# THE EXTRA TOKENS MATTER: DISENTANGLED REPRESENTATION LEARNING WITH VISION TRANSFORMERS

## **Anonymous authors**

000

001

002

004

006

008 009 010

011 012

013

014

015

016

017

018

019

021

023

024

026027028

029

031

033

034

037

038

040

041

042

043 044

046

047

048

049

051

052

Paper under double-blind review

#### **ABSTRACT**

Inspired by Darcet et al. (2024) where extra tokens (or registers) are introduced to offset the artifacts in feature maps due to high-norm tokens, this paper presses further and asks a more challenging question: Can we find a suitable regularization term such that the extra tokens can evolve into disentangled representations, capable of attending to finer details of objects (e.g., parts)? We propose XTRA, an intuitive yet powerful framework that augments Vision Transformers with dedicated "factor tokens" and enforces disentanglement via a novel Minimum Volume Constraint (MVC). A multi-stage aggregation process further confines these factor tokens into semantically pure components, especially when the amount of hyperparameters is large. On ImageNet-1K, XTRA boosts KNN accuracy by 5.8% and linear-probe accuracy by 2.3% over leading self-supervised learning (SSL) baselines, outperforming even models trained on larger datasets.

#### 1 Introduction

It is widely believed that the power of deep learning lies in its ability to learn meaningful representations (Bengio et al., 2013), which remains a central challenge. In recent years, self-supervised learning (SSL) (He et al., 2020; 2021; Bao et al., 2022; Zhou et al., 2022) has sparked growing interest in representation learning and achieved remarkable performance in various downstream tasks (Caron et al., 2021a; Touvron et al., 2021a;b; Wang et al., 2021). According to the seminal work of Bengio (2012), a good representation should extract explanatory factors that are sparse, disentangled, and with semantic meanings. In particular, it has been shown through DINO (Caron et al., 2021a; Oquab et al., 2024) that features from self-supervised Vision Transformer (ViT) contain explicit information about the semantic segmentation of an image. More recently, Darcet et al. (2024) demonstrated that by appending additional tokens (or registers) to the input sequence, a correlation can be established between high-norm tokens and artifacts of the feature maps. While making breakthrough discoveries of the semantic meaning of extra tokens, these works have not considered the disentanglement aspect of representation learning. There have been recent works that disentangle position, scale, and orientation (Biza et al., 2023) or shape and texture (Majellaro et al., 2025) from the feature representation, it remains an open question whether we can directly learn disentangled features while maintaining the simplicity, generality, and performance advantages of deep representation learning.

Direct learning of disentangled features requires explicit constraint(s) to regularize the learning trajectory. Here, we draw inspiration from the field of remote sensing and spectral unmixing for potential choices of constraints. In remote sensing, due to the large footprint, where, for example, a Landsat 8 pixel typically covers a ground area of  $30 \times 30 \text{ m}^2$  (Roy et al., 2014), the measured spectrum at a single pixel is usually a so-called "mixture" and can be modeled as a linear combination of several ground cover spectra, referred to as the "pure spectra" or signature. Spectral unmixing is to extract the pure substance from the mixture to achieve subpixel-level quantification. One of the most effective regularization terms in extracting the pure spectra is to minimize the "volume" of the space spanned by the pure spectra (Craig, 1994; Miao & Qi, 2007).

The mixture model and the unmixing process resemble the generation of disentangled attention maps (i.e., pure spectra) pertaining to *consistent parts* across multiple objects in the scene (i.e., mixture),

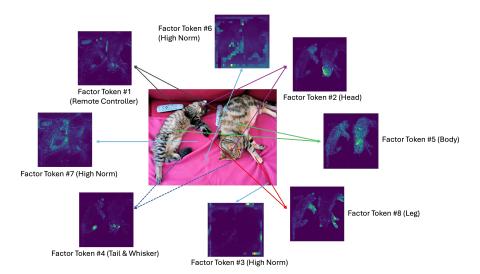


Figure 1: XTRA enables disentangled attention maps pertaining to consistent parts across multiple objects in the scene.

as shown in Fig. 1. Disentangled representation learning also is also analogous of teh well-known cocktail party problem, where the "listening attention" should be focused on a single talker among a mixture of conversations and background noise.

Built on top of (Darcet et al., 2024) where non-regularized extra tokens are added to the input, in this paper, we consider the patch tokens as "mixtures" of semantic contents in the scene. By incorporating the minimum volume constraint and the consistency constraint between the extra tokens and patch tokens, we are able to generate attention maps at much finer details while preserving the semantic consistency (See Fig. 1). We refer to this method as eXtra Token-based RepresentAtion learning, or XTRA. Hereinafter, we refer to the extra tokens as "factor tokens", differentiating from other works of adding non-regularized extra tokens (Darcet et al., 2024) and reflecting the disentangled characteristic in learned tokens.

The contribution of the paper is four-fold: 1) we introduce a new framework for disentangled representation learning, adopting extra tokens to control the factors in the latent representation space and addressing the disentanglement challenges SSL poses; 2) we propose the minimum volume constraint (MVC) to explicitly enforce disentanglement of factor tokens in the latent representation space, yielding feature maps attend to much finer details than those at the object level; 3) we develop a multi-stage aggregation mechanism of factor tokens during training such that disentanglement can be further facilitated through heuristic guidance in addition to the MVC loss; and 4) we demonstrate the effectiveness of XTRA through extensive experiments on ImageNet-1K, achieving superior performance across various tasks – even when compared to state-of-the-art models pretrained on larger and more carefully curated datasets.

### 2 Related Work

Object-centric Representation Learning. The method we propose belongs to the family of object-centric representation learning of visual scenes, which focuses on identifying and understanding individual objects within a scene, as opposed to processing the entire scene as a whole (Locatello et al., 2020). Object-centric learning models assume that the image is composed of K distinct objects, including the background, and the model is trained in an unsupervised manner to identify these K objects, thereby providing a more detailed and nuanced understanding of the scene. Earlier work like Eslami et al. (2016) adopted a recurrent neural network (RNN) to perform probabilistic inference that attends to and processes one object in a scene at a time. Greff et al. (2019); Engelcke et al. (2019) achieved meaningful decomposition of non-trivial scenes with a variable number of objects using, e.g., the CLEVR dataset (Johnson et al., 2017). More recently, Slot Attention (Locatello et al.,

2020) and variants (Kipf et al., 2022; Singh et al., 2022; Zhang et al., 2022; Jia et al., 2023; Biza et al., 2023; Kori et al., 2024) introduced a non-probabilistic iterative mechanism that is competitive with its predecessors while being faster to train and more memory efficient.

**Disentanglement in Representation Learning.** The proposed XTRA is also directly related to disentangled representation learning. Within this area, probabilistic models such as Greff et al. (2020); Burgess et al. (2019) can obtain a degree of disentanglement due to their VAE backbone. Other works, such as Anciukevicius et al. (2020), pursued explicit disentanglement of position and depth, also within a probabilistic framework. Mansouri et al. (2023), instead, exploited weak supervision from sparse perturbations and causal representation learning to disentangle object properties. In a non-probabilistic setting, Singh et al. (2022) learned disentangled representations in a non-explicit manner, while Biza et al. (2023) introduced invariance to changes in position, scale, and rotation with the use of slot-centric reference frames, allowing for the explicit disentanglement of those three factors.

Extra Tokens in Transformers. BERT (Devlin et al., 2019) is among the first that uses special tokens (e.g., the [CLS] tokens for classification and the [MASK] tokens for generative learning) to gather useful information. Beyond the [CLS] tokens, Visual Prompt Tuning (VPT) and its variants (Jia et al., 2022; Yoo et al., 2023; Wang et al., 2024b) introduced a small set of learnable tokens injected at every transformer layer, enabling efficient downstream adaptation without modifying the pretrained weights. Tokens have also been studied in relation to uninformativeness. For example, A-ViT (Yin et al., 2022) learns a per-token halting probability to discard low-value tokens; Attentive Tokens (Long et al., 2022) select or merge tokens based on learned importance scores; and more recently, Darcet et al. (2024) introduced extra tokens were used to offset artifact behaviors to yield a smoother attention map.

Unlike explicitly disentangling shape and texture as in object-centric learning, this paper focuses on data-driven feature disentanglement via introducing regularized extra tokens for self-distillation. To the best of our knowledge, no research has addressed the explicit disentanglement in self-supervised learning, which is the primary focus of our work.

# 3 METHOD

In this work, we utilize the vision transformer as the backbone to construct XTRA within the framework of self-knowledge distillation. In the following, we first explain the rationale behind the minimum volume constraint (MVC) and how volume is calculated based on the factor tokens. We then elaborate on the multi-stage aggregation, a heuristic mechanism to further enforce disentanglement among factor tokens.

# 3.1 LEARNING FACTOR TOKENS WITH THE MINIMUM VOLUME CONSTRAINT (MVC)

The goal of factor tokens is not merely to store high-level information, such as high norm or noise, as in Darcet et al. (2024), but to ensure that the patch tokens (i.e., the mixture) can be adequately represented by the factor tokens (e.g., pure spectra) in the representation space. Specifically, given the set of N patch tokens,  $\{\mathbf{p}_i\}_{i=1}^N$ , and the set of M extra factor tokens,  $\{\mathbf{f}_i\}_{i=1}^M$ , we seek a disentangled representation of  $\mathbf{p}_i$  such that

$$\mathbf{p}_i = F \cdot \mathbf{w}_i, \ F = [\mathbf{f}_1, \cdots, \mathbf{f}_M] \tag{1}$$

where a linear mixing model has been assumed as in most spectral unmixing formulations (Miao & Qi, 2007).  $\mathbf{w}_i$  is the learnable weight vector indicating the contribution of each factor token in making up the patch token.

We thus define the latent loss on the relationship between the patch tokens and the factor tokens as:

$$\mathcal{L}_{\text{latent}} = \lambda_{\text{factor}} \cdot \mathcal{L}_{\text{factor}} + \lambda_{\text{volume}} \cdot J(F)$$
 (2)

$$\mathcal{L}_{\text{factor}} = \frac{1}{2} \log \left( \sum_{i=1}^{N} \| \mathbf{p}_i - F \cdot \mathbf{w}_i \|^2 \right)$$
 (3)

$$J(F) = \|F^T F - I\|_F^2 \tag{4}$$

where J(F) is the volume penalty term on the space spanned by the factor vectors in F, and  $\lambda_{\text{volume}}$  is a hyperparameter controlling the strength of this penalty. The two loss terms in Eq. 2 has an intuitive geometrical interpretation, as shown in Fig. 2a, where the circles indicate patch tokens in the latent space and the vertices of the triangle (or simplex) indicate the factor tokens. As such, the first term,  $\mathcal{L}_{\text{factor}}$ , serves as the external force to drive the search to move outward, so that the generated simplex contains all patch tokens with relatively small errors, and the second term, J(F), serves as the internal force, which constrains the simplex volume to be small. A solution is found when these two forces balance each other.

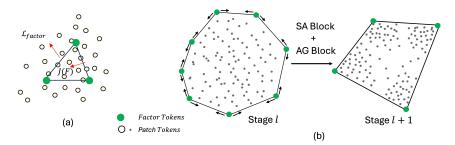


Figure 2: (a) A geometric illustration of the two loss terms within the latent loss (Eq. 2) where the minimum volume constraint, J(F), serves as the internal force pointing inward and the patch reconstruction constraint,  $\mathcal{L}_{factor}$ , serves as the external force pointing outward. (b) Illustration of how the factor tokens and patch tokens evolve across two stages of aggregations.

In addition to controlling the volume of the simplex, J(F) also encourages the vectors in F to be orthogonal. In Eq. 4,  $F^TF$  is the Gram matrix of F, and I is the identity matrix. The Frobenius norm of  $F^TF - I$  quantifies the deviation from orthogonality, and minimizing this term encourages the vectors in F to be mutually orthogonal. The orthogonality reduces redundancy by ensuring that each vector in F carries unique information, thus enhancing separability; in addition, it guarantees that the factor set F spans a unique subspace, avoiding overfitting and promoting better generalization.

To simplify computation, the volume of the space spanned by the factor tokens  $\{\mathbf{f}_i\}_{i=1}^M$  can be computed through Singular Value Decomposition (SVD). Given the SVD of  $F = U\Sigma V^T$ , where U and V are orthogonal matrices, and  $\Sigma$  is a diagonal matrix of singular values  $\sigma_i$ , the volume of the space spanned by F is then given by  $J(F) = \sum_{i=1}^r \sigma_i^2$  with r being the rank of matrix F.

One of the key advantages of the proposed framework is that it promotes interpretability by ensuring that the factor tokens  $\{\mathbf{f}_i\}_{i=1}^M$  is decoupled and structured. This structure provides a way to interpret how the factor vectors contribute to the final representation of the patch tokens, making the model more explainable.

### 3.2 Multi-Stage Aggregation of Factor Tokens

Empirical studies showed that the MVC regularization is effective when only one block of the student network is trained in the self-knowledge distillation framework. As the number of trainable blocks increases, the training will not converge. See the first data point in Fig. 6b with 12 trainable blocks. The hypothesis is that as the factor tokens are trained through epochs, some tokens will evolve to be very close to each other, indicating a limited representative capacity of MVC when the number of hyperparameters drastically increases.

To achieve the representation disentanglement in self-knowledge distillation via extra tokens, we design a dual-stream framework, including a self-attention stream [Fig. 3(top)] of the teacher network and a multi-stage aggregation stream [Fig. 3(bottom)] of the student network. The multi-stage aggregation stream is further illustrated in Fig. 4, where each stage incorporates an aggregation block at its end to merge correlated factor tokens into a new factor token. Fig. 2b illustrates how the factor tokens and patch tokens evolve across two stages of aggregations.

Formally, suppose there are L aggregation stages indexed by l, a set of learnable aggregation tokens  $\{\mathbf{g}_i\}_{i=1}^{M_l}$ , and the initial factor tokens  $\{\mathbf{f}_i\}_{i=1}^{M_0}$ , where  $M_0$  is the initial number of factor tokens. We

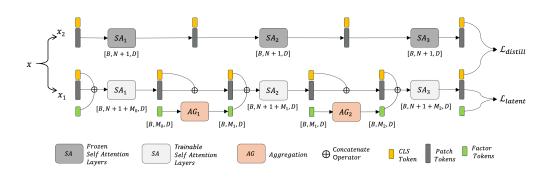


Figure 3: Illustration of XTRA built upon the dual-stream self-knowledge distillation network. Top: teacher network. Bottom: The multi-stage aggregation student network.

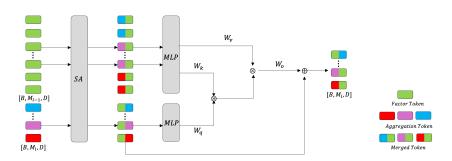


Figure 4: Illustration of a 2-stage aggregation of factor tokens

simplify  $\left\{\mathbf{f}_i^l\right\}_{i=1}^{M_{l-1}}$  to  $\left\{\mathbf{f}_i^l\right\}$  and similarly  $\left\{\mathbf{g}_i^l\right\}_{i=1}^{M_l}$  to  $\left\{\mathbf{g}_i^l\right\}$ . Starting with l=1, for each aggregation stage, the number of <code>[CLS]</code> token and patch tokens are fixed at 1 and N, respectively. We first concatenate factor tokens  $\left\{\mathbf{f}_i^l\right\}$ , the <code>[CLS]</code> token,  $\left\{\mathbf{c}^l\right\}$ , and the patch tokens,  $\left\{\mathbf{p}_i^l\right\}$ , together and then input them into the self-attention layers, each of which performs information propagation between them,

$$\left\{\hat{\mathbf{c}}^{l}\right\}, \left\{\hat{\mathbf{f}}_{i}^{l}\right\}, \left\{\hat{\mathbf{p}}_{i}^{l}\right\} = \text{Self-Attentions}\left(\left[\left\{\mathbf{c}^{l}\right\}; \left\{\mathbf{f}_{i}^{l}\right\}; \left\{\mathbf{p}_{i}^{l}\right\}\right]\right)$$
 (5)

where [; ] denotes the concatenation operator. Then we aggregate the updated  $M_{l-1}$  factor tokens  $\left\{\hat{\mathbf{f}}_i^l\right\}$  into  $M_l$  new factor tokens  $\left\{\mathbf{f}_i^{l+1}\right\}$  via an Aggregation Block as

$$\left\{\mathbf{f}_{i}^{l+1}\right\} = \operatorname{Aggregation}\left(\left\{\mathbf{g}_{i}^{l}\right\}, \left\{\hat{\mathbf{f}}_{i}^{l}\right\}\right).$$
 (6)

In each aggregation stage  $M_l < M_{l-1}$ , i.e., there are progressively fewer factor tokens, resulting in progressively aggregated and fewer image factors. See details in Appendix A. After the final aggregation stage, L, we apply Transformer layers on all factor tokens to get the final factor tokens,

$$\left\{\hat{\mathbf{f}}_{i}^{L+1}\right\} = \text{Self-Attentions}\left(\left\{\mathbf{f}_{i}^{L+1}\right\}\right)$$
 (7)

### 3.3 Knowledge Distillation from the Foundation Model

As discussed in Sec. 3.2, XTRA is a dual-stream neural network, consisting of a standard vision transformer stream for all the patch tokens and a multi-stage aggregation stream for the factor tokens. Specifically, rather than concatenating only one trainable [CLS] token with the patch tokens, the M trainable factor tokens are also concatenated with the patch tokens. These trainable tokens are then fed to the designed network that outputs the learned [CLS] token, patch tokens, and M factor tokens, after L aggregation stages.

Following the standard self-knowledge distillation framework, given the image x, first, random data augmentations are used to generate distinct views. For clarity, we consider two views, i.e.,  $x_1$  and  $x_2$ , whose representations are extracted by the teacher network T and the student network S, respectively. Then, the <code>[CLS]</code> tokens are further processed using projection heads. In this paper, we select the asymmetric contrastive loss to measure the similarity between the <code>[CLS]</code> tokens output from the teacher and the student networks, representing the distillation loss,  $\mathcal{L}_{\text{distill}}$ . See details in Appendix B.

# 3.4 TOTAL LOSS FUNCTION

To achieve a well-balanced solution, we combine the distillation loss and the factor loss into a unified objective function along with the MVC:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{distill}} \cdot \mathcal{L}_{\text{distill}} + \lambda_{\text{factor}} \cdot \mathcal{L}_{\text{factor}} + \lambda_{\text{volume}} \cdot J(F)$$
 (8)

where  $\lambda_{\text{distill}}$ ,  $\lambda_{\text{factor}}$ , and  $\lambda_{\text{volume}}$  are hyperparameter that control the trade-off among the different loss terms. We minimize the total loss function  $\mathcal{L}_{\text{total}}$  that results in a model that effectively represents the patch tokens through a set of factor tokens that is both structurally simple and robust, with mutually independent vectors that span a well-defined subspace. Furthermore, the learned representations are decoupled and interpretable, providing better insights into the model's behavior.

#### 3.5 IMPLEMENTATION DETAILS

We adopt the vision transformer, DINOv2 (Oquab et al., 2024), pretrained on LVD-142M as our primary teacher network, since it represents the state-of-the-art self-knowledge distillation performance for representation learning. Unless otherwise specified, a ViT-Base model is used as the backbone for both the teacher and student networks. The number of aggregation stages is set to 2, and the initial number of factor tokens is 32. The aggregation follows  $32 \to 16 \to 8$ , and the final number of factor tokens is 8. Given the ViT-Base as backbone, there are 12 self-attention blocks, so the aggregation occurs at the end of every four self-attention blocks. The weights for the different loss terms are preset at  $[\lambda_{\text{distill}}, \lambda_{\text{factor}}, \lambda_{\text{volume}}] = [1, 0.45, 0.05]$  according to extensive empirical studies. We pretrain the models on the ImageNet1K without labels. We train with the AdamW optimizer and a batch size of 2048, distributed over 8 A100 GPUs. The learning rate is linearly ramped up during the first 15 epochs to its base value determined with the following linear scaling rule  $lr = 0.0005 \times batchsize \div 256$ . After this warmup, we decay the learning rate with a cosine schedule. The weight decay also follows a cosine schedule from 0.04 to 0.4. The temperature  $\tau$  is set to 0.1 while we use a linear warmup for  $\tau$  from 0.04 to 0.07 during the first 30 epochs. For consistency, we use the same augmentations as in DINO (Caron et al., 2021b).

# 4 EXPERIMENTS AND RESULTS

#### 4.1 Comparison with State-of-the-Art

KNN & Linear Probing. Following standard self-supervised evaluation protocols, we evaluate XTRA's representations on ImageNet-1K using KNN and linear-probe accuracy, as shown in Table 1. We observe that XTRA outperforms all prior ImageNet-1K pre-training methods by 2.1% in KNN and 1.5% in linear probing. XTRA also outperforms every DINO models, including DINO v2 and its variant with register. It demonstrates that XTRA, using a foundation model as a teacher, can generate better representations, with a lightweight trainable student network and extra token regularization.

We further investigate the performance of learned representation "without" the strong pre-trained teacher network. For fair comparison, we use the same backbone, ViT-Base, and pre-train both DINO v2 and XTRA on the same dataset, ImageNet-1K. The results are reported in Table 2. We observe that even without a pre-trained foundation model as teacher, XTRA maintains its superior performance.

**Downstream Tasks.** We evaluate XTRA's generality by fine-tuning on three downstream benchmarks—ImageNet-1K classification, ADE20K semantic segmentation, and COCO2017 ob-

Table 1: Evaluation of representation from pre-trained model in KNN and Linear Probing (%).

	Backbone	Dataset	Epochs	KNN	Linear
MoCo-v3	ViT-B	IN-1K	1200	51.2	76.3
MAE	ViT-B	IN-1K	800	54.75	71.8
BEiT	ViT-B	IN-1K	800	49.06	56.7
iBOT	ViT-B	IN-1K	1600	72.9	82.3
DINO v1	ViT-B	IN-1K	300	76.1	78.2
OpenCLIP	ViT-G	IN-1K	800	75.2	78.2
OpenCLIP + Reg	ViT-G	IN-1K	800	75.8	78.1
DINO v2	ViT-G	LVD-142M	-	82.1	84.5
DINO v2 + Reg	ViT-G	LVD-142M	-	82.0	83.6
XTRA	ViT-B	IN-1K	300	84.2	86.0

Table 2: Evaluation of representation without pre-trained teacher network in KNN and Linear Probing (%).

	Backbone	Dataset	Epochs	KNN	Linear
DINO v2	ViT-Base	IN-1K	300	76.9	80.1
DINO v2 + Register	ViT-Base	IN-1K	300	77.3	82.1
XTRA	ViT-Base	IN-1K	300	81.9	83.8

ject detection—each for 100 epochs. See Appendix C for detailed hyperparameter setup. Table 3 summarizes Top-1 accuracy for classification, mIoU for segmentation, and  $AP_{\rm box}$  for detection. XTRA surpasses state-of-the-art methods on ImageNet classification, ADE20K segmentation, and COCO2017 object detection, demonstrating the robustness of its learned representations across diverse tasks.

# 4.2 REPRESENTATION DISENTANGLEMENT

Since disentangled representation learning by explicit regularization is the main claim of XTRA, in this set of experiments, we evaluate the degree of disentanglement of the learned representation. Given no ground truth, we follow Wang et al. (2024a) and adopt an unsupervised disentanglement metric SEPIN@k (Do & Tran, 2021). SEPIN@k measures how each token  $\{\mathbf{p}_i\}$  is disentangled from others  $\{\mathbf{p}_{\neq i}\}$  by computing their conditional mutual information with the top k features.

As shown in Table 4, the representation from XTRA exhibits significantly better disentanglement than DINO v2 and its variant in all top-k dimensions. Since the learned features also contain noisy

Table 3: Evaluation of representation from pre-trained model with different downstream tasks (%).

	Classification (Top-1)	Segmentation (mIoU)	Detection (AP box)
Backbone	ImageNet 1K	ADE20K	COCO2017
MoCo-v3	83.1	47.3	47.9
MAE	83.6	48.1	50.3
BEiT	83.2	47.1	49.8
iBOT	84.0	50.0	48.2
DINO v1	82.8	51.3	46.8
DINO v2	85.8	54.4	51.2
DINO v2 + Register	85.6	54.2	50.5
XTRA	85.9	55.1	52.1

components, the all-dimension (k=768) results are close among all methods, with XTRA still maintaining a slight advantage.

In Fig. 5, we further show the representation SEPIN@k score at the different aggregation stages, where the first two stages are the representation after aggregation, and the last stage is the output representation. For comparison purpose, we also use DINO v2 and DINO v2-Reg, both of which have four self-attention blocks at each stage. The results again demonstrate that the factor token aggregation helps drastically enhance the disentanglement of representation.

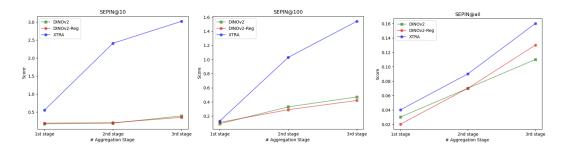


Figure 5: Evaluation of the disentanglement score at different aggregation stages in XTRA

Table 4: Representation disentanglement score with SEPIN@k on ImageNet-1k, where k denotes the top-k dimensions (higher is better).

	SEPIN@1	SEPIN@10	SEPIN@100	SEPIN@all
DINO v2	0.47	0.39	0.28	0.11
DINO v2 + Register	0.42	0.35	0.25	0.13
XTRA	3.95	3.02	1.54	0.16

#### 4.3 ABLATION STUDY

Effectiveness of Each Module XTRA integrates three components—knowledge distillation from a frozen LVD-142M DINO v2 teacher ( $\mathcal{L}_{distill}$ ), factor representation capacity ( $\mathcal{L}_{factor}$ ), and a volume penalty, (J(F)), on the space spanned by the factor tokens , as shown in Eq. 8. Table 5 reports results from an incremental ablation study. We observe that freezing the teacher reduces KNN accuracy but improves linear-probe performance, indicating more generative representations; adding factor reasoning alone reduces both metrics by over 3.5%; but incorporating the volume penalty improves performance by 6.8% (KNN) and 7.9% (linear probe). We view the combination of factor reasoning and volume regularization as a Min-Max operator in latent space that robustly pushes representations toward the desired properties.

**Effect of Model Complexity** We further study how the number of trainable self-attention blocks in the student affects performance. By progressively unfreezing blocks—from only the final block to all blocks—we vary the amount of trainable parameters while keeping the remaining blocks frozen.

Table 5: Ablation study of the effect of each module in XTRA (%)

Frozen Teacher	Factors Rep	Volume Penalty	kNN	Linear Probing
0	0	0	76.1	78.2
$\checkmark$	0	0	$76.0 \downarrow 0.1$	$79.2 \uparrow 1.0$
$\checkmark$	$\checkmark$	0	$72.4 \downarrow 3.6$	$74.8 \downarrow 4.4$
$\checkmark$	0	$\checkmark$	$79.2 \uparrow 6.8$	$82.9 \uparrow 7.9$
$\checkmark$	$\checkmark$	$\checkmark$	<b>84.2</b> † 4.0	<b>86.0</b> ↑ 3.1

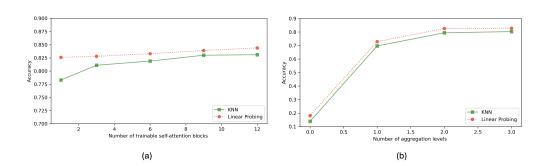


Figure 6: The effect of model complexity and aggregation. KNN & linear probing performance on (a) student networks of different numbers of trainable blocks and (b) different aggregation levels.

As shown in Fig. 6(a), KNN accuracy improves from 78.3% to 84.2%, and linear-probe accuracy from 82.6% to 86.0%, as more blocks become trainable. Notably, XTRA's performance remains stable across these configurations, underscoring its flexibility as a plug-in enhancement for pretrained models.

Effect of Multi-Stage Aggregation In addition to the above two studies, we also investigate the effects of the number of aggregation stages. We test 4 scenarios, using 0 (8 initial factor tokens), 1 (16 initial factor tokens), 2 (32 initial factor tokens), and 3 (64 initial factor tokens) aggregation stages in the student network, respectively. The results are shown in Fig. 6b. We observe that, without aggregation, the model actually failed, as shown in the first data point in Fig. 6b (KNN 13.9%, linear probing 18.1%). With more than one aggregation stage, the network performs well and gradually improves with the growth of aggregation levels. Comparing performance between the two and three aggregation stages, we see that the improvement is limited (KNN increases 0.9%, linear probing increases 0.2%), so more aggregation may not bring improvement. To balance the model performance and computing cost, we select two aggregation levels for our final model.

# 5 CONCLUSION & LIMITATION

Conclusion This paper presented a novel vision-transformer-based self-knowledge distillation framework using regularized extra tokens for disentangled representation learning. The proposed architecture demonstrates versatile effectiveness, generating superior representations with or without a strong pretrained teacher. The key innovation lies in utilizing regularized extra tokens as interpretable factors through multiple aggregatable stages and structured reasoning between factor and patch tokens, decomposing visual information into semantically meaningful components. Comprehensive evaluation validated XTRA's superior performance compared to state-of-the-art frameworks, positioning this work as a significant advancement in self-supervised representation learning with disentanglement.

Limitation and Future Work We list the limitations of XTRA as follows. (1) Although XTRA can produce disentangled factor tokens that attend to finer details, it cannot automatically map the semantics with the token without human inspection. Controllable generative learning will be our future work. (2) We have not looked into the dynamics of the evolution of factor tokens. Current work focuses on the structure of the latent space in the final stage of the vision transformer, rather than the entire network. The dynamics of the factor tokens throughout the entire network can reveal interesting behaviors of the learning mechanism, further providing a potential way to control the learning target. (3) The aggregation mechanism lacks the flexibility of data-driven clustering with a variable number of resulting tokens.

# REFERENCES

- Titas Anciukevicius, Christoph H. Lampert, and Paul Henderson. Object-centric image generation with factored depths, locations, and appearances, 2020. URL https://arxiv.org/abs/2004.00642.
- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. URL https://arxiv.org/abs/2106.08254.
- Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.
  - Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
  - Ondrej Biza, Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin F. Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames, 2023. URL https://arxiv.org/abs/2302.04973.
  - Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019. URL https://arxiv.org/abs/1901.11390.
  - Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021a.
  - Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021b. URL https://arxiv.org/abs/2104.14294.
  - M. D. Craig. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3):542–552, 1994.
  - Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. URL https://arxiv.org/abs/2309.16588.
  - Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.
  - Kien Do and Truyen Tran. Theory and evaluation metrics for learning disentangled representations, 2021. URL https://arxiv.org/abs/1908.09961.
  - Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv* preprint *arXiv*:1907.13052, 2019.
  - SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.
- Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pp. 2424–2433. PMLR, 2019.
  - Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference, 2020. URL https://arxiv.org/abs/1903.00450.

- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. URL https://arxiv.org/abs/1911.05722.
  - Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.
    - Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization, 2023. URL https://arxiv.org/abs/2210.08990.
    - Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning, 2022. URL https://arxiv.org/abs/2203.12119.
    - Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1988–1997, 2017. doi: 10.1109/CVPR.2017.215.
    - Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video, 2022. URL https://arxiv.org/abs/2111.12594.
    - Avinash Kori, Francesco Locatello, Fabio De Sousa Ribeiro, Francesca Toni, and Ben Glocker. Grounded object centric learning, 2024. URL https://arxiv.org/abs/2307.09437.
    - Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020. URL https://arxiv.org/abs/2006.15055.
    - Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers, 2022. URL https://arxiv.org/abs/2211.11315.
    - Riccardo Majellaro, Jonathan Collu, Aske Plaat, and Thomas M. Moerland. Explicitly disentangled representations in object-centric learning, 2025. URL https://arxiv.org/abs/2401.10148.
    - Amin Mansouri, Jason Hartford, Yan Zhang, and Yoshua Bengio. Object-centric architectures enable efficient causal representation learning, 2023. URL https://arxiv.org/abs/2310.19054.
    - Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, 2007.
    - Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.
    - D.P. Roy, M.A. Wulder, T.R. Loveland, Woodcock C.E., R.G. Allen, M.C. Anderson, D. Helder, J.R. Irons, D.M. Johnson, R. Kennedy, T.A. Scambos, C.B. Schaaf, J.R. Schott, Y. Sheng, E.F. Vermote, A.S. Belward, R. Bindschadler, W.B. Cohen, F. Gao, J.D. Hipple, P. Hostert, J. Huntington, C.O. Justice, A. Kilic, V. Kovalskyy, Z.P. Lee, L. Lymburner, J.G. Masek, J. McCorkel, Y. Shuai, R. Trezza, J. Vogelmann, R.H. Wynne, and Z. Zhu. Landsat-8: Science and product vision for terrestrial global change research. *Remote Sensing of Environment*, 145:154–172, 2014. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2014.02.001. URL https://www.sciencedirect.com/science/article/pii/S003442571400042X.

- Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos, 2022. URL https://arxiv.org/abs/2205.14065.
  - Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In International conference on machine learning, pp. 10347–10357. PMLR, 2021a.
  - Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 32–42, 2021b.
  - Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.
  - Yifei Wang, Qi Zhang, Yaoyu Guo, and Yisen Wang. Non-negative contrastive learning, 2024a. URL https://arxiv.org/abs/2403.12459.
  - Yuzhu Wang, Lechao Cheng, Chaowei Fang, Dingwen Zhang, Manni Duan, and Meng Wang. Revisiting the power of prompt for visual tuning, 2024b. URL https://arxiv.org/abs/2402.02382.
  - Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. Adavit: Adaptive tokens for efficient vision transformer, 2022. URL https://arxiv.org/abs/2112.07658.
  - Seungryong Yoo, Eunji Kim, Dahuin Jung, Jungbeom Lee, and Sungroh Yoon. Improving visual prompt tuning for self-supervised vision transformers, 2023. URL https://arxiv.org/abs/2306.05067.
  - Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Is an object-centric video representation beneficial for transfer?, 2022. URL https://arxiv.org/abs/2207.10075.
  - Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022. URL https://arxiv.org/abs/2111.07832.

# A MULTI-STAGE AGGREGATION OF FACTOR TOKENS

As a supplement explanation to Sec . 3.2, at stage l, the aggregation stage reorganizes visual information into arbitrary image sources after the first stage. It merges all the factor tokens assigned to the same aggregation token into a new factor based on similarity in the embedding space. Formally, we compute the similarity matrix  $\mathbf{A}^l$  between the aggregation tokens  $\left\{\hat{\mathbf{g}}_i^l\right\}$  and factor tokens  $\left\{\hat{\mathbf{f}}_i^l\right\}$  via a Gumbel-Softmax operation computed over the group tokens as

$$\mathbf{A}_{i,j}^{l} = \frac{\exp\left(W_{q}\mathbf{g}_{i}^{l} \cdot W_{k}\hat{\mathbf{f}}_{j}^{l} + \gamma_{i}\right)}{\sum_{k=1}^{M_{l}} \exp\left(W_{q}\mathbf{g}_{k}^{l} \cdot W_{k}\hat{\mathbf{f}}_{j}^{l} + \gamma_{k}\right)}$$
(9)

where  $W_q$  and  $W_k$  are the weights of the learned linear projections for the aggregation and factor tokens, respectively, and  $\{\gamma_i\}$  are i.i.d random samples drawn from the Gumbel (0, 1) distribution. We compute the aggregation to assign a factor token by taking the one-hot operation of its argmax over all the aggregations. Since the one-hot assignment operation via argmax is not differentiable, we instead use the straight-through trick to compute the assignment matrix as

$$\hat{\mathbf{A}}^{l} = \text{one} - \text{hot} \left( \mathbf{A}_{\text{argmax}}^{l} \right) + \mathbf{A}^{l} - \text{sg} \left( \mathbf{A}^{l} \right)$$
 (10)

where sg is the stop gradient operator, with straight-through trick,  $\hat{\mathbf{A}}^l$  has the one-hot value of assignment to a single aggregation, but its gradient is equal to the gradient of  $\mathbf{A}^l$ , which makes the aggregation block differentiable and trainable from end to end. We call this one-hot assignment strategy a hard assignment. After assigning the factor tokens to the different learned aggregations, we merge the embedding of all the tokens belonging to the same aggregation to form a new factor token  $\mathbf{f}_i^{l+1}$ . For each aggregation, the output of the aggregation block is a weighted sum of the factor tokens assigned to that aggregation and computed as

$$\mathbf{f}_{i}^{l+1} = \mathbf{g}_{i}^{l} + W_{o} \frac{\sum_{j=1}^{M_{l-1}} \hat{\mathbf{A}}_{i,j}^{l} W_{v} \hat{\mathbf{f}}_{j}^{l}}{\sum_{j=1}^{M_{l-1}} \hat{\mathbf{A}}_{i,j}^{l}}$$
(11)

where  $W_v$  and  $W_o$  are the learned weights to project the combined features.

#### B KNOWLEDGE DISTILLATION FROM THE FOUNDATION MODEL

As a supplement explanation to Sec . 3.3, given the image x, first, random data augmentations are used to generate distinct views. For clarity, we consider two views, i.e.,  $x_1$  and  $x_2$ , whose representations are extracted by the teacher network T and the student network S, respectively. So,  $\left[\hat{c},\hat{f},\hat{p}\right]=T(x_1)$  and  $\left[\tilde{c},\tilde{f},\tilde{p}\right]=S(x_2)$ , respectively. Then, the <code>[CLS]</code> tokens are further processed using projection heads. i.e.,  $\hat{h}^c=p\hat{roj}\left(\hat{c}\right)$  and  $\tilde{h}^c=p\hat{roj}\left(\tilde{c}\right)$ . Subsequently, they are substituted in different loss functions, such as InfoNCE loss, clustering loss, and similarity loss. Some methods also attach a predictor head to the student network, and we omit it for conciseness.

In this paper, we select the asymmetric contrastive loss to measure the similarity between the <code>[CLS]</code> tokens output from the teacher and the student networks, representing the distillation loss,  $\mathcal{L}_{distill}$ , and is defined as

$$\mathcal{L}_{dist} = \mathcal{L}_{\hat{h}^c \to \tilde{h}^c} = \mathcal{L}_{\hat{h}^c \to \tilde{h}^c} + \mathcal{L}_{\tilde{h}^c \to \hat{h}^c} \tag{12}$$

which is composed of two asymmetric contrastive losses defined as

$$\mathcal{L}_{\hat{h}^c \to \tilde{h}^c} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(\hat{h}_i^c \cdot \tilde{h}_i^c / \tau\right)}{\sum_{j=1}^{B} \exp\left(\hat{h}_i^c \cdot \tilde{h}_i^c / \tau\right)}$$
(13)

Table 6: Hyperparameters for pre-training on ImageNet-1K using ViT-Base model.

Hyperparameters	Base Size
SA layers in SA Block	4
Aggregation Levels	2
Initial Number of Factor Tokens	32
Final Number of Factor Tokens	8
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Layer scale	0.1
Patch size	$16 \times 16$
Relative positional embeddings	✓
Shared relative positional embeddings	0
Training epochs	300
Batch size	2048
Adam $\epsilon$	1e-8
Adam $\beta$	(0.9, 0.999)
Peak learning rate	1.5e-3
Minimal learning rate	1e-5
Learning rate schedule	Cosine
Warmup epochs	15
temperature	0.1
Stoch. depth	0.1
Gradient clipping	3.0
Dropout	0
Stoch. depth	0
Weight decay	0.05
Data Augment	RandomResizeAndCrop
Input resolution	$224 \times 224$
Color jitter	0.4
CLS Loss	InfoNCE
Latent Loss	Smooth L1
Factor Constraint	MVC

$$\mathcal{L}_{\tilde{h}^c \to \hat{h}^c} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp\left(\tilde{h}_i^c \cdot \hat{h}_i^c / \tau\right)}{\sum_{j=1}^{B} \exp\left(\tilde{h}_i^c \cdot \hat{h}_i^c / \tau\right)}$$
(14)

Here, B is the batch size. The CLS token is often adopted to encode the global context, which could be a good representation for global semantic information. However, it may be less representative of factors controlling different aspects of an image, such as foreground/background, object position/rotation, object properties, etc. To enhance the representation in the capability of explainability and disentanglement, we introduce the factor tokens, which can be complementary to enhance representations. Specifically, we design the properties of the latent space spanned by the factor tokens and look into the relationship between factor tokens and patches.

# C MORE TRAINING DETAILS

#### C.1 Pretraining and Evaluation

In our pretraining, we adopt the vision transformer, DINO v2 Oquab et al. (2024), pretrained on LVD-142M as our primary teacher network, since it represents the state-of-the-art self-knowledge distillation performance for representation learning. Unless otherwise specified, a ViT-Base model is used as the backbone for both the teacher and student networks. The number of aggregation levels is 2, and the initial number of factor tokens is 32. The aggregation follows  $32 \rightarrow 16 \rightarrow 8$ , and

Table 7: Hyperparameters for linear-probing on ImageNet-1K.

Hyperparameters	ViT-B/16
Peak learning rate	5e-4
Fine-tuning epochs	100
Warmup epochs	20
Layer-wise learning rate decay	0.65
Batch size	1024
Adam $\epsilon$	1e-8
Adam $\beta$	(0.9, 0.999)
Minimal learning rate	1e-6
Learning rate schedule	Cosine
Stoch. depth	0.1
Repeated Aug	✓
Weight decay	0.05
Dropout	0
Gradient clipping	0
Input resolution	$224 \times 224$

Table 8: Hyperparameters for fine-tuning on ImageNet-1K.

Hyperparameters	ViT-B/16
Peak learning rate	5e-4
Fine-tuning epochs	100
Epochs	100
Warmup epochs	10
Layer-wise learning rate decay	0.65
Batch size	1024
Adam $\epsilon$	1e-8
Adam $\beta$	(0.9, 0.999)
Minimal learning rate	4e-4
Learning rate schedule	Cosine
Stoch. depth	0.1
Repeated Aug	0
Weight decay	0.05
Label smoothing $\varepsilon$	0.1
Dropout	0
Gradient clipping	0
Erasing prob.	0.25
Input resolution	$224 \times 224$
Rand Augment	9/0.5
Mixup prob.	0.6
Cutmix prob.	0.75

the final factor tokens are 8. Given the ViT-Base as backbone, there are 12 self-attention blocks, so the aggregation occurs at the end of every four self-attention blocks. The weights for the different loss terms are preset at  $[\lambda_{\text{distill}}, \lambda_{\text{factor}}, \lambda_{\text{volume}}] = [1, 0.45, 0.05]$  according to extensive empirical studies. We pretrain the models on the ImageNet 1K without labels. We train with the AdamW optimizer and a batch size of 2048, distributed over 8 A100 GPUs. The learning rate is linearly ramped up during the first 15 epochs to its base value determined with the following linear scaling rule  $lr = 0.0005 \times batchsize \div 256$ . After this warmup, we decay the learning rate with a cosine schedule. The weight decay also follows a cosine schedule from 0.04 to 0.4. The temperature  $\tau$  is set to 0.1 while we use a linear warm-up for  $\tau$  from 0.04 to 0.07 during the first 15 epochs. For consistency, we use the same augmentations as in DINO v1 Caron et al. (2021b).

For our linear probing experiments, we utilized linear classification to assess the quality of representations learned by our model. Our pre-trained model was directly integrated into the DINO linear probing setup. We adopted the ViT-base architecture with a patch size 16 and an input resolution of  $224 \times 224$  for the linear probing implementation. Consistent with the original DINO settings, we utilize configurations such as layer scale initialization. Following standard linear evaluation protocols, a supervised linear classifier was appended to the frozen backbone. The training was conducted using the AdamW optimizer with a learning rate of  $4 \times 10^{-3}$ , and the models were trained for 100 epochs on the ImageNet-1K dataset. Linear probing hyperparameter setups are shown in Table 7.

# C.2 DOWNSTREAM TASKS

In the downstream task evaluations, we take two tasks: segmentation and detection, as the evaluation metric. For each task, a specific task head is integrated with the pretrained XTRA model, the UpNet for segmentation, and Mask R-CNN for detection. For the segmentation task, with ADK20K, the The hyperparameter setups are shown in Table 9 and Table 10.

# MORE EXPERIMENTS RESULTS

#### D.1 **ABLATION STUDY**

Besides the experiments in Sec. 4.3, we conducted more research and showed them in this part. First, in Figure 6(a), we further study how the number of trainable self-attention blocks in the student af-

Table 9: Hyperparameters for fine-tuning on ADE20K.

Hyperparameters	ViT-B
Segmentation Head	UpNet
Pretrained Model Finetune	✓
Relative positional embeddings	✓
Shared relative positional embeddings	0
Epochs	100
Peak learning rate	0.5e-4
Fine-tuning steps	160K
Batch size	16
Adam $\epsilon$	1e-8
Adam $\beta$	(0.9, 0.999)
Layer-wise learning rate decay	0.75
Minimal learning rate	0
Learning rate schedule	Linear
Warmup steps	1500
Dropout	0
Stoch. depth	0.1
Weight decay	0.05
Input resolution	$512 \times 512$

Table 10: Hyperparameters for fine-tuning on COCO2017.

Hyperparameters	ViT-B
Detection Head	Mask R-CNN
Pretrained Model Finetune	✓
Relative positional embeddings	✓
Shared relative positional embeddings	0
Epochs	100
Peak learning rate	0.5e-4
Fine-tuning steps	160K
Batch size	16
Adam $\epsilon$	1e-8
Adam $\beta$	(0.9, 0.999)
Layer-wise learning rate decay	0.75
Minimal learning rate	0
Learning rate schedule	Linear
Warmup steps	1500
Dropout	0
Stoch. depth	0.1
Weight decay	0.05
Input resolution	$640 \times 640$

Table 11: The effect of different volumes method (%)

Metric	Backbone	Gram	SVD
KNN	ViT-Base	13.9	83.1
Linear Probing	ViT-Base	18.1	84.4

fects performance. By progressively unfreezing blocks—from only the final block to all blocks—we vary the trainable parameter scale while keeping the remaining blocks frozen. As shown in Figure 6(a), KNN accuracy improves from 78.3% to 83.1%, and linear-probe accuracy from 82.6% to 84.4%, as more blocks become trainable. Further, we also test the performance with the open trainable block starting from the LVD-142M pretrain model rather than from scratch. The results are shown in Figure 7, KNN accuracy improves from 79.5% to 83.5%, and linear-probe accuracy from 83.6% to 85.5%. Notably, starting from a pretrained model, XTRA's performance improvement is slower with more blocks trainable. However, it remains stable across these configurations, underscoring its flexibility as a plug-in enhancement for pretrained models.

Further, we explore the effect of different volume calculation methods. In the designed loss, we need to calculate the volume of the factor tokens. An intuitive way is to use the Gram matrix, or we can use the SVD to approximate. We explore the effect of different choices and show them in the Tab. 11. From the results, we can find that the Gram matrix failed, but SVD works well. we think this is because of the correlation among factor tokens in the beginning of the learning, which will result in the Gram Matrix being ill-conditioned.

#### D.2 FACTOR TOKEN VISUALIZATION

In this paper, we consider the initial extra tokens as "mixtures" of semantic contents in the scene. By incorporating the minimum volume constraint and the consistency constraint between the extra tokens and patch tokens, we can generate remarkable attention maps with much finer details while preserving semantic consistency. The main paper showed that the different factor tokens can present different parts of the object, including high norms Darcet et al. (2024). In this part, we present more results. See Fig. 8 and Fig. 9. In Fig. 8, we show the results on different animals, and the results show the capability of the factor tokens to capture the semantic part. We also show the factor token with a high norm. In Fig. 9, we show the results on other objects, such as no animals, which is easier to capture the whole object. We think it may be because these objects are difficult to disentangle into part-wise properties. This is a potential direction for our future exploration.

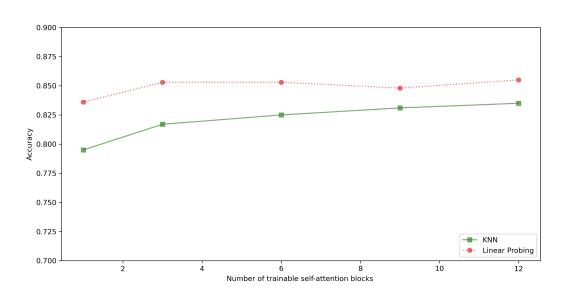


Figure 7: Effect of Model Complexity

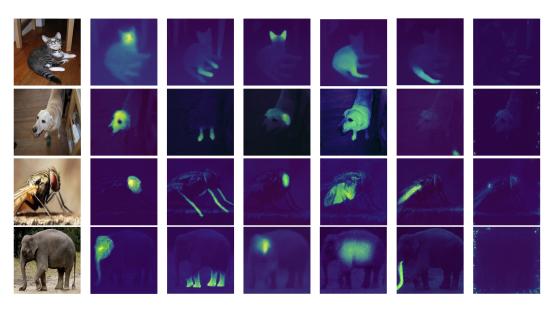


Figure 8: Visualization of Factor Tokens

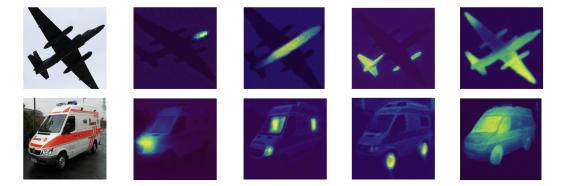


Figure 9: Visualization of Factor Tokens (Other Objects)