Learning Sparse Visual Representations via Spatial-Semantic Factorization

Theodore Zhengde Zhao[1] Sid Kiblawi[1] Jianwei Yang[2*] Naoto Usuyama[1] Reuben Tan[1]
Noel C Codella[1] Tristan Naumann[1] Hoifung Poon[1] Mu Wei[1]
[1]Microsoft, [2]xAI, *Work done at Microsoft

Abstract

*Self-supervised learning in vision is split between methods that learn strong semantics through invariance and methods that preserve spatial detail through reconstruction. We argue that this tension is largely induced by the dense grid representation itself. We propose **STELLAR**, which factorizes an image representation into sparse semantic tokens and their spatial assignment map. This separation lets the semantic tokens remain view-invariant while the localization matrix absorbs spatial equivariance, enabling both semantic alignment and image reconstruction within one latent space. STELLAR learns the factorized representation with low-rank reconstruction, online concept clustering, and optimal-transport alignment across views. With only 16 tokens, STELLAR achieves strong reconstruction and semantic quality simultaneously, including 2.60 FID and 79.10% ImageNet linear accuracy. The results show that sparse factorized latents can bridge discriminative and generative visual representation learning.*

1. Introduction

Visual concept discovery aims to extract compact, structured, and reusable representations of the visual world. Such representations are valuable across both discriminative and generative vision problems: they can support scene understanding, localization, retrieval, interpretation, and reconstruction from a small set of latent units. A central challenge, however, is to learn visual concepts that are simultaneously *semantic* and *grounded*. Concepts should remain stable across appearance-preserving transformations, but they must also preserve enough spatial structure for faithful image decoding and fine-grained understanding.

Learning visual representations has long been central to computer vision [5], yet modern encoders still represent images primarily as dense spatial grids [15, 18]. This format is effective for local processing, but it creates a persistent tension in self-supervised learning. Reconstruction-based methods such as MAE [20] preserve patch-level spatial detail and therefore support decoding, but often lag in semantic abstraction. In contrast, joint-embedding methods such as DINO [8] learn strong invariances and therefore strong semantics, but they suppress the spatial variation needed for faithful reconstruction. As a result, current methods often favor either semantic abstraction or spatial grounding rather than concept representations that support both.

We argue that this conflict is largely a consequence of the *representation format*. A dense latent is asked to satisfy two incompatible requirements at once. For reconstruction, the representation must remain spatially equivariant: shifting or cropping the image should induce corresponding changes in the latent. For semantic alignment across views, the representation should instead be invariant to those same transformations. This *invariance paradox* makes it difficult for a single dense representation to simultaneously support high-level semantics and pixel-level grounding.

Our key idea is to move away from dense grids and represent an image with two complementary sparse factors: *what* concepts are present, and *where* they appear. Specifically, we factorize the latent into a semantic matrix containing a small set of sparse concept tokens and a localization matrix describing their spatial support over image patches. This separation allows the semantic tokens to remain stable across views while the localization matrix captures the spatial change under transformations. Reconstruction is performed from the recomposed low-rank latent, so the model still preserves the information needed for detailed decoding.

This factorization also introduces a strong information bottleneck. Reconstructing an image from only a handful of sparse tokens forces the model to allocate capacity to transferable scene structure rather than to every local patch. As a result, STELLAR encourages a form of semantic triage: the model must explain the image using a compact set of concepts that are useful both for reconstruction and for downstream understanding. This directly matches the goal of visual concept discovery: the latent units are not merely compressed tokens, but semantically coherent and spatially grounded visual concepts.

Our contributions can be summarized as follows.

- We propose a form of sparse visual representation factorized into semantic concepts and spatial assignments.

- We introduce a self-supervised learning framework combining low-rank reconstruction, online concept clustering, and set alignment across views.
- We show that the learned sparse latents are both useful and interpretable: they support strong reconstruction and semantic performance while exhibiting coherent retrieval and localized concept grounding.

2. Related Work

Self-supervised vision learning. Modern SSL in vision broadly falls into two paradigms. Joint-embedding methods, including MoCo [19] and DINO [8], emphasize augmentation invariance and typically produce strong global semantics. Masked-image-modeling methods, including MAE [20] and SimMIM [31], preserve spatial structure for reconstruction, but their learned features often trail semantic-centric methods on discriminative tasks. Hybrid approaches such as iBOT [37] and DINOv2 [26] partially combine these ideas, but they still operate on dense features and do not reconstruct from a compact sparse latent.

Visual concept learning and sparse representations. Visual concept discovery seeks latent units that are compact, reusable, and interpretable. Sparse queries have been effective in supervised perception and multimodal models, including Sparse R-CNN [28], Mask2Former [12], and BLIP-2 [23]. TiTok [32] showed that images can be reconstructed from a small set of tokens. STELLAR differs in two ways: sparse tokens are the *primary* latent representation rather than an auxiliary interface, and they are learned in a purely visual self-supervised setting. The resulting latent is explicitly structured into concept identity and spatial support, facilitating grounded concept learning.

Factorized representations and interpretability. Unlike low-rank parameterizations used for parameter efficiency such as LoRA [21], STELLAR applies factorization directly to the image representation itself. This makes the latent structure semantically meaningful: one component encodes which concepts are present, while the other specifies where they appear. Such disentanglement offers a more interpretable interface than conventional dense grids and connects representation learning with concept-centric reasoning. The low-rank form is also related in spirit to convex semi-nonnegative matrix factorization [14].

Empirical dilemma. Current vision frameworks exhibit a persistent gap: models that excel at pixel-level reconstruction often produce weaker semantics [10], while models with top-tier semantic performance usually abandon reconstruction or make it secondary [2]. STELLAR is motivated by the hypothesis that this tradeoff is not inevitable, but largely induced by the dense representation format itself.

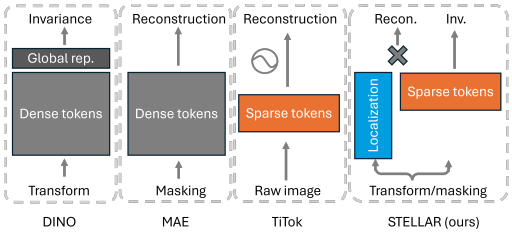

Figure 1. Comparison of learning different latent representations. STELLAR resolves the conflict by disentangling semantic tokens from spatial assignments.

3. Method

Factorized latent space. For an image X, conventional encoders produce a dense feature map $\boldsymbol{Z}(X) \in \mathbb{R}^{n\times d}$, as in modern ViT-style vision encoders [15]. STELLAR instead represents the image as

$$\boldsymbol{Z}(X) = \boldsymbol{L}(X)\boldsymbol{S}(X), \tag{1}$$

where $\boldsymbol{S} \in \mathbb{R}^{r\times d}$ contains $r \ll n$ sparse semantic tokens and $\boldsymbol{L} \in \mathbb{R}^{n\times r}$ is a localization matrix with nonnegative rows summing to one. Each dense patch feature is therefore expressed as a convex combination of the semantic tokens.

This factorization separates semantic identity from spatial geometry. Under a spatial transformation t_θ, the total variation of the latent decomposes as

$$\frac{\partial \boldsymbol{Z}(t_\theta \circ X)}{\partial \theta} = \left(\frac{\partial \boldsymbol{L}(t_\theta \circ X)}{\partial \theta}\right)\boldsymbol{S} + \boldsymbol{L}\left(\frac{\partial \boldsymbol{S}(t_\theta \circ X)}{\partial \theta}\right). \tag{2}$$

The first term carries spatial equivariance through the localization matrix, while the second term measures semantic variation. STELLAR is trained so that the semantic tokens remain stable across views and the localization matrix absorbs the spatial transformation.

Low-rank reconstruction. A lightweight decoder reconstructs the image from the recomposed low-rank latent:

$$\mathcal{L}_{\text{recon}} = \ell\big(\mathcal{D}(\boldsymbol{L}(X)\boldsymbol{S}(X)), X\big). \tag{3}$$

Because the decoder receives only r semantic tokens and their assignment map, the model must compress the image into transferable concepts rather than retain a dense patch-aligned code. This bottleneck is central to the semantic triage effect observed in STELLAR.

Vision concept clustering. To make sparse tokens semantically consistent across images, each token is projected and matched against a learnable prototype set $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]$. Given soft prototype assignments $q^i_{j,k}$ for token j of image i, we compute balanced Sinkhorn targets

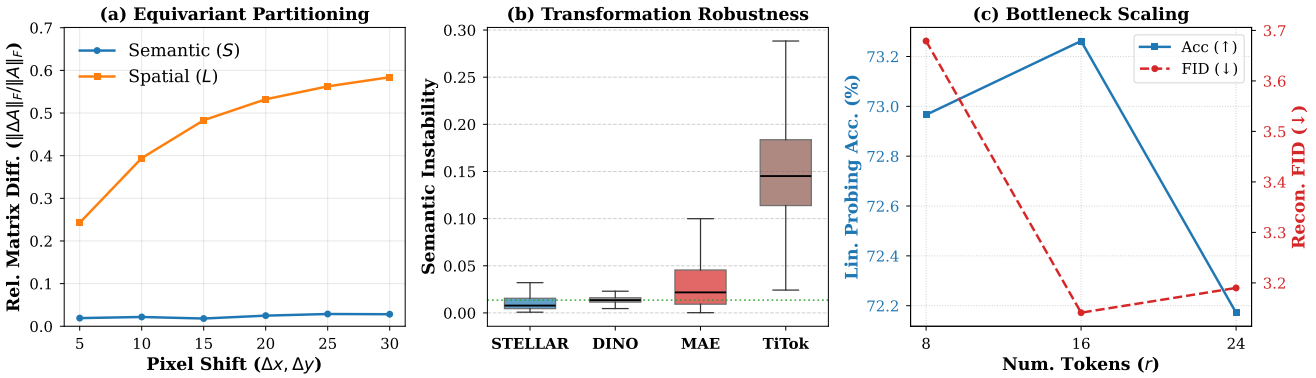

Figure 2. Probing the factorized representation. (a) Under controlled pixel shifts, the semantic matrix $\boldsymbol{S}$ remains nearly invariant while the localization matrix $\boldsymbol{L}$ changes smoothly. (b) Under random crops, STELLAR's sparse tokens are substantially more stable than reconstruction-only sparse latents. (c) Increasing the token count improves reconstruction but weakens semantic quality, showing a tradeoff.

$\tilde{q}^i_{j,k}$ following prior clustering work [7, 13] and optimize

$$\mathcal{L}_{\text{cluster}} = -\frac{1}{mr} \sum_{i,j,k} \tilde{q}^i_{j,k} \log q^i_{j,k}. \tag{4}$$

This organizes the sparse token set into reusable visual concepts instead of arbitrary latent slots.

Set alignment across views. For a transformed view, the sparse tokens have no canonical order. We therefore align the transformed token set $\{\boldsymbol{s}'_{j'}\}$ to the reference set $\{\boldsymbol{s}_j\}$. Let $\Theta_{j'j} = \|\boldsymbol{s}'_{j'} - \boldsymbol{s}_j\|_2$ be the pairwise cost matrix. We match prototype targets with entropy-regularized optimal transport, and supervise with transformed tokens $\sigma(j')$ via

$$\mathcal{L}_{\text{align}} = -\frac{1}{r} \sum_{j',k} \tilde{q}_{\sigma(j'),k} \log q'_{j',k}. \tag{5}$$

We additionally apply a KoLeo regularizer to spread tokens on the unit sphere and optionally align a CLS token for stronger global probing.

Training objective. The overall objective jointly optimizes reconstruction, clustering, and set alignment:

$$\mathcal{L} = a_1\mathcal{L}_{\text{recon}} + a_2\mathcal{L}_{\text{cluster}} + a_3\mathcal{L}_{\text{align}} + a_4\mathcal{L}_{\text{KoLeo}}. \tag{6}$$

In practice, we use a standard ViT backbone augmented with r learnable latent queries to produce $\boldsymbol{S}$. The dense patch features and sparse tokens are projected into a shared space, and their cosine similarities followed by a row-wise softmax produce $\boldsymbol{L}$. The decoder is a lightweight 6-layer ViT. This design keeps the framework simple while isolating the benefit of the factorized latent itself.

4. Experiments

We train STELLAR on ImageNet-1K without labels. Unless noted otherwise, the encoder is ViT-B with 16 latent queries and a lightweight 6-layer decoder. We evaluate both the sparse tokens and the induced dense feature map. When using a foundation prior, pretraining is also restricted to ImageNet-1K, with MAE as the default initialization.

4.1. Probing the factorized representation

Controlled translation experiments verify the decomposition in Eq. 2, as shown in Fig. 2. The semantic matrix $\boldsymbol{S}$ remains nearly invariant under pixel shifts, while the localization matrix $\boldsymbol{L}$ changes smoothly with the transformation. Under random resized crops, STELLAR's sparse tokens are markedly more stable than reconstruction-only sparse latents such as TiTok, confirming that explicit factorization is crucial for separating semantics from geometry. We also vary the token count r and find a clear tradeoff: increasing r improves reconstruction but weakens semantic quality, with $r = 16$ giving the best balance.

Beyond aggregate metrics, the learned sparse tokens correspond to reusable visual concepts with coherent spatial support. Fig. 3 shows that retrieved examples within a concept remain semantically consistent, while localization maps highlight the corresponding image regions. This is central to the visual concept discovery setting: the sparse units are compact, semantically reusable, and grounded in image space rather than being opaque compressed codes. In STELLAR, concept discovery emerges from the joint pressure of cross-view alignment and low-rank reconstruction: a useful concept must be stable enough to match across views, yet specific enough to explain localized image content through its spatial map. This indicates that STELLAR does not merely compress the image for reconstruction, but learns concepts that capture useful semantic structure.

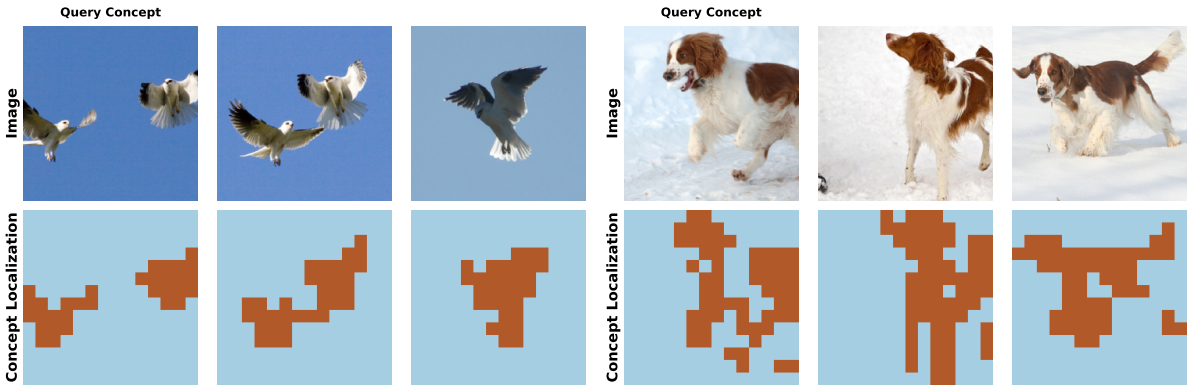

Figure 3. Learned visual concepts from STELLAR. For each query concept, we show retrieved examples together with thresholded localization maps. The retrieved images are semantically coherent, while the localization maps remain spatially selective, indicating that STELLAR learns transferable and grounded visual concepts.

Table 1. Main reconstruction-semantics tradeoff on IN1K.

Model	#Tks	FID↓	LPIPS↓	Lin.↑	kNN↑
DINO (CLS)	1	–	–	76.46	74.69
DINO (dense)	196	3.27	0.2121	70.31	54.41
MAE	196	3.02	**0.2071**	66.32	25.82
TiTok-32	32	2.75	0.3281	33.42	7.30
TiTok-64	64	1.99	0.2571	32.87	7.29
STELLAR-B	16	3.06	0.2077	**73.26**	**67.25**
STELLAR-H	16	**2.60**	0.1729	79.10	77.31

Table 2. Downstream results with linear probing.

Model	Segmentation		Classification	
	ADE20K	VOC	IN1K	GlaS
DINO ViT-B	26.87	79.29	76.46	95.00
MAE ViT-B	30.91	76.43	66.32	93.75
iBOT ViT-B	31.78	77.06	76.40	96.25
TiTok-64 ViT-B	–	–	32.87	97.50
STELLAR ViT-B	31.33	81.83	73.26	95.00
STELLAR ViT-L	34.02	85.90	76.94	97.50
STELLAR ViT-H	36.66	85.66	79.10	92.50

4.2. Reconstruction-semantics tradeoff

Table 1 compares STELLAR with dense and sparse baselines. With only 16 tokens, STELLAR substantially outperforms dense baselines in semantic probing while remaining competitive in reconstruction. At large scale, STELLAR reaches 79.10% ImageNet-1K linear accuracy together with 2.60 FID. Compared to MAE, it preserves similar reconstruction quality from a much smaller latent.

4.3. Downstream understanding and ablations

On downstream image understanding, STELLAR is particularly strong on fine-grained dense tasks. As shown in Table 2, STELLAR improves over MAE on ADE20K and Pascal VOC and remains competitive on Cityscapes, indicating that sparse token modeling implicitly organizes the dense backbone features into region-aware semantics. On global classification, STELLAR consistently outperforms reconstruction-centric baselines, while the gap to top joint-embedding methods mainly appears on simple object-centric datasets where mean-pooling over multiple concept tokens can dilute discriminative cues.

Ablations further show that all three ingredients are necessary. Removing low-rank reconstruction weakens both global and dense understanding; removing concept clustering or set alignment leads to major collapse in semantic quality, KoLeo regularization provide smaller but consistent gains. Detailed tables are deferred to the appendix.

5. Conclusion

STELLAR shows that the apparent conflict between semantic invariance and spatial grounding is largely a consequence of dense visual representations. By factorizing the latent into sparse semantic tokens and spatial assignments, it supports both cross-view semantic alignment and image reconstruction within one SSL framework.

The key mechanism is semantic triage: because reconstruction must proceed through a low-rank latent, the model allocates capacity to transferable concepts rather than every local patch. As a result, the latent units are not merely efficient tokens, but visual concepts with coherent semantics and grounded localization.

From the perspective of visual concept discovery, the representation is compact, structured, and interpretable, while useful for reconstruction and image understanding. We view sparse factorized latent as a promising direction for representation learning with interpretability, grounding, and versatility across discriminative and generative tasks.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European conference on computer vision*, pages 456–473. Springer, 2022. 11

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. 2, 7, 11

[3] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pages 1298–1312. PMLR, 2022. 11

[4] Hangbo Bao, Li Dong, and Furu Wei. Beit: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 11

[5] 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. 1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 13

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 3

[8] 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*, pages 9650–9660, 2021. 1, 2, 11

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021. 11

[10] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024. 2, 7

[11] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021. 13

[12] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2, 7

[13] Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latent patches for improved masked image modeling. *arXiv preprint arXiv:2502.08769*, 2025. 3, 7

[14] Chris HQ Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2008. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 8

[16] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*, 2024. 11

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 11

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 7

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2, 7, 11

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2, 7

[22] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *Advances in Neural Information Processing Systems*, 35:14290–14302, 2022. 7, 11

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2, 7

[24] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 10

[25] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. Supervised dictionary learning. *Advances in neural information processing systems*, 21, 2008. 7

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 7, 11

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 12

[28] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021. 2, 7, 13

[29] Chenxin Tao, Xizhou Zhu, Weijie Su, Gao Huang, Bin Li, Jie Zhou, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2132–2141, 2023. 11

[30] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 11

[31] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022. 2, 7, 11

[32] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024. 2, 7, 11

[33] Le Zhang, Qian Yang, and Aishwarya Agrawal. Assessing and learning alignment of unimodal vision and language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14604–14614, 2025. 12

[34] Mingtian Zhang, Tim Z Xiao, Brooks Paige, and David Barber. Improving vae-based representation learning. *arXiv preprint arXiv:2205.14539*, 2022. 7

[35] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[36] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 12

[37] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2, 7, 11

A. Extended Contents

A.1. Extended Related Work

Self-supervised Learning. Modern SSL generally falls into two paradigms. *Joint Embedding* (JE) methods, such as the MoCo [19] and DINO [26] families, prioritize global invariance via multi-view alignment, yielding strong semantics but often losing spatial grounding. Conversely, *Masked Image Modeling* (MIM), exemplified by MAE [20] and SimMIM [31], emphasizes spatial equivariance through pixel reconstruction. While hybrids like iBOT [37] and DINOv2 [26] attempt to combine these objectives, they still rely heavily on global invariance and forgo pixel reconstruction.

Sparse Representation. A growing body of work replaces dense feature maps with compact embeddings. Sparse R-CNN [28] and Mask2Former [12] utilize sparse queries for supervised tasks, while BLIP-2 [23] and TiTok [32] employ sparse tokens for vision–language or generative efficiency. SemMAE [22] utilizes sparse tokens to guide masking using a pretrained teacher. Unlike these methods, STELLAR treats sparse tokens as the primary latent representation and learns in a self-supervised manner.

Disentanglement & Low-rank Factorization. The assumption that high-dimensional data lie on low-dimensional manifolds is foundational to dictionary learning [25]. In deep learning, low-rank constraints are typically applied to weights for efficiency (e.g., LoRA [21]). STELLAR differs by applying low-rank factorization to the feature map itself, disentangling "what" from "where."

The Empirical Dilemma. Current vision frameworks face a persistent gap: models excelling at pixel-level reconstruction often produce weaker semantic representations [10, 34], while those achieving top-tier semantics often abandon reconstruction to avoid low-level shortcuts [2, 13]. We demonstrate that by factorizing the latent representation, it is possible to achieve strong performance on both image understanding and reconstruction.

A.2. Formal Preliminaries

Representation learning involves encoding an image $X \in \mathcal{X}$ to latent features $\boldsymbol{Z}(X)$ for downstream tasks. Traditionally, vision representations take a dense spatial form:

$$\boldsymbol{Z} \in \mathbb{R}^{n \times d},$$

where $n = h \times w$ denotes the number of patches on a dense grid that partitions the image. Each grid location is represented by a feature vector $\mathbf{z}_i := \boldsymbol{Z}_{i,:} \in \mathbb{R}^d$ for $1 \le i \le n$. Most vision architectures also incorporate a global representation $\mathbf{z}_0 \in \mathbb{R}^d$, typically obtained via global pooling or a specialized `[CLS]` token that undergoes self-attention with patch tokens.

Ideally, we want $\boldsymbol{Z}$ to serve as a holistic representation of the image X, which retains sufficient information about the image details while at the same time possesses rich semantics for downstream tasks. Mathematically, we define such a representation as follows:

- **Reconstruction**: There exists a decoder $\mathcal{D}$ such that $\mathcal{D}(\boldsymbol{Z}(X)) \approx X$. This ensures the representation is spatially and texturally grounded in the physical input.
- **Semantics**: For a downstream task with joint distribution $(X, Y) \sim \mathcal{X} \times \mathcal{Y}$, there exists a simple predictor $f \in \mathcal{F}$ (e.g., a linear layer) such that the expected task loss $\mathbb{E}_{(X,Y)}\big[\mathcal{L}(f(\boldsymbol{Z}(X)), Y)\big]$ is minimized using frozen features. Typically Y reflects human perception.

Current SSL paradigms are caught in a fundamental *Invariance Paradox*. To learn high-level semantics, joint-embedding methods impose invariance to spatial transformations, even when the image is heavily cropped. On the other hand, reconstruction requires spatial detail, because every pixel shift requires a different set of features for precise reconstruction. This results in representations that are inherently equivariant to transformation.

Let $\mathcal{T}$ be a group of spatial transformations (e.g., translations), and $t_\theta \in \mathcal{T}$ be parametrized by θ. A representation $\boldsymbol{Z}(X)$ suffers from the invariance paradox if it must simultaneously satisfy two contradictory constraints:

- **Semantic Invariance:**

$$\left\| \frac{\partial}{\partial \theta} \boldsymbol{Z}(t_\theta \circ X) \right\|_F \approx 0.$$

- **Spatial Equivariance:** To allow high-fidelity reconstruction, the representation must track spatial shifts: $\mathcal{D}(\boldsymbol{Z}(t_\theta \circ X)) \approx t_\theta \circ X$. With chain rule and matrix norm inequalities,

$$\left\| \frac{\partial}{\partial \theta} \boldsymbol{Z}(t_\theta \circ X) \right\|_F \gtrsim \frac{\left\| \frac{\partial (t_\theta \circ X)}{\partial \theta} \right\|_F}{\sigma_{\max}\left(\frac{\partial \mathcal{D}}{\partial \boldsymbol{Z}}\right)} > 0.$$

A.3. Extended Model Design Details

The framework only specifies the latent space and does not prescribe a particular encoder or decoder architecture. In the full submission, we used a simple design with common modules to obtain $\boldsymbol{S}$ and $\boldsymbol{L}$.

For the encoder part, we use an existing ViT [15] as the backbone, and equip it with r learnable latent query vectors, which are passed to the transformer blocks alongside the patch tokens. Processed jointly by the ViT, the latent queries produce sparse tokens $\boldsymbol{S} \in \mathbb{R}^{r \times d}$.

To obtain the localization matrix $\boldsymbol{L} \in \mathbb{R}^{n \times r}$ associated with the sparse tokens, we use the dense feature map $\boldsymbol{U} \in \mathbb{R}^{n \times d}$ output from the image patches. We project both $\boldsymbol{S}$ and $\boldsymbol{U}$ into a shared embedding space and compute their pairwise cosine similarities, followed by a softmax normalization with temperature τ_{spatial} along the second dimension:

$$\boldsymbol{L} = \text{softmax}(\text{cossim}(\boldsymbol{U}\boldsymbol{W}_1, \boldsymbol{S}\boldsymbol{W}_2)/\tau_{\text{spatial}}) \,. \tag{7}$$

$\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are learnable linear projections, and τ_{spatial} controls the sharpness of the spatial distribution. This mapping is structurally similar to attention weights in a single-head cross-attention layer, up to the use of ℓ_2 normalization and an explicit temperature parameter. Therefore, the latent representation $\boldsymbol{Z} = \boldsymbol{L}\boldsymbol{S}$ can be viewed as rebuilding a dense feature map for reconstruction by cross-attending to only r sparse concept tokens.

All together, the encoder $\mathcal{E}$ includes ViT transformer blocks, r learnable latent query vectors, and projection layers $\boldsymbol{W}_1, \boldsymbol{W}_2$. The decoder $\mathcal{D}$ is a 6-layer lightweight ViT reconstructing the image patches. The STELLAR framework can be used on a pretrained ViT such as MAE or DINO to leverage the foundation prior and shape it into a sparse holistic representation. It can also be trained from a random prior and reach competitive spatial, semantic, and reconstruction quality.

B. Full Experimental Results

B.1. Full Reconstruction–Semantics Comparison

The full submission contained the following reconstruction/semantic comparison table and its accompanying discussion.

Table 3. Reconstruction and semantic metrics on IN1K of STELLAR and baseline models. For reference, we also report semantic metrics of the global representation from DINO, and huge-size STELLAR. Model sizes are ViT-B by default, with larger sizes indicated in parentheses. *: TiTok used its native ViT decoder of larger size.

		Reconstruction		Semantics	
Model	# tks	FID ↓	LPIPS ↓	Lin.	kNN
DINO	1	-	-	76.46	74.69
DINO	196	3.27	0.2121	70.31	54.41
MAE	196	3.02	**0.2071**	66.32	25.82
TiTok*	32	2.75	0.3281	33.42	7.30
TiTok*	64	1.99	0.2571	32.87	7.29
ours	16	3.06	0.2077	**73.26**	**67.25**
ours	196	**2.85**	0.2085	72.21	64.71
ours(H)	16	2.60	0.1729	79.10	77.31

On reconstruction, STELLAR shows comparable FID and LPIPS loss [35] to the dense feature map from MAE, with 90% reduction in latent size. Although TiTok achieved lower FID with a much larger decoder, it shows higher LPIPS loss, indicating poor spatial consistency. In contrast, STELLAR exhibits superior reconstruction locality even with fewer tokens. The full-rank dense feature map $\boldsymbol{U}$ from the ViT in STELLAR shows even lower reconstruction FID, while dropping in semantic quality. When scaling to a huge-sized ViT, STELLAR achieves top reconstruction and semantic quality, even without decoder finetuning.

B.2. Full Downstream Benchmarking Table

Lastly, the full submission benchmarked STELLAR in classical image-understanding tasks with linear probing on frozen features, comparing against other ImageNet-pretrained SSL models.

Table 4. **Evaluation of Fine-grained and Global Image Understanding.** We evaluate semantic segmentation (mIoU %) and classification accuracy (%) via linear probing on frozen features. We used the dense feature map from the backbone for all segmentation tasks and all models. **Bold**: best with ImageNet training. Underline: best in architectural class.

		SSL Type		Segmentation (mIoU)			Classification (Acc)			
Model	**Arch.**	**Target**	**Method**	**ADE20K**	**CitySc**	**VOC**	**IN1K**	**Pets**	**Food**	**GlaS**
Semantic-Centric (Joint Embedding / Invariance)										
BYOL	RN-50	GLOBAL	DISTILL	18.43	18.66	63.89	70.39	82.77	64.57	95.00
MOCO V3	VIT-B	GLOBAL	CONTR.	29.45	25.13	74.08	74.31	91.14	77.47	**97.50**
DINO	VIT-B	GLOBAL	DISTILL	26.87	26.82	79.29	76.46	**93.84**	79.28	95.00
MSN	VIT-B	GLOBAL	MASKING	26.66	25.39	68.59	73.65	75.91	68.93	92.50
DENSECL	RN-50	DENSE	CONTR.	23.08	18.63	70.95	61.10	72.99	59.16	85.00
DATA2VEC	VIT-B	DENSE	LAT-MIM	22.03	23.49	61.33	54.90	26.47	34.40	73.75
SIAMESEIM	VIT-B	DENSE	LAT-MIM	29.24	26.52	81.38	74.97	91.61	71.01	91.25
I-JEPA	VIT-H	DENSE	LAT-MIM	21.57	18.59	74.13	71.72	84.68	70.34	87.50
IBOT	VIT-B	GL+DE	DIST+MIM	31.78	25.69	77.06	76.40	92.40	78.08	96.25
IBOT	VIT-L	GL+DE	DIST+MIM	33.26	26.37	77.57	78.53	92.12	**81.07**	96.25
Image-Centric (Reconstruction)										
BEIT	VIT-B	DENSE	TOK MIM	11.58	18.90	27.44	32.94	36.20	54.49	90.00
BEIT	VIT-L	DENSE	TOK MIM	12.64	20.37	25.48	36.77	36.71	56.03	90.00
SIMMIM	SWIN-B	DENSE	PIX MIM	12.46	17.23	35.14	24.77	27.39	40.94	77.50
MAE	VIT-B	DENSE	PIX MIM	30.91	29.44	76.43	66.32	81.58	70.40	93.75
MAE	VIT-L	DENSE	PIX MIM	34.36	32.53	77.79	73.09	84.30	76.22	95.00
MAE	VIT-H	DENSE	PIX MIM	36.16	**35.21**	78.07	75.22	84.96	78.36	95.00
SEMMAE	VIT-B	DENSE	PIX MIM	3.52	25.48	48.33	43.84	56.99	58.90	92.50
TITOK-64	VIT-B	SPARSE	SPRS REC	–	–	–	32.87	42.06	43.68	**97.50**
TITOK-32	VIT-L	SPARSE	SPRS REC	–	–	–	33.42	27.83	38.83	78.75
Our Method (Sparse Factorized Modeling)										
STELLAR	VIT-B	SPARSE	INV+REC	31.33	27.74	81.83	73.26	89.70	74.09	95.00
STELLAR	VIT-L	SPARSE	INV+REC	34.02	31.32	**85.90**	76.94	92.53	74.78	**97.50**
STELLAR	VIT-H	SPARSE	INV+REC	**36.66**	33.30	85.66	**79.10**	92.53	77.43	92.50
Larger Scale Pretraining Beyond ImageNet (Reference Only)										
AIM	600 M	DENSE	IMAGE AR	29.00	27.04	64.55	63.78	64.68	75.19	98.75
AIM	1 B	DENSE	IMAGE AR	29.59	27.05	63.90	66.86	64.21	77.96	96.25
DINOV2	VIT-B*	GL+DE	DIST+MIM	40.10	34.66	89.52	82.82	95.59	91.08	98.75
DINOV2	VIT-L*	GL+DE	DIST+MIM	40.45	32.07	89.19	84.23	96.08	92.94	98.75

As discussed in the full version, the feature map from STELLAR achieves superior performance on ADE20K and Pascal VOC, showing strong fine-grained understanding despite not applying SSL objectives directly to the dense feature map $\boldsymbol{U}$. Sparse token modeling implicitly organizes the feature map into semantic regions: to reconstruct the image, each token must encode information covering all spatial parts of the scene, resulting in region-aware representations. While MAE leads in Cityscapes, STELLAR follows closely with performance comparable to MAE and DINOv2.

On global image-understanding tasks, STELLAR achieves the highest accuracy on IN1K at large model scale, but smaller variants underperform methods such as DINO, which explicitly optimize for global representations. In general, STELLAR outperforms image reconstruction models and most JE methods, but trails behind top JE models in global semantics. Because STELLAR does not model the image as a single concept, averaging token features can dilute discriminative information, which is particularly detrimental on object-centric datasets like Pets and Food. On histopathology images involving complex tissue microenvironments, however, STELLAR achieves the best performance, suggesting that it excels at modeling complex multi-object scenes.

Table 5. **Ablation.** We isolate the impact of each objective on semantic abstraction (IN1K) and spatial grounding (ADE20K), and reconstruction (FID). *Default* denotes the full STELLAR framework. All results are based on ViT-B.

	Recon.	Cluster	Set Align	CLS Align	KoLeo	rFID ↓	IN1K ↑	ADE ↑
DEFAULT	✓	✓	✓	✓	✓	**3.14**	**73.26**	**31.33**
Impact of Individual Components								
(A)	✗	✓	✓	✓	✓	—	72.44 (-0.82)	29.94 (-1.39)
(B)	✓	✗	✗	✗	✓	3.21 (+0.07)	52.07 (-21.19)	20.46 (-10.87)
(C)	✓	✓	✗	✗	✓	8.95 (+5.81)	2.73 (-70.53)	1.93 (-29.39)
(D)	✓	✗	✓	✓	✓	3.62 (+0.48)	42.14 (-31.12)	18.90 (-12.43)
(E)	✓	✓	✓	✗	✓	3.26 (+0.12)	70.79 (-2.47)	30.20 (-1.12)
(F)	✓	✓	✓	✓	✗	3.25 (+0.11)	72.05 (-1.21)	30.10 (-1.23)

B.3. Full Ablation Analysis and Prior Study

Low-rank approximated reconstruction. Removing the low-rank reconstruction objective reduces both global and fine-grained understanding. Since the remaining objectives resemble typical SSL methods, the model still retains reasonable global performance, but fine-grained understanding suffers more. This indicates that low-rank reconstruction encourages sparse tokens to serve as holistic representations covering the entire image.

Concept clustering. Eliminating online clustering and set alignment leads to a sharp drop in understanding, highlighting the necessity of structuring sparse tokens into view-invariant concepts. Even when the alignment loss is present, missing the clustering loss still leads to collapse.

Set alignment. Training with only reconstruction and clustering collapses, underscoring the critical role of set-concept alignment. Additional alignment on the CLS token primarily benefits global classification but has limited effect on spatial grounding. KoLeo regularization consistently improves all tasks at a smaller but consistent level.

Table 6. Evaluating STELLAR trained from different foundational priors. *Base* represents the performance of the original backbone.

Prior	Recon FID ↓	Semantic (IN1K) BASE	Semantic (IN1K) +STELLAR	Spatial (ADE20K) BASE	Spatial (ADE20K) +STELLAR
MAE	**3.14**	66.32	**73.26 (+6.9)**	30.91	**31.33 (+0.4)**
DINO	3.31	76.46	73.31 (-3.2)	26.87	**28.17 (+1.3)**
RAND	3.21	–	65.28	–	28.10

We also studied the effect of different pretrained foundation priors. STELLAR substantially boosts the semantic quality from an MAE prior and the spatial grounding from a DINO prior. The semantic performance falls to a similar level despite different foundation priors. When training from random prior, STELLAR still reaches semantics at roughly MAE level and spatial understanding similar to that from a DINO prior, while reconstruction quality remains consistent.

C. Implementation Details

C.1. STELLAR Training

We trained STELLAR with ViT models at size base, large, and huge, along with the latent queries, projection layers, clustering head, and a 6-layer ViT decoder. In the default setting, we initialized the ViT part in the encoder from public MAE checkpoint, and trained for 150 epochs for STELLAR-B, 100 epochs for STELLAR-L, and 50 epochs for STELLAR-H. We used 16 NVIDIA A100-80GB with batch size 128 each, totaling 2048. We used AdamW[24] with base learning rate 1.5×10^{-4} for STELLAR-B, and 5×10^{-5} for STELLAR-L and STELLAR-H.

For concept clustering, we used 16384 prototypes for sparse and CLS tokens each. The projector is a 2-layer MLP before the prototype layer. We used 3 steps of Sinkhon-Knopp algorithm. The temperature in sparse-dense cosine similarity softmax is 0.06. We used 6-8 random masked views to align the sparse tokens, and additional 6-8 local crops to align the CLS token.

Global views are of random scale 36% to 100%, and local view are of random scale 6% to 36%. We also apply color jittering, grascaling and Gaussian blurring.

In the ablation study of random prior, we trained the model from scratch and used exponential moving average (EMA) updated momentum encoder to encode the target prototype assignments in the warm-up stage. We EMA updated the full encoder (ViT, latent queries, projection, clustering head with momentum 0.996. The momentum encoder was used to encode a global view of the image into target prototype assignments, for both clustering loss and alignment loss. The masking ratio was 0.6 in the warm-up stage, and 0.8 during standard training. We trained the model with 150 epochs of EMA warm-up and 75 epochs of standard training.

C.2. Evaluation Protocol

For STELLAR and all baseline models, we evaluated the frozen feature from the pretrained model with linear probing. We used layer norm in classification tasks, and batch norm in segmentation tasks, followed by a single linear layer predicting the class of the image or patch. For all benchmarks, we split 10% from the training set for validation. We tuned hyper-parameter with learning rate $1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}$, and batch size 64, 128, 256, 512, 1024, 2048, 4096, 8192.

As the SSL methods varies across different baseline models, for classification tasks we used the mean-pooled feature from the representations where the corresponding SSL method was performed, e.g. the global CLS token for DINO, and dense patch tokens for MAE. We noticed the linear probing accuracy can vary depending on the pooling choice, and conducted experiments by using different types of tokens for each model, with results in Table 7. We observed that the SSL-ed are typically the best choice for linear probing, except for iBOT, which highly relies on the global CLS token for classification, even though the model was trained with MIM. In contrast, STELLAR and MAE are relatively more robust to token choices.

Table 7. ImageNet-1K linear probing accuracy (%) by pooling different tokens. We mark in **bold** the tokens on which the specific SSL method was applied, and the top accuracy for each method.

	DINO		MAE		iBOT			STELLAR (ours)	
tokens	**global**	dense	global	**dense**	global	dense	**gl.+de.**	**sparse**	dense
lin. acc.	**76.46**	70.31	65.61	**66.32**	**76.40**	71.44	71.58	**73.26**	72.21

Table 8. List of baseline models and SSL method type.

Model	Reference	Method	SSL space	SSL tokens
BYOL	[17]	augmentation alignment	latent	global
MoCo v3	[9]	contrastive learning	latent	global
DINO	[8]	augmentation alignment	latent	global
MSN	[1]	masked alignment	latent	global
DenseCL	[30]	contrastive learning	latent	dense
Data2Vec	[3]	latent MIM	latent	dense
SiameseIM	[29]	latent MIM	latent	dense
IJEPA	[2]	latent MIM	latent	dense
iBOT	[37]	align + latent MIM	latent	global+dense
BEIT	[4]	token MIM	image	dense
SimMIM	[31]	pixel MIM	image	dense
MAE	[20]	pixel MIM	image	dense
SemMAE	[22]	pixel MIM	image	dense
TiTok	[32]	reconstruction + clustering	image	sparse
AIM	[16]	autoregressive	image	dense
DINOv2	[26]	align + latent MIM	latent	global+dense

D. Additional Results

D.1. Effect of pretraining data

We pretrained separate STELLAR versions on ImageNet-1K, Places365 [36] and compared their linear probing performance in Table 9.

Table 9. Effect of pretraining data.

	linear probing acc.	
Pretraining data	ImageNet-1K	Places 365
ImageNet-1K	76.94	49.25
Places365	66.08	51.98

D.2. Semantics from different features

We conducted linear probing of different mean-pooled features of different types, and compared in Table 10. Sparse feature showed strongest global understanding quality.

Table 10. Semantics in different features

Feature	sparse	cls	dense
IN-1K lin. acc (%)	73.26	72.23	72.21

D.3. Concept alignment with language

Inspired by [33], we used frozen feature from STELLAR and aligned with the text tower of CLIP [27] with a single attention pooled probing layer. The evaluation on vision language tasks with comparison to baseline models are shown in Table 11.

Table 11. Language alignment evaluation.

	IN-1K 0-shot		MS COCO		Winoground		MMVP
	@1	@5	T2I	I2T	Text	Image	Avg.
MAE	23.18	50.43	11.28	13.46	20.75	9.00	19.26
iBOT	50.01	80.43	20.79	29.38	24.75	12.00	18.52
STELLAR	51.53	80.04	17.94	22.34	26.25	8.25	19.26
CLIP	72.7	-	43.0	59.7	30.5	11.5	20.0

D.4. Finetuning

We performed finetuning for STELLAR on ImageNet-1K classification and ADE20K segmentation, and compared with baseline models. We used the same evaluation protocol as in Sec. C.2, with the backbone unfrozen and finetuned for 75 epochs. We used ViT-B for all models. The finetuning results are shown in Table 13. STELLAR showed consistent performance gain across different tasks, and close to the top model iBOT with slight difference.

D.5. Efficiency analysis

To analyze the efficiency of the STELLAR framework, we printed the processing time of the main components in the STELLAR framework with one A100 GPU at different batch sizes. Encoding the main global view of the image takes up most of the processing time, followed by encoding the masked views (8 views at 80% masking ratio) and decoding to the original

Table 12. Finetuning performance in ImageNet-1K classification accuracy and ADE20K segmentation mIOU (%). We show in parentheses the gain over the respective linear probing results.

Model	ImageNet-1K Acc.	ADE20K mIOU
DINO	79.58 (+3.12)	39.22 (+12.35)
MAE	77.75 (+11.43)	40.33 (+9.42)
iBOT	80.72 (+9.14)	42.76 (+10.97)
STELLAR	80.05 (+6.78)	41.98 (+10.65)

image. The Sinkhorn-Knopp algorithm used for clustering and the Sinkhorn algorithm used in optimal transport matching take up much less amount of time, and their total processing time stay at similar level when increasing the batch size.

In comparison to the Sinkhorn matching algorithm we used in our experiments, we show the processing time using an alternative Hungarian matching algorithm commonly used in previous literature such as Sparse R-CNN [28], DETR [6] and MaskFormer [11]. As the implementation of the exact matching is not scalable with GPU parallelization, it's computational time increases linearly with the batch size. At batch size 64, it is already 6 times of the encoder processing, while the Sinkhorn algorithm is over 100 times faster. For this reason, we added a small entropy regularization term in the bipartite matching objective, allowing us to use the Sinkhorn algorithm for efficient matching with GPU parallelization.

Table 13. Processing time (s) of the main components in the STELLAR framework with one A100 GPU at different batch sizes. In comparison to the Sinkhorn matching algorithm we used in our experiments, we show the processing time using an alternative Hungarian matching algorithm commonly used in previous literature (shown in gray).

Batch size	4	8	16	32	64
Encoder	8.2×10^{-3}	9.1×10^{-3}	1.4×10^{-2}	2.0×10^{-2}	3.2×10^{-2}
Decoder	4.6×10^{-3}	6.8×10^{-3}	8.8×10^{-3}	1.2×10^{-2}	1.5×10^{-2}
Mask encoding	7.9×10^{-3}	8.9×10^{-3}	1.1×10^{-2}	1.8×10^{-2}	1.7×10^{-2}
SK clustering	3.4×10^{-4}	3.4×10^{-4}	3.4×10^{-4}	3.7×10^{-4}	3.9×10^{-4}
Sinkhorn matching	1.4×10^{-3}	1.4×10^{-3}	1.4×10^{-3}	1.4×10^{-3}	1.2×10^{-3}
Hungarian matching	5.7×10^{-3}	1.7×10^{-2}	4.0×10^{-2}	9.0×10^{-2}	1.8×10^{-1}

E. Additional Illustration

See Fig. 4 and 5.

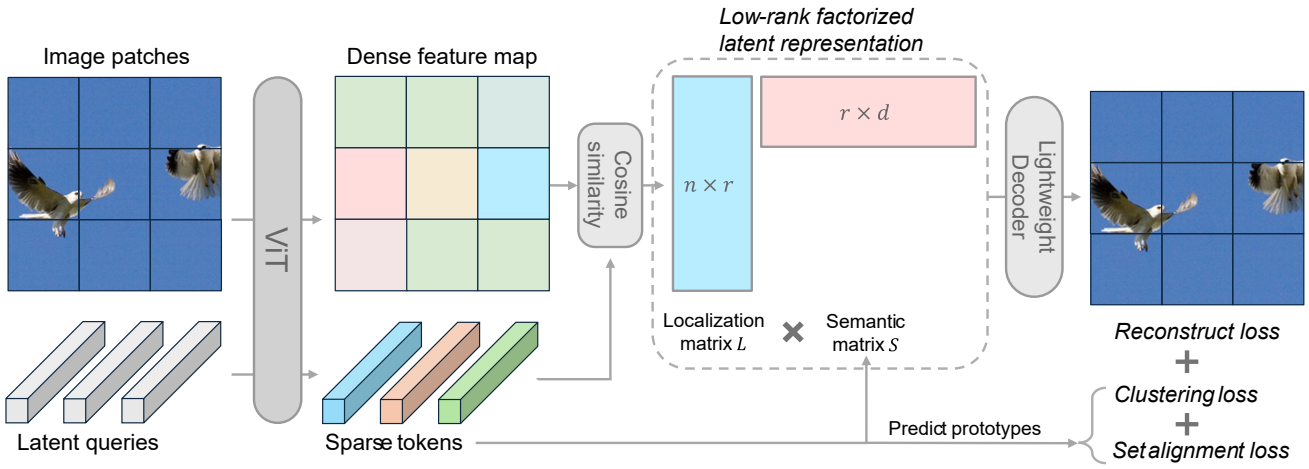

Figure 4. The STELLAR framework. We use a vanilla ViT to extract sparse tokens from an image, and model the latent representation as a low-rank matrix factorization, ensuring reconstruction of the original image. Clustering loss and set alignment loss are applied on the disentangled sparse tokens.

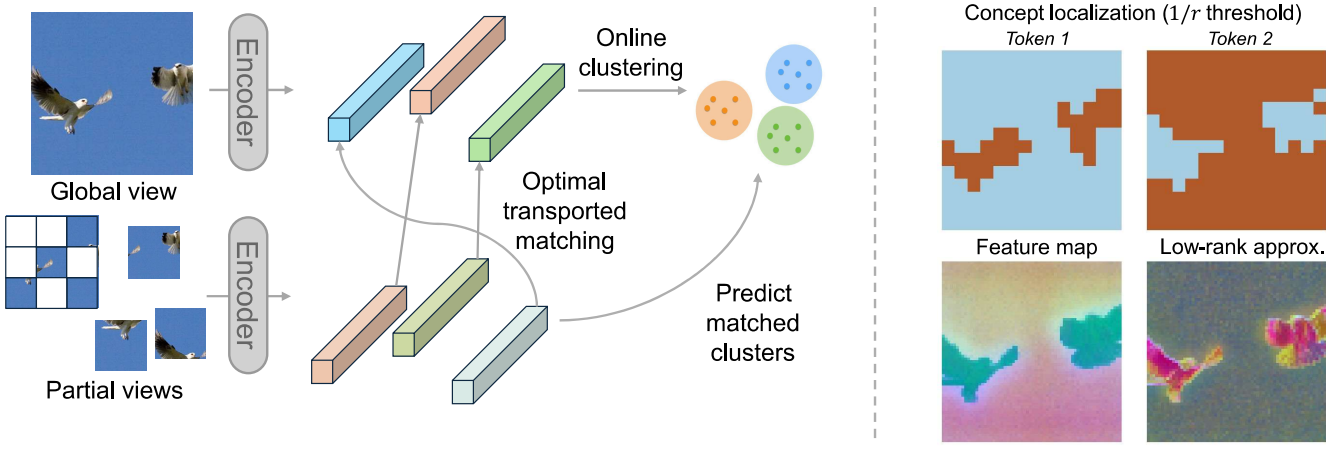

Figure 5. Left: Concept clustering and alignment workflow. Right: visualization of learned representation.