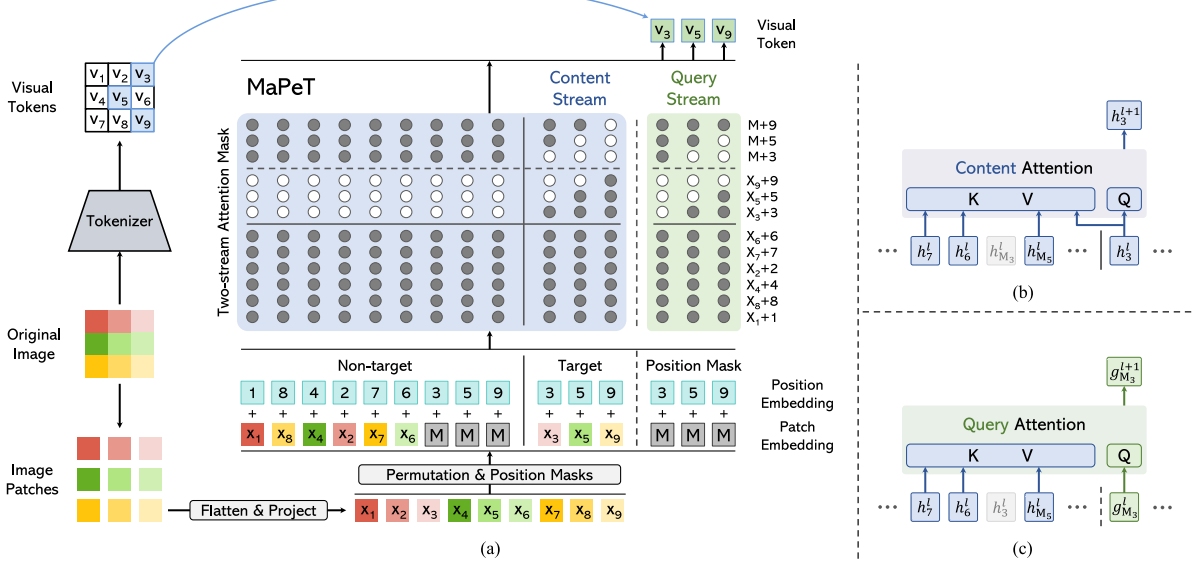


Fig. 2. Overview of our MaPeT pre-training (a), with content stream attention (b), which follows the standard self-attention mechanism, and query stream attention (c), which lacks information about the content of the patch embedding $\boldsymbol{x}_{z_t}$ whose visual token v_{z_t} is to be predicted. The blue and green masks in (a) are the content and query attention masks employed in the two-stream self-attention. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Two-stream self-attention pre-training. Since the target patch embeddings $\boldsymbol{x}_{z>c}$ follow the permuted order, the next predicted patch can occur in any position. As a consequence, masking the attention matrix of a ViT encoder, instead of corrupting the input like in MIM, makes the backbone architecture non-trivial. To implement PIM on a ViT backbone, we adopt the two-stream self-attention mechanism introduced by XLNet (Yang et al., 2019). Specifically, our model consists of two attention streams: a query stream and a content stream. The query stream $g_\theta(\boldsymbol{x}_{z_{<t}}, \boldsymbol{M}_{z_{\geq t}})$ accesses the content of the previous patches in the permuted sequence at a given position t. However, it does not access the content of $\boldsymbol{x}_{z_t}$, only viewing the position $\boldsymbol{M}_{z_{\geq t}}$ of the subsequent patches. In contrast, the content stream $h_\theta(\boldsymbol{x}_{z_{\leq t}}, \boldsymbol{M}_{z_{>t}})$ encodes the content of both the previous elements in the sequence and the element in position z_t, with positional information $\boldsymbol{M}_{z_{>t}}$ of the remaining patches. The input of the first query stream layer $\boldsymbol{g}^{(0)}$ consists of the masked elements $\boldsymbol{M}_{\text{pos}}$, which encode the position of the target patch embeddings $\boldsymbol{x}_{z_{>c}}$. Instead, the input of the first content stream layer $\boldsymbol{h}^{(0)}$ is the augmented input sequence $\boldsymbol{H}_M^0$. Formally, for each Transformer layer l with $l = 1, \ldots, L$, the attention mechanism of both content and query streams can be defined as:

$$\begin{aligned} h_{z_t}^{(l)} &\leftarrow \text{Attention}(\boldsymbol{Q} = h_{z_t}^{(l-1)}, \boldsymbol{KV} = (\boldsymbol{h}_{z_{\leq t}}^{(l-1)}, \boldsymbol{h}_{\boldsymbol{M}_{z_{>t}}}^{(l-1)}); \theta), \\ g_{z_t}^{(l)} &\leftarrow \text{Attention}(\boldsymbol{Q} = g_{z_t}^{(l-1)}, \boldsymbol{KV} = (\boldsymbol{h}_{z_{<t}}^{(l-1)}, \boldsymbol{h}_{\boldsymbol{M}_{z_{\geq t}}}^{(l-1)}); \theta), \end{aligned} \tag{3}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are respectively queries, keys, and values of the attention operator. Both query and content streams use separate attention masks to limit the visible contextual and positional information for each patch. During pre-training, the output of the query stream is used as the model output. At the fine-tuning step, the query stream is dropped, and only the content stream is used, returning to the standard ViT backbone. Fig. 2 shows the content stream (b), the query stream (c), and how they are integrated into our MaPeT architecture.

Attention masking. To limit the number of visible patches in the content and query attention operations, as described in Eq. (3), MaPeT leverages attention masking. In particular, the permutation z influences the creation of two distinct attention masks: one for the content attention stream and one for the query attention stream. The content mask guarantees that only patch embeddings $\boldsymbol{x}_{z_{\leq t}}$ and positional tokens $\boldsymbol{M}_{z_{>t}}$ are visible to the patch embedding $\boldsymbol{x}_{z_t}$ in the content stream. On the other hand, the query mask ensures that only patch embeddings $\boldsymbol{x}_{z_{<t}}$ and positional tokens $\boldsymbol{M}_{z_{\geq t}}$ are visible to $\boldsymbol{x}_{z_t}$ in the query stream.

4.2. *k*-CLIP: Discretized CLIP-based tokenizer

The role of visual tokenizers in pre-training pipelines is significant, as they provide crucial guidance for downstream fine-tuning outcomes. To possibly reduce the overhead of tokenizer retraining of previously proposed approaches (Peng et al., 2022), we explore the impact of directly utilizing discretized CLIP features. In particular, we propose a novel visual tokenizer, called *k*-CLIP, that employs discretized CLIP features as visual tokens. Our method does not rely on any pre-training or supervised data to create the visual tokens, thus enabling pre-training without access to large amounts of labeled data or a particular pre-training dataset. Furthermore, the use of CLIP features enables our method to capture high-level visual semantics that are more meaningful for downstream tasks, further improving the performance of the learned representations.

Specifically, we sample visual features from the ImageNet dataset (Deng et al., 2009) using the CLIP model (Radford et al., 2021) and cluster them using k-means to obtain $|\mathcal{V}| = 8192$ centroids. During pre-training, these centroids are indexed to retrieve the corresponding visual tokens for prediction. Formally, our visual tokenizer $\mathcal{T}(\boldsymbol{x})$ consists of a ResNet-based CLIP visual encoder and a k-means model. The visual encoder $f_v : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H_c \times W_c \times D_c}$ maps an input image $\boldsymbol{x}$ to a grid of visual features that correspond to the activations produced by the last convolutional layer of the CLIP backbone. The visual features are then reshaped to a $N_c \times D_c$ matrix, where $N_c = H_c \times W_c$. In our method, the CLIP visual encoder generates visual features of shape $N_c \times D_c = 196 \times 4096$, which are subsequently indexed by k-means and mapped to a sequence of 196 discrete visual tokens $\boldsymbol{v} = [v_1, \ldots, v_N] \in \mathcal{I}$. The set of visual tokens $\boldsymbol{v} = \{v_i\}_{i=1}^N$ is defined by the k-means centroid indexes $\mathcal{I} = \{1, \ldots, |\mathcal{V}|\}$. To mitigate the computational complexity, we randomly sample 2% of the approximately 250 million D_c-dimensional CLIP features extracted from ImageNet. The sampled feature collection is then used to fit the k-means clustering model.

5. Experiments

5.1. Experimental setup

Self-supervised pre-training literature has introduced several training and fine-tuning procedures that encompass different tokenizers,

hyperparameters, and visual backbones. To ensure a fair and unbiased comparison among these pre-training algorithms, we opt to evaluate our MaPeT method, as well as the related approaches in the literature, using identical experimental configurations. This strategy allows for the isolation of the algorithmic factor in the experiments and promotes unbiased comparisons of pre-training objectives. To assess the effectiveness of the pre-trained objective under consideration, we fine-tune our models on a downstream classification task. Additionally, following recent literature, we perform fine-tuning experiments for semantic segmentation, to assess the generalization of the proposed pre-training objective to dense predictions.

Pre-training setup. We first investigate the influence of the visual tokenizer and evaluate the proposed k-CLIP against VQ-KD (Peng et al., 2022). To minimize the computational effort, we pre-process the ImageNet-1k training dataset for both the tokenizers and store the visual tokens associated with each image. For this reason, our pre-training augmentation policy only includes color jittering to preserve image patch positions corresponding to the pre-extracted visual tokens.

We compare MaPeT against different pre-training objectives. In particular, we employ a ViT-based backbone that is pre-trained according to the pure MIM objective formulation. Note that this differs from the pre-training strategy proposed by the BEiT approach (Bao et al., 2022) as it lacks the block-wise masking algorithm, which progressively extracts multiple blocks of patches until 40% of the positions are masked. In our MIM-based pre-training, we replace the block-wise masking strategy with a random patch masking approach, thus keeping it similar to a BERT-like solution (Devlin et al., 2019) applied to Computer Vision tasks. Analogously, we also pre-train a ViT backbone through a double-stream architecture according to the PIM objective described in Section 3. Moreover, we consider the standard BEiT model (Bao et al., 2022) (*i.e.*, a MIM-based pre-training with block-wise masking) and CAE (Chen et al., 2023). The CAE method employs an encoder–decoder architecture where the encoder processes only visible image patches (50% of the entire image), while the remaining 50% is masked. A latent contextual regressor predicts the masked representation based on the encoder output, and a lightweight decoder processes the output of the regressor, which is then used to predict the visual token of the related masked patches.

Image classification setup. During the classification fine-tuning stage, the final hidden layer of the ViT-based backbone extracts features that are then combined via average pooling to generate a global image representation. This representation is subsequently fed into a softmax classifier. Following the linear probing experiment reported in Bao et al. (2022), we also train a linear classifier head over the output representation produced by the frozen pre-trained backbone.

We design three different model variants based on Vision Transformer (Dosovitskiy et al., 2021), *i.e.*, ViT-Tiny (ViT-T), ViT-Small (ViT-S), and ViT-Base (ViT-B). Our MaPeT model is trained by setting the cutting point c to 50, 50, and 60 respectively, using AdamW (Loshchilov and Hutter, 2019) as optimizer. We refer the reader to the supplementary material for detailed pre-training and fine-tuning hyperparameters.

Semantic segmentation setup. Semantic segmentation is a pixel-wise classification task that predicts semantic labels for each input image pixel. Our experimental framework follows the setting proposed in BEiT v2 (Peng et al., 2022) and utilizes the ADE20K dataset (Zhou et al., 2017, 2019) as a comprehensive benchmark comprising 25k images spanning 150 semantic categories. For the segmentation architecture, we employ UperNet (Xiao et al., 2018) task layer and fine-tune the model for 160k iterations with input images resized to a resolution of 512×512. The results are reported in terms of mean Intersection over Union (mIoU). To facilitate reproducibility, we include a detailed list of hyperparameters in the supplementary material.

Computational requirements. The pre-training experiments conducted with our MaPeT model involve the utilization of different GPU configurations. Specifically, the ViT-T, ViT-S, and ViT-B models require the deployment of 16, 32, and 64 GPUs, respectively. The pre-training process for each model takes approximately one day to complete. In contrast, during the fine-tuning phase, we employ 4 GPUs for the ViT-T and ViT-S models, and 16 GPUs for the ViT-B model. The fine-tuning duration for each model is 48 h, 36 h, and 12 h, respectively. For all experiments, we utilize an NVIDIA V100 GPU architecture with 16 GB of memory.

5.2. Experimental results

Pre-training objectives comparison. To validate the assumptions made on the pre-training objectives presented in Section 4, we conduct a comparison of MIM, PIM, and our proposed MaPeT, as shown in Table 1. This comparison evaluates the top-1 accuracy and linear probe accuracy of these approaches. The results indicate that MIM pre-training is less effective compared to PIM due to the input noise introduced by masked tokens and the independent reconstruction of patches. PIM outperforms MIM in most comparisons. For instance, when employing ViT-T with k-CLIP and ViT-S with VQ-KD, PIM achieves improvements of 0.4% and 0.1% in classification accuracy, respectively, over MIM. However, PIM demonstrates comparable performance to MIM when applied to the ViT-B backbone. Furthermore, our proposed MaPeT consistently outperforms PIM in nearly all cases. It exhibits accuracy gains of 0.2%, 0.3%, and 0.7% in top-1 accuracy, as well as 2.8%, 2.1%, and 0.5% in linear probe accuracy respectively on ViT-T and ViT-S when using the k-CLIP tokenizer, and on ViT-B when employing the VQ-KD tokenizer. These findings underscore the significance of addressing position inconsistency between pre-training and fine-tuning, particularly in the context of permutation-based image pre-training.

Comparison with state-of-the-art models. Table 2 presents a comprehensive analysis of the performance of MaPeT, BEiT, and CAE in terms of top-1 accuracy and linear probe accuracy across all ViT-based backbones. Firstly, our pre-trained MaPeT model showcases significant performance improvements across all Tiny, Small, and Base backbones compared to ViT-based models trained with random initialization, as evidenced in Table 2. Secondly, our results demonstrate that both BEiT and MaPeT outperform CAE. We hypothesize that the CAE encoder, which only observes 50% of the total sequence during pre-training, may suffer from position discrepancies between the pre-training and fine-tuning phases. Thirdly, it is noteworthy that BEiT improves overall results compared to MIM in Table 1. We attribute this improvement to the blockwise masking technique employed by BEiT. This technique follows the principle of image spatial locality, which posits that adjacent patches exhibit similarities in terms of visual information. By employing blockwise masking, the density of uncorrupted visual content is increased, while the noise introduced by masked tokens is concentrated in fewer locations instead of being sparsely distributed. Furthermore, MaPeT consistently outperforms all competitors across the three model variants when employing the k-CLIP tokenizer. Specifically, our model achieves top-1 accuracy margins of 0.7%, 0.2%, and 0.3% against BEiT on ViT-T, ViT-S, and ViT-B, respectively, while exhibiting margins of 1.8%, 0.6%, and 0.9% against CAE across the same three backbones. In contrast, MaPeT demonstrates comparable results to BEiT when employing the VQ-KD tokenizer on the ViT-S and ViT-B backbones.

We also report the comparison of our best variants and other state-of-the-art self-supervised pre-training models in Table 3. Note that the variability in the experimental settings and the different supervisory signals used in each approach may affect the fairness of the comparisons. Notably, although current literature has not extensively explored performance benchmarking on the ViT-S backbone, the Small version of our MaPeT model outperforms BEiT (Bao et al., 2022) and CAE (Chen et al., 2023) by 0.5% and 0.2% respectively, yielding the best performance for both k-CLIP and VQ-KD. In the case of the ViT-B backbone, MaPeT surpasses most of the considered approaches especially when using the VQ-KD tokenizer, except for BEiT v2 (Peng et al., 2022) which gets slightly better results. This performance gap can be explained by

Table 1
Fine-tuning results of different pre-training objectives in terms of top-1 and linear probe accuracy on ImageNet-1k. We report the results using both the VQ-KD and *k*-CLIP tokenizers.

Method	Tokenizer	*ViT-T*		*ViT-S*		*ViT-B*	
		Top-1 Acc. (%)	Linear probe Acc. (%)	Top-1 Acc. (%)	Linear probe Acc. (%)	Top-1 Acc. (%)	Linear probe Acc. (%)
MIM	VQ-KD	74.6	55.3	82.0	**69.9**	83.8	72.3
PIM	VQ-KD	**74.9**	56.4	82.1	68.7	83.7	73.3
MaPeT	VQ-KD	74.5	**59.6**	**82.2**	69.8	**84.4**	**73.8**
MIM	*k*-CLIP	74.9	60.4	82.0	71.1	83.3	72.3
PIM	*k*-CLIP	75.3	59.7	81.8	69.5	83.3	**73.5**
MaPeT	*k*-CLIP	**75.5**	**62.5**	**82.1**	**71.6**	**83.6**	**73.5**

Table 2
Fine-tuning results compared to other self-supervised pre-training approaches, in terms of top-1 and linear probe accuracy.

Method	Tokenizer	*ViT-T*		*ViT-S*		*ViT-B*	
		Top-1 Acc. (%)	Linear probe Acc. (%)	Top-1 Acc. (%)	Linear probe Acc. (%)	Top-1 Acc. (%)	Linear probe Acc. (%)
ViT (Dosovitskiy et al., 2021)	–	73.7	–	79.8	–	81.8	–
CAE (Chen et al., 2023)	VQ-KD	73.4	52.4	81.6	63.4	83.5	69.4
BEiT (Bao et al., 2022)	VQ-KD	**75.0**	**62.0**	**82.2**	**72.6**	**84.4**	**75.0**
MaPeT	VQ-KD	74.5	59.6	**82.2**	69.8	**84.4**	73.8
CAE (Chen et al., 2023)	*k*-CLIP	73.7	53.9	81.5	62.9	82.7	67.6
BEiT (Bao et al., 2022)	*k*-CLIP	74.8	61.5	81.9	71.1	83.3	73.3
MaPeT	*k*-CLIP	**75.5**	**62.5**	**82.1**	**71.6**	**83.6**	**73.5**

Table 3
Comparison with state-of-the-art self-supervised pre-training models in terms of top-1 accuracy on ImageNet-1k. Results of competitors are extracted from the original papers.

Method	# Epochs	*ViT-S*	*ViT-B*
		Top-1 Acc. (%)	Top-1 Acc. (%)
BEiT (Bao et al., 2022)	300	81.7	82.9
CAE (Chen et al., 2023)	300	82.0	83.6
SplitMask (El-Nouby et al., 2021)	300	–	83.6
MaskFeat (Wei et al., 2022a)	300	–	83.6
PeCo (Dong et al., 2023)	300	–	84.1
MVP (Wei et al., 2022b)	300	–	84.4
BEiT v2 (Peng et al., 2022)	300	–	**85.0**
MaPeT (k-CLIP)	300	82.1	83.6
MaPeT (VQ-KD)	300	**82.2**	84.4
BEiT (Bao et al., 2022)	800	–	83.2
CAE (Chen et al., 2023)	800	–	83.8
CAE (Chen et al., 2023)	1600	–	83.9
BEiT v2 (Peng et al., 2022)	1600	–	85.5

Table 4
Semantic segmentation results in terms of mIoU on ADE20K, comparing MaPeT with alternative pre-training objectives (top) and other self-supervised pre-training approaches (bottom).

Method	Tokenizer	mIoU (%)		
		ViT-T	*ViT-S*	*ViT-B*
MIM	VQ-KD	39.0	45.6	49.3
PIM	VQ-KD	**40.2**	46.1	49.0
MaPeT	VQ-KD	39.3	**47.0**	**50.4**
CAE (Chen et al., 2023)	VQ-KD	39.2	45.9	50.2
BEiT (Bao et al., 2022)	VQ-KD	**39.9**	46.7	50.3
MaPeT	VQ-KD	39.3	**47.0**	**50.4**

the lack of data augmentation in our model pre-training, which can significantly increase performance but at a higher computational cost.

The results in Table 3 are obtained after pre-training the models for 300 epochs. For completeness, in the lower part, we report the performance of other methods when pre-trained for a considerably large number of epochs (*i.e.*, 800 and 1600). While a direct comparison using different pre-training epochs may not be completely informative, MaPeT still proves to perform better than other methods pre-trained for a larger number of epochs such as BEiT and CAE.

Semantic segmentation results. To evaluate the effectiveness of MaPeT on dense prediction tasks, we present its performance on the semantic segmentation benchmark ADE20K using the UperNet framework with the VQ-KD tokenizer. Table 4 presents the mIoU scores achieved with various ViT backbones and pre-training strategies. Notably, MaPeT surpasses both MIM and PIM objectives, delivering significant improvements of +1.4% and +0.9% on ViT-S, and +1.1% and +1.4% on ViT-B, respectively. Compared to CAE (Chen et al., 2023) and BEiT (Bao et al., 2022), MaPeT achieves the highest mIoU across ViT-S and ViT-B backbones, while maintaining competitive performance on ViT-T. These results underscore the superiority of MaPeT over existing pre-training objectives and self-supervised approaches, particularly in addressing positional inconsistencies during pre-training and fine-tuning, which is critical for dense tasks like semantic segmentation. The robust generalization demonstrated by MaPeT highlights its potential as an effective pre-training strategy for segmentation tasks.

Cross-domain transfer learning. As an additional analysis, we examine the generalization capabilities of our proposed self-supervised pre-training technique, compared to BEiT (Bao et al., 2022) and CAE (Chen et al., 2023). All the considered architectures employ the VQ-KD visual tokenizer (Peng et al., 2022) and undergo pre-training on the ImageNet-1k dataset (Deng et al., 2009). Subsequently, they are fine-tuned on three distinct data domains, namely Stanford-Cars (Krause et al., 2013), Food-101 (Bossard et al., 2014), and FGVC-Aircraft (Maji et al., 2013). These datasets are chosen to represent diverse and real-world scenarios, ranging from object recognition in the automotive domain to food and aircraft classification. By employing a linear probe evaluation, we can quantitatively measure the ability of our pre-trained model to transfer knowledge and adapt to new tasks without fine-tuning.

The results, presented in Table 5, clearly demonstrate the superior performance of MaPeT across all considered datasets. Our model outperforms both BEiT and CAE, highlighting its robustness and efficacy in capturing meaningful visual representations. These findings not only

Table 5
Comparison of model performance on cross-domain transfer learning. We measure the linear probe accuracy of our proposed MaPeT model compared to BEiT and CAE on three distinct data domains: Stanford-Cars (Krause et al., 2013), Food-101 (Bossard et al., 2014), and FGVC-Aircraft (Maji et al., 2013).

Method	Tokenizer	Linear Probe Acc. (%)		
		Stanford-Cars	*Food-101*	*FGVC-Aircraft*
CAE (Chen et al., 2023)	VQ-KD	53.6	81.4	40.5
BEiT (Bao et al., 2022)	VQ-KD	64.5	**86.9**	44.9
MaPeT	VQ-KD	**68.5**	**86.9**	**46.8**

Table 6
Ablation study on the impact of the reconstruction factor on the top-1 fine-tuning accuracy of pre-trained ViT-B under the VQ-KD setting.

Cutting Point c	Reconstruction ratio	Tokenizer	Top-1 Accuracy (%)
30	85%	VQ-KD	84.2
40	80%	VQ-KD	84.3
50	75%	VQ-KD	84.3
60	70%	VQ-KD	**84.4**
98	50%	VQ-KD	84.1

underline the potential of MaPeT as a powerful pre-training technique but also emphasize its cross-domain transfer learning capabilities, which enable practical relevance in various real-world applications.

Reconstruction ratio analysis. Here we discuss the relationship between the reconstruction ratio employed in MaPeT and its impact on image classification results. The reconstruction ratio indicates the proportion of target patches in relation to the entire input sequence. An advantage of MaPeT is that it attends to the target patch $\boldsymbol{x}_{z_t} \in \boldsymbol{x}_{z_{>c}}$ during the prediction of $\boldsymbol{x}_{z_{t+1}}$, allowing for the cutting point c to be positioned at every value in the interval $c \in [1 \dots N-1]$. This unique feature of MaPeT enables the potential reconstruction of all patches in an incremental and randomized manner, from the first to the last patch. On the other hand, models like BEiT and CAE lack the ability to reconstruct all image patches because this would lead to a masking ratio that is too high, resulting in the loss of visual context necessary for the reconstruction of the masked elements.

Table 6 refers to the analysis of the reconstruction ratio on the ViT-B architecture. It can be noted that MaPeT exhibits a preference for a reconstruction ratio of approximately 70% of the complete sequence. Indeed, the initial target patches are compelled to establish correlations with a small portion of visible patches (namely, 30% of the entire sequence) randomly distributed throughout the image. The model can acquire long-range spatial dependencies that are contingent on the position of the target patch in the permuted order. As the final patches attend to the majority of the visual content, they are likely to concentrate more on neighboring image patches, consistent with the principle of spatial locality. A reduction in the reconstruction ratio leads to an increase in the number of visible patches attended by the target patches that occur early in the permuted order. This results in an increase in the likelihood of having neighboring visible patches, which, in turn, diminishes the learned long-range spatial dependencies. In contrast, an excessively high reconstruction ratio of 85% makes the pre-training objective of MaPeT excessively difficult, as early predictions are likely to be arbitrary, having access to only 15% of the overall visual information. Consequently, excessively high or low reconstruction ratios impair the performance.

5.3. Visual tokenizer analysis

In this section, we present a comprehensive qualitative and quantitative analysis of our proposed k-CLIP and compare its performance to the existing DALL-E (Ramesh et al., 2021) and VQ-KD (Peng et al., 2022) tokenizers.

Table 7
Accuracy of image classification when employing visual tokens as model input. Note that using CLIP features for zero-shot image classification lead to a top-1 accuracy of 73.6%.

Tokenizer	*MLP*		*ViT-T*	
	Top-1 Acc. (%)	Top-5 Acc. (%)	Top-1 Acc. (%)	Top-5 Acc. (%)
DALL-E	4.1	11.7	9.1	21.9
VQ-KD	68.1	89.8	72.3	92.2
k-CLIP	**72.8**	**93.2**	**76.2**	**94.9**

Impact of tokenizer on model performance. Analyzing the results in Table 1, it can be noted that k-CLIP outperforms VQ-KD across all models evaluated on ViT-T, with improvements in top-1 accuracy of 1.0%, 0.4%, and 0.3% observed on MaPeT, PIM, and MIM, respectively. However, the results differ when evaluated on ViT-S and ViT-B, where VQ-KD exhibits better classification performance. We argue that the simpler semantic features of k-CLIP, with their inherent correlation with the semantic density of the image, can serve as a more effective self-supervisory signal for smaller models. On the other hand, the VQ-KD codebook is trained specifically to reconstruct CLIP features, making it a more demanding pre-training signal and thus more suitable for larger models.

Image classification with discrete visual tokens. In this section, we analyze image classification performance when using discretized visual tokens directly as input to the model. We conduct this analysis using the previously employed tokenizers (*i.e.*, our k-CLIP and VQ-KD (Peng et al., 2022)) and also include the DALL-E visual tokenizer (Ramesh et al., 2021), which consists of a discrete variational autoencoder model. Specifically, we map the sequence of visual tokens to an embedding space of dimension D through learnable embeddings trained in conjunction with two different model backbones: a lightweight MLP classification head and a ViT-Tiny model. The MLP head comprises an embedding layer, a linear layer with output dimension $D = 192$, a ReLU activation, an average pooling operation, and an additional linear layer that projects the pooled features to the number of classes (*i.e.*, 1000).

The results shown in Table 7 indicate that the use of DALL-E visual tokens yields limited accuracy in this setting. This can be attributed to the relatively low amount of semantic information derived from the reconstruction task that DALL-E is trained on. In contrast, k-CLIP demonstrates significant performance superiority over alternative tokenizers when employed with both the MLP head and ViT-Tiny backbone. This observed trend can be attributed to the rich semantic information inherent in CLIP visual features, which is effectively preserved through the process of k-means discretization. Consequently, a strong correlation between visual tokens and image classes can be established, leading to favorable performance outcomes even when utilizing small models such as the MLP head.

Codebook visualization. Fig. 3 presents a collection of examples that showcase the semantic associations between image patches and visual tokens, extracted using our k-CLIP tokenizer. This visualization effectively demonstrates the efficacy of our tokenizer in accurately capturing and representing the semantic content of the images within the considered dataset and shows its ability to recognize and group image patches that share common visual features and semantic meaning. As it can be noticed, the visual tokens shown are congruent with specific semantic concepts, albeit resulting in distinct representations for similar visual elements such as `sunglasses` and `carnival mask`, `owl` and `peacock`, or `airplane` and `space shuttle`. Furthermore, it is worth noting that the visual tokens remain resilient to variations in color, style, rotation, and size. This is exemplified by the middle example, where the image depicts an owl print on a cup, yet it is still accurately identified using the `owl` visual token. These observations emphasize the ability of our k-CLIP tokenizer to effectively identify and classify complex visual concepts.

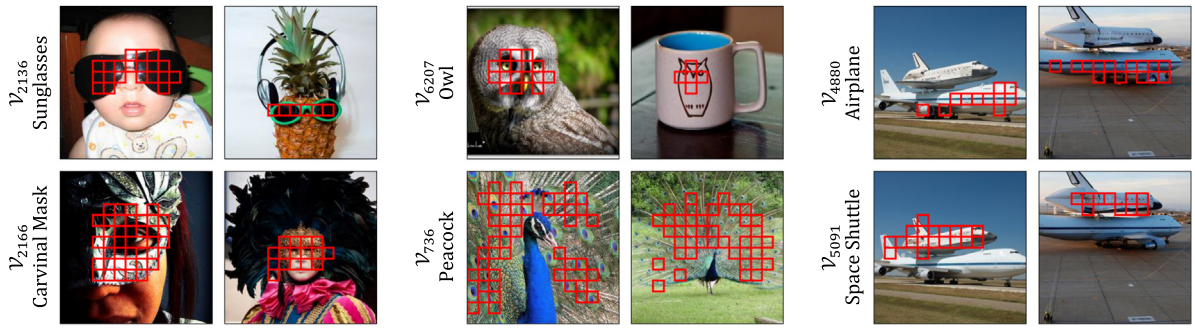


Fig. 3. Visualization of image patches corresponding to the discrete tokens of our k-CLIP codebook, where semantically similar patches consistently share the same discrete token.

6. Conclusion

This paper presents MaPeT, a novel self-supervised pre-training approach designed for vision tasks. Our model effectively tackles the limitations of standard Masked Image Modeling by employing a permutation-based objective to capture the interdependencies among predicted tokens and auxiliary position information to enable the model to access a full sequence of image patches, thus reducing the discrepancy between pre-training and fine-tuning. Moreover, we introduce the k-CLIP tokenizer that can densely capture the semantic information of the visual input by leveraging discretized CLIP features as visual tokens. Experimental results demonstrate that our approach achieves improved fine-tuning performance on image classification and semantic segmentation, outperforming direct competitors under the same experimental setting.

CRediT authorship contribution statement

Lorenzo Baraldi: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Roberto Amoroso:** Writing – original draft, Validation, Software, Methodology, Conceptualization. **Marcella Cornia:** Writing – review & editing, Supervision, Conceptualization. **Lorenzo Baraldi:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Andrea Pilzer:** Writing – review & editing, Validation, Software. **Rita Cucchiara:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources. This work has partially been supported by the PNRR-M4C2 project “FAIR - Future Artificial Intelligence Research” and the Horizon Europe project “ELSA - European Lighthouse on Secure and Safe AI” (GA 101070617), both funded by the European Union.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cviu.2025.104294.

Data availability

Data will be made available on request.

References

Bao, H., Dong, L., Piao, S., Wei, F., 2022. BEiT: BERT pre-training of image transformers. In: Proceedings of the International Conference on Learning Representations.

Bossard, L., Guillaumin, M., Van Gool, L., 2014. Food-101–mining discriminative components with random forests. In: Proceedings of the European Conference on Computer Vision.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. In: Advances in Neural Information Processing Systems.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., Wang, J., 2023. Context autoencoder for self-supervised representation learning. Int. J. Comput. Vis..

Chen, X., He, K., 2021. Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020a. A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I., 2020b. Generative pretraining from pixels. In: Proceedings of the International Conference on Machine Learning.

Chen, X., Xie, S., He, K., 2021. An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics.

Doersch, C., Gupta, A., Efros, A.A., 2015. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N., 2023. PeCo: Perceptual codebook for BERT pre-training of vision transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations.

El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., Grave, E., 2021. Are large-scale datasets necessary for self-supervised pre-training?. arXiv preprint arXiv:2112.10740.

Fang, Y., Dong, L., Bao, H., Wang, X., Wei, F., 2023. Corrupted image modeling for self-supervised visual pre-training. In: Proceedings of the International Conference on Learning Representations.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. In: Proceedings of the International Conference on Learning Representations.

Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., 2020. Bootstrap your own latent-a new approach to self-supervised learning. In: Advances in Neural Information Processing Systems.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2022. Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Hou, Z., Sun, F., Chen, Y.K., Xie, Y., Kung, S.Y., 2022. MILAN: Masked image pretraining on language assisted representation. arXiv preprint arXiv:2208.06049.

Huang, Z., Jin, X., Lu, C., Hou, Q., Cheng, M.M., Fu, D., Shen, X., Feng, J., 2023. Contrastive masked autoencoders are stronger vision learners. IEEE Trans. Pattern Anal. Mach. Intell..

Krause, J., Stark, M., Deng, J., Fei-Fei, L., 2013. 3D object representations for fine-grained categorization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al., 2022. Swin transformer V2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. In: Proceedings of the International Conference on Learning Representations.

Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A., 2013. Fine-Grained Visual Classification of Aircraft. Technical Report.

Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of the European Conference on Computer Vision.

Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F., 2022. Beit v2: Masked image modeling with vector-quantized visual tokenizers. arXiv preprint arXiv:2208.06366.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I., 2021. Zero-shot text-to-image generation. In: Proceedings of the International Conference on Machine Learning.

Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y., 2020. Mpnet: Masked and permuted pre-training for language understanding. In: Advances in Neural Information Processing Systems.

Tian, Y., Xie, L., Fang, J., Shi, M., Peng, J., Zhang, X., Jiao, J., Tian, Q., Ye, Q., 2022. Beyond masking: Demystifying token-based pre-training for vision transformers. arXiv preprint arXiv:2203.14313.

Tian, Y., Xie, L., Wang, Z., Wei, L., Zhang, X., Jiao, J., Wang, Y., Tian, Q., Ye, Q., 2023. Integrally pre-trained transformer pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Wang, X., Gupta, A., 2015. Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.

Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C., 2022a. Masked feature prediction for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Wei, L., Xie, L., Zhou, W., Li, H., Tian, Q., 2022b. MVP: Multimodality-guided visual pre-training. In: Proceedings of the European Conference on Computer Vision.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H., 2022. SimMIM: a simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. XLNet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems.

Zhang, X., Chen, J., Yuan, J., Chen, Q., Wang, J., Wang, X., Han, S., Chen, X., Pi, J., Yao, K., Han, J., Ding, E., Wang, J., 2022. CAE v2: Context autoencoder with CLIP target. arXiv preprint arXiv:2211.09799.

Zhang, R., Isola, P., Efros, A.A., 2016. Colorful image colorization. In: Proceedings of the European Conference on Computer Vision.

Zhang, X., Tian, Y., Huang, W., Ye, Q., Dai, Q., Xie, L., Tian, Q., 2023. Hivit: Hierarchical vision transformer meets masked image modeling. In: Proceedings of the International Conference on Learning Representations.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2017. Scene parsing through ADE20K dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic understanding of scenes through the ADE20K dataset. Int. J. Comput. Vis. 127, 302–321.