LexDINO: Language-Grounded Prototypes for Visual Pretraining

Anonymous authors
Paper under double-blind review

Abstract

A general-purpose visual encoder should transfer effectively across recognition, dense prediction, and generative vision-language tasks. Existing pretraining approaches offer complementary strengths across these settings: visual self-supervision (e.g., DINO) is particularly effective for dense prediction, while image-text supervision (e.g., CLIP) often yields better transfer to generative vision-language models (VLMs). We introduce LexDINO, which brings language-derived semantic structure into DINO's visual self-distillation to improve vision-language transfer while retaining its superior visual capabilities. An independent caption corpus is clustered in an LLM's feature space, and the resulting centroids replace DINO's learned image-level prototypes. A second bank represents the same caption groups in a frozen image-text model's joint space, allowing its image tower to ground the corresponding DINO assignments. Semantic guidance therefore acts through DINO's own prediction space, alongside cross-view self-distillation and masked patch learning, without directly regressing backbone features or requiring captions paired with the training images. With the ImageNet-1k image corpus, backbone, and training schedule held fixed, LexDINO improves over a DINOv3 trained in the same way by 2.0 points in k-NN accuracy and 3.3 points in ADE20K mIoU. Under the same downstream VLM training recipe, it improves all twelve benchmarks, including a 25.4-point gain in the five-task OCR average, and its LLM-derived prototype bank outperforms a SigLIP2-text bank by 6.7 OCR points. At scale, a roughly 1B parameter LexDINO encoder, trained with a fraction of DINOv3's pretraining compute, achieves best or tied performance on all five dense benchmarks among the evaluated models below 2B parameters. Batch-independent sparse targets and full-grid patch supervision simplify its distillation into ViT-L/16 and ViT-B/16 students. Both students outperform size-matched DINOv3 encoders in recognition and VLM transfer while retaining strong dense prediction.

1 Introduction

A general-purpose visual encoder should support visual recognition and dense prediction while also serving as an effective vision tower for a generative vision-language model (VLM). It should distinguish objects and preserve the spatial detail needed for segmentation and depth estimation, while providing features that a language model can use to answer questions, read text, and interpret documents. Progress on one set of tasks does not necessarily carry over to the other. The DINO family learns strong global and dense visual representations through visual self-supervision (Oquab et al., 2024; Siméoni et al., 2026). Yet controlled comparisons in generative VLMs find that image–text encoders often serve as more effective vision towers than visually self-supervised encoders, particularly for text-rich images and charts (Karamcheti et al., 2024; Tong et al., 2024), even though text-supervised encoders tend to be weaker on dense prediction (Siméoni et al., 2026). Our goal is therefore to make a self-distilled visual encoder more useful to a VLM while retaining its strengths in recognition and dense prediction.

Two relevant lines of work approach this problem differently. One augments paired image–text pretraining with self-distillation or spatial objectives (Naeem et al., 2024; Maninis et al., 2025; Tschannen et al., 2025); the other transfers pretrained teachers through feature or similarity distillation (Fang et al., 2023; Sun et al., 2023; Ranzinger et al., 2024). The former retains an image–text

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

objective during visual training, while the latter trains a student to reproduce a selected teacher's representation or cross-modal similarity structure. We take a different route: similarities from a frozen contrastive image tower route images over a fixed semantic vocabulary inside DINO's self-distillation objective. This requires neither image labels nor captions paired with the training images, and does not train the backbone to match teacher features.

In `LexDINO`, we use the semantic structure of an LLM to re-define the prototype space through which DINO is originally designed to learn, prescribing to its prototypes semantic meaning. We cluster a separate caption corpus in the LLM's feature space and use the resulting centroids as a fixed prototype bank. This provides us a semantic vocabulary, but does not determine how images should use its entries. Cross-view agreement is unchanged if visual concepts are consistently associated with the wrong rows.

We resolve this ambiguity through a shared caption-group index (Fig. 1). We encode the same captions with a frozen text tower from an image-text contrastive model and average them within the LLM-defined groups, rather than clustering them again. Row j in the LLM bank and row j in the contrastive bank therefore represent the same captions, although the two banks occupy different feature spaces. The frozen image tower from the image-text model serves as a semantic router: it scores each image against the second bank, and the resulting affinities supervise the logits for the corresponding rows of `LexDINO`'s prototype bank. The shared index thus connects the two spaces without a learned map between them or regression to the router's feature coordinates.

Results: We first evaluate `LexDINO` in a controlled setting (dataset (ImageNet-1k), backbone (ViT-L/16), and optimisation fixed). Relative to a DINOv3 trained in the same way, `LexDINO` improves linear-probe accuracy by 2.2 points, k-NN accuracy by 2.0 points, and ADE20K mIoU by 3.3 points (Table 1). When used as a VLM's vision encoder (under the same training recipe), it improves all twelve benchmarks, including a 25.4-point average gain on OCR tasks (Table 2). Replacing the LLM-derived prototypes with SigLIP2 text centroids of the same caption groups lowers the OCR average by 6.7 points. All three LLM-derived banks lie within 1.2 points of each other yet at least 5.5 points above this variant, which is the lowest on every OCR task (Table 12), supporting the use of LLM representations for the prototype bank.

We next **scale** `LexDINO` to a roughly 1B parameter ViT-g/16 which is subsequently distilled into ViT-L/16 and ViT-B/16 students. Our full teacher–student pipeline uses a fraction of DINOv3's pretraining compute, with a 1B rather than 7B teacher. Among encoders below 2B parameters, our `LexDINO` ViT-g/16 is best or tied on all five dense benchmarks. Both distilled students outperform their size-matched DINOv3 counterparts on the two ImageNet probes and improve multiple dense benchmarks (Table 6). When used as vision encoders of a VLM, our distilled students also improve the OCR and non-OCR task averages (Table 3; Appendix C.4).

Our main **contributions** are:

- We introduce `LexDINO` that replaces DINO's learned prototype bank with fixed centroids of caption groups in an LLM feature space, giving its output vocabulary a language-derived geometry without requiring captions paired with the training images.
- We ground these indices by representing the same caption partition in the space of the text encoder of a frozen contrastive image-text model and routing image-dependent affinities to the corresponding `LexDINO` logits. The shared index requires neither a map between the two text spaces nor regression to the router's features.
- We establish the impact of `LexDINO` in controlled experiments. Then, we scale it to a strong ViT-g/16 encoder, and, finally, compress it into ViT-L/16 and ViT-B/16 students. The resulting family transfers across VLM, recognition, segmentation, and depth tasks.

2 Related Work

Self-supervised visual pretraining: Clustering and prototype assignment underpin many of the strongest self-supervised visual encoders. DeepCluster and SwAV established this direction which remains central to more recent methods (Caron et al., 2018; 2020; Assran et al., 2022; Darcet et al., 2025; Venkataramanan et al., 2026). DINO learns image-level invariance by matching a student to a momentum teacher across augmented views (Caron et al., 2021). iBOT brings categorical self-

distillation to masked patch tokens (Zhou et al., 2022). DINOv2 and DINOv3 scale this combination through stronger training recipes (Oquab et al., 2024; Siméoni et al., 2026). Across these methods, visual prototypes emerge from image training and carry no predetermined semantics. To this end, LexDINO gives DINO's image-level prototype bank the geometry of an LLM-derived vocabulary and grounds its entries through image-dependent alignment.

Language supervision and multimodal transfer: Models such as CLIP, ALIGN, and SigLIP gain broad semantic coverage from paired image-text pretraining, but their global alignment objectives give weak supervision for local structure (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023). Recent work adds visual learning signals without leaving this paired setting. SILC introduces local-to-global EMA consistency. TIPS combines web and synthetic captions with local-crop consistency and masked-patch prediction, while SigLIP 2 adds captioning, self-distillation, and masked prediction (Naeem et al., 2024; Maninis et al., 2025; Tschannen et al., 2025). In all three, the visual losses augment a contrastive image-text objective, with paired captions still part of the training loop.

In LexDINO, we take a different path. We begin with DINO/iBOT and use LLM-derived semantics to structure its native prediction space. First, we cluster an independent caption corpus in LLM space to create semantic prototypes. Then, we represent the prototypes in a contrastive text space using the text encoder of a frozen image-text model, and finally use the corresponding frozen image encoder to map training images onto them. Once these banks are built, visual pretraining itself uses neither captions nor a text encoder. Visual self-distillation therefore remains the training framework rather than an auxiliary component of image-text pretraining.

Transferring pretrained semantic structure: Feature-level teacher transfer spans both vision-language and vision-only settings. EVA predicts masked frozen-CLIP visual features. AM-RADIO uses teacher-specific heads to match global and spatial outputs from heterogeneous teachers including CLIP. Theia uses teacher-specific translators to distill spatial outputs only, and UNIC distills complementary supervised and self-supervised visual teachers (Fang et al., 2023; Ranzinger et al., 2024; Sarıyıldız et al., 2024; Shang et al., 2025). These methods transfer teacher feature coordinates through learned projectors. DIME-FM contrastively distills CLIP's cross-modal similarity function by matching a teacher's row- and column-wise distributions over a similarity matrix formed from unpaired images and sentences (Sun et al., 2023). In contrast, LexDINO trains the backbone to reproduce neither the teacher's features nor its full similarity matrix: only the tower's similarity distribution over each image's top-ranked caption groups supervises DINO's own logits over a fixed LLM-derived vocabulary. DINO's cross-view matching and iBOT masked-patch prediction remain the representation-learning objectives, so external semantics determine what DINO's assignments mean without regressing backbone features. Distilling the same tower's features instead is weaker for VLM transfer, especially on OCR (Table 11).

3 METHOD

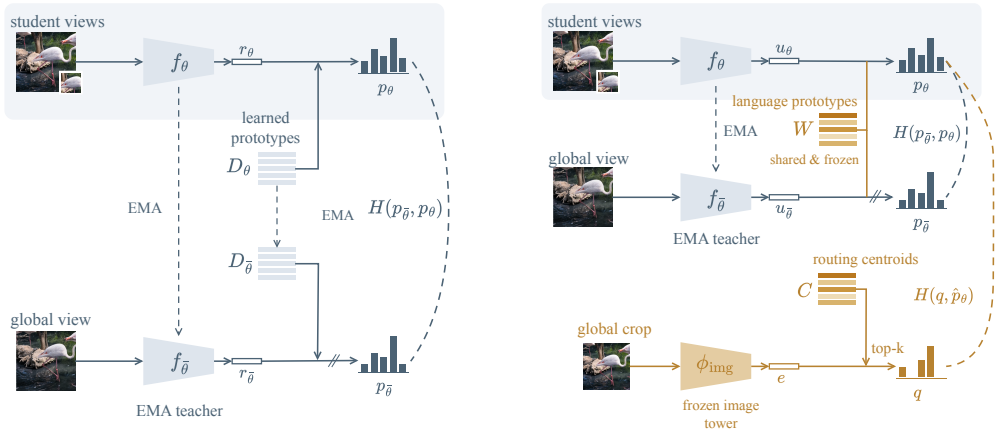
LexDINO changes both the vocabulary and the supervision of DINO's image-level prediction head. DINO predicts over learned prototypes whose indices have no externally specified meaning. We replace the student bank and its EMA copy with a shared, fixed vocabulary of caption-group centroids constructed in an LLM feature space. The same caption groups are pooled in the joint space of a frozen image-text model to produce the routing bank $\boldsymbol{C}$. Its image tower supplies sparse cross-space alignment targets over the corresponding group indices, while the EMA teacher supplies full-vocabulary targets across views. Both signals supervise the same logits over $\boldsymbol{W}$. iBOT retains a separate learned patch head (Fig. 1).

3.1 PRELIMINARIES

Let f_θ denote the student and $f_{\bar\theta}$ its EMA teacher, with $\bar\theta \leftarrow m\bar\theta + (1-m)\theta$. For each image, we sample two global views $\mathcal{V}_{\text{glob}}$ and eight local views $\mathcal{V}_{\text{loc}}$, and write $\mathcal{V} = \mathcal{V}_{\text{glob}} \cup \mathcal{V}_{\text{loc}}$. The student processes every view, whereas the teacher processes only the global views.

Image-level self-distillation (DINO): For $a \in \{\theta, \bar\theta\}$, let $\boldsymbol{r}_a^{\text{D}}(\boldsymbol{v}) \in \mathbb{R}^{d_{\text{D}}}$ denote the output of the image-level projection head before its final layer. DINO parameterises this layer as a learned prototype matrix $\boldsymbol{D}_a \in \mathbb{R}^{K \times d_{\text{D}}}$, giving logits

$$\boldsymbol{z}_a^{\text{D}}(\boldsymbol{v}) = \boldsymbol{D}_a \boldsymbol{r}_a^{\text{D}}(\boldsymbol{v}), \qquad a \in \{\theta, \bar\theta\}.$$



(a) DINOv3 (b) LexDINO

Figure 1: **Language-grounded image-level self-distillation.** (a) DINOv3 learns a student prototype bank and its EMA copy. (b) LexDINO replaces both with a shared, fixed LLM-derived bank $\boldsymbol{W}$. The EMA teacher provides full-vocabulary cross-view targets, while the frozen image tower and routing bank $\boldsymbol{C}$ provide sparse alignment targets over the corresponding group indices. Both supervise the same image-level logits. Separate learned iBOT heads are omitted.

The student bank $\boldsymbol{D}_\theta$ is learned by gradient descent, while $\boldsymbol{D}_{\bar\theta}$ follows it through the EMA update. The student prediction and teacher target are then

$$\boldsymbol{p}^{\mathrm{D}}_\theta(\boldsymbol{v}) = \mathrm{softmax}\left(\boldsymbol{z}^{\mathrm{D}}_\theta(\boldsymbol{v})/\tau_{\mathrm{s}}\right), \qquad \boldsymbol{p}^{\mathrm{D}}_{\bar\theta}(\boldsymbol{v}) = \mathrm{SK}\left(\boldsymbol{z}^{\mathrm{D}}_{\bar\theta}(\boldsymbol{v})/\tau_{\mathrm{t}}\right).$$

τ_{s} and τ_{t} are the student and teacher temperatures. SK denotes Sinkhorn-Knopp balancing applied jointly to the teacher logits from both global views over all images in the distributed minibatch. For each image, a teacher prediction from a global view $\boldsymbol{v} \in \mathcal{V}_{\mathrm{glob}}$ supervises every distinct student view $\boldsymbol{v}' \in \mathcal{V}$, the other global view and all local views. For one such pair:

$$\ell_{\mathrm{view}}(\boldsymbol{v}, \boldsymbol{v}') = -\boldsymbol{p}^{\mathrm{D}}_{\bar\theta}(\boldsymbol{v})^\top \log \boldsymbol{p}^{\mathrm{D}}_\theta(\boldsymbol{v}'). \tag{1}$$

$\mathcal{L}_{\mathrm{view}}$ averages these terms over images and all eligible view pairs. With the learned banks above, this is the standard DINO image-level objective. Gradients do not pass through the teacher branch.

Patch-level distillation (iBOT): iBOT uses a separate learned head for masked patch prediction. For selected global views, the student receives masked patch tokens while the teacher processes the corresponding unmasked view. The loss matches their patch-level distributions at the masked positions, averaging first within each view and then across global views and images. Local views do not enter this objective. We denote the resulting loss by $\mathcal{L}_{\mathrm{iBOT}}$.

3.2 Language-derived prototype bank

LexDINO replaces DINO's learned student bank and its EMA copy, $(\boldsymbol{D}_\theta, \boldsymbol{D}_{\bar\theta})$, with a single fixed bank $\boldsymbol{W}$ shared by both branches. Let $\{t_i\}_{i=1}^N$ be an independent corpus of captions. We embed these captions in the target LLM's feature space and cluster them into K groups (prototypes), whose centroids form $\boldsymbol{W}$. Section 3.3 reuses the same group assignments to construct the routing bank $\boldsymbol{C}$, so corresponding rows of the two banks represent the same prototype.

LexDINO prototypes: Since LLMs are not trained as sentence encoders, we follow recent work showing that prompting an LLM for a one-word summary yields useful text embeddings (Jiang et al., 2024a;b; Ouali et al., 2025). For each caption t_i, we append the instruction "Summarize the following sentence in one word:" and use the penultimate-layer hidden state at the final prompt position, $\boldsymbol{h}_i \in \mathbb{R}^d$, as its representation. Moreover, LLM hidden states are often anisotropic, with a common direction shared across many inputs (Ethayarajh, 2019). Left untreated, this component dominates the cosine geometry used for clustering. Our pre-processing combines centering, removing the leading principal direction, ZCA whitening, and unit-length normalization, so that clustering depends on direction rather than the original feature magnitudes (Mu & Viswanath, 2018).

After this pre-processing, we apply k-means to the caption embeddings. Let $\pi(i) \in \{1, \dots, K\}$ denote the cluster assigned to caption i, and let $\boldsymbol{w}_j$ denote the unit-normalised centroid of cluster j. With K=65,536, these centroids form LexDINO's prototype bank $\boldsymbol{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_K]^\top \in \mathbb{R}^{K \times d}$.

Generally, the choice of LLM is flexible. In our experiments, we primarily use Qwen2.5-3B as the target LLM (Yang et al., 2024). The bank is constructed from approx. $8M$ captions drawn from CC12M and LAION-COCO (Changpinyo et al., 2021; Schuhmann et al., 2022).

Image-level prediction over language prototypes: Once the bank has been constructed, it becomes the vocabulary of LexDINO's full-bank cross-view term. Each row $\boldsymbol{w}_j$ replaces a learned weight vector in the model's final layer and remains fixed throughout training. Both the student and EMA teacher score their views using the same $\boldsymbol{W}$. For $a \in \{\theta, \bar{\theta}\}$, the corresponding projection head maps the CLS token to the LLM dimension d and normalises the resulting feature:

$$\boldsymbol{u}_a(\boldsymbol{v}) = \Pi\left(\mathrm{MLP}_a(f_a(\boldsymbol{v})_{\mathrm{cls}})\right), \qquad \boldsymbol{z}_a(\boldsymbol{v}) = \boldsymbol{W}\boldsymbol{u}_a(\boldsymbol{v}), \qquad a \in \{\theta, \bar{\theta}\}. \tag{2}$$

Here $\Pi(\boldsymbol{x}) = \boldsymbol{x}/\|\boldsymbol{x}\|_2$, and $f_a(\boldsymbol{v})_{\mathrm{cls}}$ denotes the CLS token. Since both $\boldsymbol{u}_a(\boldsymbol{v})$ and the rows of $\boldsymbol{W}$ have unit norm, $z_{a,j}(\boldsymbol{v}) = \boldsymbol{w}_j^\top \boldsymbol{u}_a(\boldsymbol{v})$ is a cosine-similarity logit. The student prediction and teacher target are

$$\boldsymbol{p}_\theta^{\mathrm{L}}(\boldsymbol{v}) = \mathrm{softmax}\left(\boldsymbol{z}_\theta(\boldsymbol{v})/\tau_{\mathrm{s}}\right), \qquad \boldsymbol{p}_{\bar{\theta}}^{\mathrm{L}}(\boldsymbol{v}) = \mathrm{SK}\left(\boldsymbol{z}_{\bar{\theta}}(\boldsymbol{v})/\tau_{\mathrm{t}}\right).$$

Replacing $\boldsymbol{p}_a^{\mathrm{D}}$ in Eq. 1 with these distributions and averaging over the eligible view pairs defines the LexDINO loss, $\mathcal{L}_{\mathrm{LexDINO}} \equiv \mathcal{L}_{\mathrm{view}}^{\boldsymbol{W}}$.

Sharing $\boldsymbol{W}$ gives every output index a stable semantic identity. Cross-view matching alone, however, only requires different views of an image to select the same indices. It can still associate a visual concept consistently with caption groups that do not describe it. Section 3.3 grounds this correspondence in image content.

3.3 Visual grounding through a shared semantic index

To establish this connection, we construct a companion bank in the joint space of a frozen contrastive model. Its rows preserve the caption groups used to construct $\boldsymbol{W}$, so the shared index, rather than a learned transformation, connects the two feature spaces.

A routing bank in the contrastive space: We use the frozen text tower ϕ_{txt} to obtain a normalized feature $\boldsymbol{b}_i = \Pi(\phi_{\mathrm{txt}}(t_i))$ for each caption. Rather than clustering these features again, we preserve the LLM assignments $\pi(i)$ and pool the contrastive features within the same groups:

$$\boldsymbol{c}_j = \Pi\left(\sum_{i:\pi(i)=j} \boldsymbol{b}_i\right), \qquad \boldsymbol{C} = [\boldsymbol{c}_1, \dots, \boldsymbol{c}_K]^\top \in \mathbb{R}^{K \times d_c}. \tag{3}$$

Each caption group now has two representations: $\boldsymbol{w}_j$ in LLM space and $\boldsymbol{c}_j$ in contrastive space. Because both rows summarize the same captions, row j denotes the same group in both banks. The frozen image tower can therefore score an image against $\boldsymbol{C}$, and those scores can supervise the corresponding logits produced through $\boldsymbol{W}$. This transfers evidence through the shared group index without regressing features or learning a mapping between the two spaces. For a balance of speed and expressivity, we use SigLIP2 ViT-L/16 as the frozen contrastive model (Tschannen et al., 2025).

Cross-space alignment: For each global student view $\boldsymbol{v}$, let $\tilde{\boldsymbol{v}}$ denote the associated input to the frozen image tower ϕ_{img}. We compare its normalized feature with the rows of $\boldsymbol{C}$ to measure how strongly the image matches each caption group. We retain the k highest-scoring groups and normalize their scores to form a sparse target:

$$\boldsymbol{e}(\tilde{\boldsymbol{v}}) = \Pi\left(\phi_{\mathrm{img}}(\tilde{\boldsymbol{v}})\right), \qquad \boldsymbol{s}(\tilde{\boldsymbol{v}}) = \boldsymbol{C}\,\boldsymbol{e}(\tilde{\boldsymbol{v}}), \qquad \mathcal{I}(\tilde{\boldsymbol{v}}) = \text{top-}k\left(\boldsymbol{s}(\tilde{\boldsymbol{v}}), k\right),$$

$$q_j(\tilde{\boldsymbol{v}}) = \frac{\exp(s_j(\tilde{\boldsymbol{v}})/\tau_{\mathrm{a}})}{\sum_{\ell \in \mathcal{I}(\tilde{\boldsymbol{v}})} \exp(s_\ell(\tilde{\boldsymbol{v}})/\tau_{\mathrm{a}})}, \qquad j \in \mathcal{I}(\tilde{\boldsymbol{v}}). \tag{4}$$

where k=64 and τ_{a}=0.04. Since row j refers to the same caption group in both banks, the indices in $\mathcal{I}(\tilde{\boldsymbol{v}})$ also select the corresponding logits from the language-prototype head. Normalising the student logits over this shared support gives:

$$\hat{p}_{\theta,j}\left(\boldsymbol{v} \mid \mathcal{I}(\tilde{\boldsymbol{v}})\right) = \frac{\exp(z_{\theta,j}(\boldsymbol{v})/\tau_{\mathrm{s}})}{\sum_{\ell \in \mathcal{I}(\tilde{\boldsymbol{v}})} \exp(z_{\theta,\ell}(\boldsymbol{v})/\tau_{\mathrm{s}})}, \qquad j \in \mathcal{I}(\tilde{\boldsymbol{v}}), \tag{5}$$

where τ_s=0.1. Although the target and student distributions are computed in different feature spaces, their entries refer to the same semantic groups. We therefore use cross-entropy to align the student's relative preferences over the selected groups with those of the contrastive model. With B denoting the minibatch size and $\mathcal{V}_{\text{glob}}^{(n)}$ the global views of image n:

$$\mathcal{L}_{\text{align}} = -\frac{1}{B|\mathcal{V}_{\text{glob}}|}\sum_{n=1}^{B}\sum_{\boldsymbol{v}\in\mathcal{V}_{\text{glob}}^{(n)}}\sum_{j\in\mathcal{I}(\tilde{\boldsymbol{v}})} q_j(\tilde{\boldsymbol{v}})\log\hat{p}_{\theta,j}\left(\boldsymbol{v}\mid\mathcal{I}(\tilde{\boldsymbol{v}})\right). \tag{6}$$

Because both distributions are normalized on the selected support, this loss directly supervises only their relative affinities. The remaining $K-k$ logits are not directly supervised by alignment and continue to be trained by $\mathcal{L}_{\text{LexDINO}}$. The two terms therefore act on the same logits at complementary scopes: cross-space alignment grounds the image-relevant groups, while the `LexDINO` loss maintains consistency over the complete vocabulary. The support and target are detached, so $\mathcal{L}_{\text{align}}$ updates only the student backbone and its image-level projection head.

3.4 Training objective

The complete pretraining objective combines the `LexDINO` loss with the separate cross-space alignment term, alongside iBOT masked-patch prediction and the KoLeo feature-spreading regularizer:

$$\mathcal{L} = \mathcal{L}_{\text{LexDINO}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{ibot}}\mathcal{L}_{\text{iBOT}} + \lambda_{\text{koleo}}\mathcal{L}_{\text{KoLeo}}. \tag{7}$$

We set λ_{align}=1. $\mathcal{L}_{\text{KoLeo}}$ is applied to the student CLS features. Following DINOv3, we use λ_{koleo}=0.1. Once the two banks are constructed, neither the caption corpus nor the text encoder is required.

3.5 Scaling to a model family

We train a roughly 1B parameter `LexDINO` ViT-g/16 with the same language-grounded objective on a broader image mixture and then use the frozen model to distil ViT-L/16 and ViT-B/16 students.

Large-scale teacher pretraining: Following DINOv2, the training mixture combines ImageNet-1k (Deng et al., 2009), ImageNet-21k (Ridnik et al., 2021), Google Landmarks v2 (Weyand et al., 2020), and Mapillary Street-Level Sequences (Warburg et al., 2020), to which we add roughly 300M images sampled at random from DataComp (Gadre et al., 2023) to further broaden the visual distribution. Our dataset construction is simpler than DINOv2's and DINOv3's. DINOv2 constructs LVD-142M by retrieving images similar to curated seed datasets, starting from a web corpus reported as roughly 1.2B unique images. DINOv3 forms LVD-1689M through hierarchical clustering and balanced sampling from a roughly 17B-image pool, then mixes it with a separately retrieved component and public datasets (Oquab et al., 2024; Siméoni et al., 2026).

We train a `LexDINO` ViT-g/16 with the objective in Eq. 7. Although our mixture contains more images than LVD-142M, the shorter schedule exposes the teacher to fewer sampled images than either DINOv2 or DINOv3 and uses less than half as many optimisation steps as DINOv3.

Fixed-teacher family distillation: We follow DINOv3's distillation procedure with two changes (Siméoni et al., 2026). First, we replace its batch-balanced SK teacher distributions with sparse targets normalized independently for each token. This prevents SK's undesired effect where the same image can receive different targets when batched with different examples, and the balancing constraint can distort assignments when a batch is semantically skewed. Our sparse softmax instead normalizes each token independently, making its target invariant to batch composition while preserving the fixed teacher's strongest prototype affinities. Second, we apply patch-level supervision over the full spatial grid rather than at a random subset utilizing every available teacher prediction. For each CLS or patch token, we retain the indices of the teacher's 2,048 largest logits and apply the softmax over this support. Each student is trained for 350k iterations, substantially shorter than the 1M iterations and 250k learning-rate cooldown used by DINOv3.

Together with the smaller model (1B versus 7B parameters), our teacher requires about 5% of DINOv3's estimated pretraining compute, and about 13% even when the pretraining of the frozen router is included (Appendix B).

Table 1: **Controlled ImageNet-1k comparison.** ViT-L/16 models are trained from scratch under matched settings and evaluated frozen. "+ fixed bank" omits alignment. Mean of three runs.

	ImageNet-1k		Segmentation (mIoU ↑)			Depth (RMSE ↓)	
Model	**Linear ↑**	**k-NN ↑**	**ADE20K**	**Cityscapes**	**VOC**	**KITTI**	**NYUv2**
DINOv3	83.1	81.9	44.0	71.2	82.5	3.213	0.449
+ fixed bank	82.9	82.1	44.7	71.5	82.7	3.212	**0.437**
LexDINO (+ alignment)	**85.3**	**83.9**	**47.3**	**72.8**	**83.3**	**3.211**	**0.437**

Table 2: **VLM transfer after ImageNet-1k pretraining.** All ViT-L/16 encoders use the same downstream VLM recipe. Group averages are unweighted, with MME-P rescaled to 0–100.

	Non-OCR								OCR					
Encoder	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
DINOv3	62.2	58.8	63.6	37.7	87.2	53.7	72.9	62.3	17.8	41.0	17.7	23.5	6.2	21.3
+ fixed bank	61.1	58.9	64.2	38.1	87.2	53.9	73.8	62.5	19.1	41.6	18.9	24.1	9.0	22.5
LexDINO	**66.3**	**59.2**	**65.8**	**43.1**	**88.2**	**56.5**	**75.0**	**64.9**	**41.6**	**65.3**	**59.1**	**32.7**	**34.5**	**46.7**

4 RESULTS

Our evaluation follows the two goals that motivate LexDINO. First, we assess the frozen encoder on image classification, semantic segmentation, and depth estimation. Second, we use it as the vision tower of a generative VLM and evaluate VQA, reasoning, OCR, etc. Controlled ablations appear in Appendix C. Appendix E examines the caption banks and the stability of prototype assignments during training.

4.1 EXPERIMENTAL SETUP

Pretraining settings: For the controlled experiments, we train LexDINO ViT-L/16 encoders on ImageNet-1k (Deng et al., 2009) following the DINOv3 training recipe. At scale, we use the data mixture from Section 3.5 to train a LexDINO ViT-g/16 teacher and distill ViT-L/16 and ViT-B/16 students. Appendix A provides the complete training details.

Frozen visual evaluation: We evaluate the frozen encoders on global and dense tasks following DINOv3 (Siméoni et al., 2026). Global recognition is measured by linear probing and k-NN classification on ImageNet-1k. For dense prediction, we fit a linear decoder to patch tokens and report RMSE on NYUv2 (Silberman et al., 2012) and KITTI (Geiger et al., 2012), and mIoU on ADE20K (Zhou et al., 2017), Cityscapes (Cordts et al., 2016), and Pascal VOC (Everingham et al., 2010).

VLM evaluation: We use each pretrained encoder as the vision tower of the same LLaVA-OneVision-style VLM (Li et al., 2025), built on Qwen2.5-1.5B (Yang et al., 2024), while keeping the downstream data and training settings fixed. Appendix A provides the complete recipe. We evaluate the resulting models with LMMs-Eval (Zhang et al., 2025a) on 12 benchmarks spanning VQA, reasoning, OCR, chart interpretation, and document understanding.

4.2 CONTROLLED EXPERIMENTS ON IMAGENET-1K

We first train ViT-L/16 models from scratch on ImageNet-1k. Table 1 shows that replacing DINOv3's prototypes with LexDINO's prototypes gives 82.1 k-NN accuracy and 44.7 ADE20K mIoU, essentially matching the DINOv3 baseline. Further adding the alignment loss of Section 3.3, our full method (LexDINO) reaches 83.9 k-NN accuracy and 47.3 ADE20K mIoU. It also improves linear accuracy, Cityscapes, and Pascal VOC over the baseline while retaining its depth performance.

When used as VLM vision encoder, LexDINO improves on the DINOv3 baseline across all 12 benchmarks (Table 2). The non-OCR average rises from 62.3 to 64.9. On OCR, it rises from 21.3 to 46.7. Our encoder is a markedly stronger VLM vision tower without giving up the recognition and dense prediction strengths of visual self-supervision.

Table 3: **VLM transfer after large-scale pretraining.** All encoders use the same downstream VLM protocol and a 24×24 patch-token grid. Our ViT-L/16 is distilled from the ViT-g/16 teacher. Benchmark groups and MME-P scaling follow Table 2.

	Non-OCR								OCR					
Encoder	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
SigLIP2 L/16	**75.0**	**62.8**	71.9	**50.7**	88.4	**63.9**	**85.6**	**71.2**	**75.4**	75.4	86.2	**59.9**	**72.4**	**73.8**
AM-RADIOv2.5 L/16	72.3	62.0	69.7	48.0	87.5	60.3	83.2	69.0	67.7	75.0	85.3	52.2	68.8	69.8
PEspatial L/14	73.2	62.2	68.6	48.6	87.8	62.5	84.1	69.6	70.9	**77.5**	86.4	56.9	69.0	72.1
DINOv2 L/14	72.8	62.4	66.9	49.8	88.8	61.2	79.7	68.8	58.1	73.7	77.0	43.7	56.6	61.8
DINOv3 L/16	73.5	62.6	69.2	49.3	89.0	60.4	82.0	69.4	66.8	75.5	85.5	56.3	64.2	69.7
Web-DINO L/14	72.0	61.0	68.9	47.6	88.2	59.6	83.2	68.6	60.3	74.6	78.6	44.7	58.7	63.4
V-JEPA 2.1 L/16	70.3	61.3	67.9	44.9	88.0	57.5	79.4	67.0	49.7	70.0	69.3	37.6	45.1	54.4
LexDINO L/16	74.7	62.6	**72.0**	50.1	**89.5**	62.8	84.6	70.9	72.0	76.6	**86.6**	58.2	69.1	72.5

4.3 Training and Evaluation at Scale

VLM evaluation: We compare LexDINO ViT-L/16 (distilled from our ViT-g/16) to ViT-L encoders spanning visual self-supervision, image-text pretraining, and multi-teacher distillation (Oquab et al., 2024; Siméoni et al., 2026; Fan et al., 2025; Heinrich et al., 2025; Bolya et al., 2025; Mur-Labadia et al., 2026; Tschannen et al., 2025) when used as the vision tower of the same VLM.

LexDINO outperforms DINOv3 for both general and OCR-heavy tasks (Table 3). The non-OCR average rises from 69.4 to 70.9 while the OCR-heavy ones from 69.7 to 72.5. It also nearly matches SigLIP2 L/16, the model family of its frozen router, while retaining far stronger dense features (e.g., 54.5 vs 43.0 ADE20K mIoU; Table 6). The remaining gap narrows further with Qwen3-1.7B (Appendix C.3).

Table 4: **ImageNet-1k top-1 from frozen features.** All models are re-evaluated with the same protocol.

Model	**Linear**	**k-NN**
DINOv2 g/14	87.0	83.7
DINOv3 7B/16	87.5	85.7
Franca G/14	84.5	81.1
SigLIP2 So400m	87.3	84.7
V-JEPA 2.1 G/16	85.5	–
LexDINO g/16	**87.6**	**85.8**

Global and dense visual transfer: We first evaluate the large-scale ViT-g/16 teacher and its distilled ViT-L/16 and ViT-B/16 students on ImageNet-1k recognition, semantic segmentation, and depth estimation. The large-scale LexDINO encoder reaches 87.6 linear-probe accuracy and 85.8 k-NN accuracy on ImageNet-1k (Table 4). Under the same protocol, both results are on par with the 7B DINOv3, a model seven times larger.

For dense tasks, Table 5 compares our LexDINO ViT-g/16 encoder with models spanning different pretraining objectives and scales, alongside their dataset sizes and reported samples seen. Among

Table 5: **Dense transfer from frozen features.** Depth is evaluated by RMSE and segmentation by mIoU. ADE20K and VOC use matched patch-token grids: 448×448 inputs for patch-size-14 models and 512×512 for patch-size-16 models. * marks estimated or reconstructed counts.

Method	**Param.**	**Samples seen**	**Dataset size**		**Depth**		**Segmentation**		
			Images	**Videos**	**NYUv2 ↓**	**KITTI ↓**	**ADE20K**	**Citysc.**	**VOC**
Models with at least 2B parameters									
Web-DINO 7B/14	7B/14	8.0B	2B	–	0.466	3.158	42.7	68.3	76.1
DINOv3 ViT-7B/16	7B/16	∼4.9B	1.7B	–	0.309	**2.346**	**55.9**	**81.1**	**86.6**
V-JEPA 2.1 ViT-G/16	2B/16	∼358M	142M	21M*	**0.307**	2.461	47.9	73.5	85.0
PEcore 2B/14	2B/14	86.1B	5.4B	22M	0.590	4.119	38.9	61.1	69.2
PEspatial 2B/14	2B/14	86.4B	5.4B	22M*	0.362	3.082	49.3	73.2	82.7
Models less than 2B parameters									
AM-RADIOv2.5 1B/14	1B/14	∼0.9B	1.4B*	–	0.340	2.918	53.0	78.4	85.4
InternVideo2-1B/14	1B/14	∼330M	∼15M*	∼11M*	0.471	4.739	28.4	35.6	65.2
SigLIP 2 1B/16	1B/16	40B	10B	–	0.494	3.273	42.7	64.8	72.7
DINOv2 ViT-g/14	1B/14	∼1.92B	142M	–	0.372	2.624	49.5	75.6	83.1
DINOv3 ViT-H+/16	0.8B/16	2.4–5.1B	1.7B	–	0.352	2.635	**54.8**	79.5	85.8
V-JEPA2 ViT-g/16	1B/16	∼774M	1M	21M*	0.642	4.650	24.4	45.9	63.9
V-JEPA 2.1 ViT-g/16	1B/16	∼358M	142M*	21M*	0.350	2.601	47.8	71.8	84.7
LexDINO ViT-g/16	1B/16	1.4B	0.32B	–	**0.306**	**2.520**	**54.8**	**79.6**	**86.9**



Figure 2: **Patch features at native resolution.** Input images (top) and PCA maps of LexDINO ViT-g/16 (middle) and DINOv3 ViT-H+/16 (bottom), from non-overlapping 256×256 tiles.

Table 6: **Global and dense transfer at ViT-L and ViT-B scale.** Frozen-feature ImageNet-1k top-1 accuracy, depth RMSE, and segmentation mIoU; patch-token grids are matched across patch sizes.

Size	Model	IN1k linear	IN1k k-NN	NYU ↓	KITTI ↓	ADE	VOC	City
L	DINOv2	86.4	83.8	0.411	3.243	48.6	83.5	74.5
	DINOv3	87.3	85.3	0.351	2.643	**55.0**	86.5	**79.3**
	SigLIP2	85.7	82.9	0.516	3.641	43.0	75.6	62.5
	Franca	82.5	71.7	0.425	3.613	45.9	84.4	72.4
	V-JEPA 2.1	–	–	0.346	**2.617**	46.3	84.8	71.6
	LexDINO	**88.0**	**85.5**	**0.325**	2.717	54.5	**86.7**	78.7
B	DINOv2	84.3	82.2	0.429	3.416	48.3	83.8	74.2
	DINOv3	84.8	83.2	0.371	**2.826**	51.7	**85.9**	**77.3**
	SigLIP2	82.5	79.5	0.539	3.847	40.5	73.5	60.1
	Franca	78.5	63.5	0.462	3.790	42.7	82.8	71.7
	V-JEPA 2.1	–	–	0.410	2.883	41.4	78.0	65.0
	LexDINO	**85.3**	**83.4**	**0.370**	2.991	**52.1**	**85.9**	76.7

encoders below 2B parameters, LexDINO leads or ties all five dense benchmarks. Across the full comparison, it also obtains the best NYUv2 and Pascal VOC results. The 7B DINOv3 remains stronger on KITTI and slightly on ADE20K and Cityscapes, but is far more expensive.

Fig. 2 provides a complementary view of the patch representation. Under native resolution, LexDINO retains coherent object and boundary structure, preserving finer detail than DINOv3 with fewer tile-boundary artifacts, consistent with its performance on the dense probes.

Regarding distilled encoders (Table 6), LexDINO students outperform their size-matched DINOv3 counterparts on the two ImageNet probes. The ViT-L student also improves NYUv2 and Pascal VOC while nearly matching the best performing model on the rest. The ViT-B student improves NYUv2 and ADE20K and ties DINOv3 on Pascal VOC, while trailing it on KITTI and Cityscapes.

5 Conclusion

Our controlled ImageNet experiments show that LexDINO outperforms DINOv3 in recognition, dense transfer, and all twelve VLM benchmarks. At scale, the ViT-g/16 teacher leads or ties all five dense benchmarks among encoders below 2B parameters, and its ViT-L/16 and ViT-B/16 students outperform their size-matched DINOv3 counterparts in recognition and VLM transfer. Notably, our full teacher–student pipeline uses a fraction of DINOv3's pretraining compute. LexDINO's language-grounded prototypes can therefore improve the VLM transfer of a self-distilled visual encoder without sacrificing the global and spatial representations that make DINO broadly useful.

AI USE STATEMENT

We used generative AI tools to help draft and revise the manuscript, to find and verify related work and citations, and to check mathematical notation and consistency between the reported results and the text. The authors reviewed all AI-assisted content and take full responsibility for the final paper.

REPRODUCIBILITY STATEMENT

Section 3 specifies the construction of both prototype banks, the alignment target, and the training objective with its loss weights. Appendix A lists the settings for controlled pretraining and VLM fine-tuning (Tables 7 and 8) and describes the large-scale pretraining and distillation schedules, and Appendix B details the compute accounting. Frozen-feature evaluations follow the DINOv3 protocols, and VLM benchmarks are evaluated with LMMs-Eval. The caption corpora, image datasets, and frozen models that the method relies on (Qwen2.5 and SigLIP2) are publicly available.

REFERENCES

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision (ECCV)*, 2022.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Shang-Wen Li, Piotr Dollár, and Christoph Feichtenhofer. Perception Encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latent patches for improved masked image modeling. *Transactions on Machine Learning Research (TMLR)*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65. Association for Computational Linguistics, 2019.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, and Saining Xie. Scaling language-free visual representation learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the limits of masked visual representation learning at scale. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Greg Heinrich, Mike Ranzinger, Hongxu Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIOv2.5: Improved baselines for agglomerative vision foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: The marginal value of training the last weight layer. In *International Conference on Learning Representations (ICLR)*, 2018.

Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Usman Naseem, Chunyu Wang, Qi Dai, Xiyang Dai, Dongdong Chen, Chong Luo, Lili Qiu, and Liang Hu. LLM2CLIP: Powerful language model unlocks richer cross-modality representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(7):5131–5139, 2026.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196. Association for Computational Linguistics, 2024a.

Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-V: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024b.

Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, Oriane Siméoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. DINOv2 meets text: A unified framework for image- and pixel-level vision-language alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *International Conference on Machine Learning (ICML)*, 2024.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, pp. 235–251, 2016.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research (TMLR)*, 2025.

Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations (ICLR)*, 2022.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 292–305, 2023.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, et al. TIPS: Text-image pretraining with spatial awareness. In *International Conference on Learning Representations (ICLR)*, 2025.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, 2022.

Pascal Mettes, Elise van der Pol, and Cees G. M. Snoek. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations (ICLR)*, 2018.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. SILC: Improving vision language pretraining with self-distillation. In *European Conference on Computer Vision (ECCV)*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Yassine Ouali, Adrian Bulat, Alexandros Xenos, Anestis Zaganidis, Ioannis Maniadis Metaxas, Brais Martinez, and Georgios Tzimiropoulos. VladVA: Discriminative fine-tuning of LVLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4101–4111, 2025.

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 (ICML)*, 2021.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative vision foundation model reduce all domains into one. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K pretraining for the masses. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Mert Bülent Sarıyıldız, Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. UNIC: Universal classification models via multi-teacher distillation. In *European Conference on Computer Vision (ECCV)*, 2024.

Christoph Schuhmann, Andreas Köpf, Richard Vencu, Theo Coombes, and Romain Beaumont. LAION-COCO: 600M synthetic captions from LAION2B-en. https://laion.ai/blog/laion-coco/, 2022.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 724–748. PMLR, 2025.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, 2012.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *Transactions on Machine Learning Research (TMLR)*, 2026.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Ximeng Sun, Pengchuan Zhang, Peizhao Zhang, Hardik Shah, Kate Saenko, and Xide Xia. DIME-FM: DIstilling Multimodal and Efficient foundation models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Shashanka Venkataramanan, Valentinos Pariza, Mohammadreza Salehi, Lukas Knobel, Elias Ramzi, Spyros Gidaris, Andrei Bursuc, and Yuki M. Asano. Franca: Nested Matryoshka clustering for scalable visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Xin Wen, Bingchen Zhao, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. What makes CLIP more robust to long-tailed pre-training data? A controlled study for transferable insights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google Landmarks Dataset v2: A large-scale benchmark for instance-level recognition and retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

xAI. Grok-1.5 Vision preview. https://x.ai/news/grok-1.5v, 2024. Introduces the RealWorldQA benchmark.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916. Association for Computational Linguistics, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations (ICLR)*, 2022.

A Implementation details

Controlled ImageNet-1k pretraining: The ImageNet-1k experiments in Section 4.2 are designed to isolate the effect of the prototype bank and the alignment loss without changing the rest of pretraining. We therefore use the public DINOv3 ViT-L/16 ImageNet-1k configuration (Siméoni et al., 2026) for every variant. All models are trained from scratch under the common recipe summarized in Table 7. The DINOv3 baseline learns its image-level prototype layer, while our variants replace that layer with the fixed language bank. The full model additionally includes the cross-space alignment loss.

Table 7: **Controlled ImageNet-1k pretraining.** ViT-L/16 settings shared by all variants.

Parameter	**Setting**		
Updates	125k		
Global batch size	2,048		
Global crops	2×224^2, scale $[0.32, 1.0]$		
Local crops	8×96^2, scale $[0.05, 0.32]$		
Optimiser	AdamW		
Peak learning rate	5.66×10^{-3}		
Warmup iterations	12.5k		
Learning-rate decay	Cosine to 10^{-6}		
Weight decay	$0.04 \rightarrow 0.4$ (cosine)		
Layer decay	0.9		
Patch-embedding multiplier	0.2		
Drop-path rate	0.3		
EMA momentum	$0.992 \rightarrow 1.0$ (cosine)		
Student temperature	0.1		
Teacher temperature	$0.04 \rightarrow 0.07$		
Teacher-temperature warmup	37.5k iterations		
Loss weights (DINO, iBOT, KoLeo)	$(1.0, 1.0, 0.1)$		
Prototype counts (DINO, iBOT)	(65,536, 65,536)		
iBOT masking	p=0.5, ratio $[0.1, 0.5]$		
	DINOv3	**Fixed bank**	**LexDINO**
Prototype bank	Learned	Fixed language	Fixed language
Cross-space alignment	–	–	✓
Alignment $(k, \tau_a, \lambda_{align})$	–	–	$(64, 0.04, 1.0)$

Large-scale pretraining and distillation: We use the data mixture from Section 3.5 to train the LexDINO ViT-g/16 teacher for 300k iterations with AdamW and a global batch size of 4,096. The learning rate is warmed up linearly for 40k iterations and then held fixed. We use a weight decay of 0.04, increase the teacher temperature from 0.04 to 0.07 over the first 40k iterations, and set the EMA momentum to 0.994. Following DINO, each image yields two 256×256 global crops and eight 112×112 local crops. We then apply Gram anchoring for a further 50k iterations at half the learning rate, with an EMA momentum of 0.999 and Gram-teacher swapping every 20k iterations.

The ViT-L/16 and ViT-B/16 students are trained from this fixed teacher for 350k iterations using the distillation losses from Section 3.5. They follow the same optimisation settings, omit Gram anchoring, and decay the learning rate over the final 50k iterations. For comparison, DINOv3 distils its students for 1M iterations followed by a 250k-iteration learning-rate cooldown before high-resolution adaptation (Siméoni et al., 2026). The compute of the complete teacher–student pipeline is estimated in Appendix B.

Vision-language fine-tuning: For the VLM evaluation, we pair each vision encoder with the same Qwen2.5-1.5B language model and a two-layer MLP projector. We follow Stage 1, Stage 1.5, and the single-image phase of Stage 2 from LLaVA-OneVision (Li et al., 2025), omitting only its subsequent multi-image and video continuation. The first stage learns the projector from 558K image–

caption pairs while the two pretrained towers remain frozen. We then unfreeze the full model for visual knowledge learning on 4M examples, followed by instruction tuning on 3.2M single-image conversations. The vision encoder uses one fifth of the learning rate applied to the projector and language model during the latter two stages. We use 384×384 crops for patch-16 encoders and 336×336 crops for patch-14 encoders, so every encoder produces the same 24×24 grid of 576 patches per crop. We also hold the number of crops per image fixed across encoders. Thus, every encoder receives the same patch budget per image at each stage: at most 576, 2,880, and 5,760 patch tokens in Stages 1, 1.5, and 2, respectively. We evaluate the checkpoint obtained at the end of single-image instruction tuning. Table 8 summarizes the recipe. The seven non-OCR benchmarks are AI2D (Kembhavi et al., 2016), GQA (Hudson & Manning, 2019), MME (Fu et al., 2025), MM-Star (Chen et al., 2024), POPE (Li et al., 2023), RealWorldQA (xAI, 2024), and ScienceQA (Lu et al., 2022). The five OCR benchmarks are TextVQA (Singh et al., 2019), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), and OCRBench (Liu et al., 2024). The tables abbreviate RealWorldQA as RWQA and report the perception score of MME (MME-P) and the image subset of ScienceQA (SQA-I).

Table 8: **Vision-language fine-tuning.** Three-stage LLaVA-OneVision curriculum (Li et al., 2025) with Qwen2.5-1.5B and matched per-image patch budgets.

	Stage 1 Alignment	**Stage 1.5 Knowledge**	**Stage 2 Instruction tuning**
Base crop size	384×384 (patch 16); 336×336 (patch 14)		
Patch-token grid per crop	24×24		
Maximum crops per image	1	5	10
Maximum visual tokens	576	2,880	5,760
Training data	Image–caption	Image knowledge	Single-image instruction
Samples	558K	4M	3.2M
Trainable components	Projector	Full model	Full model
Global batch size	512	512	512
Vision learning rate	–	2×10^{-6}	2×10^{-6}
Projector learning rate	10^{-3}	10^{-5}	10^{-5}
Language-model learning rate	–	10^{-5}	10^{-5}
Epochs	1	1	1

B TRAINING-COMPUTE ESTIMATE

We estimate the compute required to train our large-scale `LexDINO` encoder and compare it with DINOv3 under a common architecture-level accounting. Let $F_m(r)$ denote the FLOPs required for one forward pass through encoder m at resolution r. In the self-distillation runs, the student processes two 256×256 global crops and eight 112×112 local crops per image, whereas the EMA teacher processes only the two global crops. Approximating a trainable forward and backward pass by three forward passes gives the per-image cost

$$C_{\mathrm{SSL}}(m) = 3\,[2F_m(256) + 8F_m(112)] + 2F_m(256). \tag{8}$$

The EMA teacher and frozen routing network require only forward passes. In our large-scale run, the ViT-L router is applied online to both 256×256 global crops of every image. During Gram refinement, the frozen Gram teacher additionally processes both global crops at 512×512. Distilled students differ from the self-distillation setting above: each is trained against a fixed teacher t rather than an EMA copy of itself, so its teacher cost is $2F_t(256)$ per image, replacing the final term in Eq. 8 (Siméoni et al., 2026). No router is used during distillation.

We count a multiply-add as two FLOPs and include the transformer's linear, feed-forward, and attention operations. This reproduces DINOv3's reported 256×256 inference costs of 3.549, 0.450, and 0.163 TFLOPs for ViT-7B, ViT-H+, and ViT-L respectively (Siméoni et al., 2026). The derived 112×112 costs are 0.727 TFLOPs for ViT-7B, 0.091 for ViT-H+, and 0.033 for ViT-L. ViT-B costs 0.047 and 0.009 TFLOPs at resolutions 256 and 112, and our ViT-g costs 0.608, 0.123, and

2.593 TFLOPs at resolutions 256, 112, and 512. Tables 9 and 10 apply these values to the reported schedules and batch sizes.

Table 9: **Estimated pretraining compute.** Component and total costs are in ZFLOPs (10^{21} FLOPs). Our total includes router inference during training, but excludes the prior pretraining of the frozen router and LLM and the one-time construction of the prototype bank.

Model	Schedule	Samples	Student forward/backward	Teacher inference	Auxiliary inference	Total cost
DINOv3 ViT-7B/16	1M + 200k Gram	4.9B	190.4	34.9	23.8	**249.1**
LexDINO ViT-g/16	300k + 50k Gram	1.4B	9.5	1.7	1.5	**12.7**

Table 10: **Estimated distillation compute.** Costs are in ZFLOPs. Teacher inference is attributed in full to each student, although DINOv3 shares it across students. DINOv3's ViT-L and ViT-B runs use the batch size of its released ViT-L distillation configuration (1,920), and its high-resolution adaptation is excluded.

Student	Teacher	Schedule	Samples	Student forward/backward	Teacher inference	Total cost
DINOv3 ViT-H+/16†	ViT-7B	1M + 250k cooldown	2.4–5.1B	11.7–25.0	17.0–36.3	**28.8–61.4**
DINOv3 ViT-L/16	ViT-7B	1M + 250k cooldown	2.4B	4.2	17.0	**21.3**
DINOv3 ViT-B/16	ViT-7B	1M + 250k cooldown	2.4B	1.2	17.0	**18.2**
LexDINO ViT-L/16	ViT-g	300k + 50k decay	1.4B	2.5	1.7	**4.3**
LexDINO ViT-B/16	ViT-g	300k + 50k decay	1.4B	0.7	1.7	**2.5**

†The global batch size used to train ViT-H+ is not reported, so we bracket its cost using 1,920 (the released ViT-L distillation configuration) and 4,096 (ViT-7B), equivalent to 2.4B and 5.12B samples over 1.25M updates. Both estimates attribute the full fixed ViT-7B inference cost to one H+ student, although DINOv3 shares it across students, and exclude the compute used to train the ViT-7B teacher.

Within this estimate, our ViT-g run uses 5.1% of the compute of DINOv3 ViT-7B through Gram refinement, or about 20× less, and training it from scratch costs 2.3–4.8× less than DINOv3's distillation of its ViT-H+ student alone. Our students are distilled for 350k iterations at a batch size of 4,096 with the same crops as the teacher run, and cost 5.0× (ViT-L) and 7.3× (ViT-B) less than DINOv3's, mainly because a 1B teacher is cheaper to run than a 7B one. The full teacher–student pipeline therefore totals 19.5 ZFLOPs, 7.8% of DINOv3 ViT-7B's teacher training alone. The estimates omit heads, optimiser and EMA updates, recomputation and data loading, and the pretraining of the LLM. Excluding DINOv3's high-resolution adaptation makes its totals conservative.

The frozen router is an off-the-shelf model whose pretraining our pipeline does not repeat. Counting it nonetheless, with a lower-bound estimate of $3F_{\mathrm{L}}(256)$ per sample over SigLIP2's 40B training samples (image tower only), adds 19.6 ZFLOPs and raises our teacher's total to 32.3 ZFLOPs, still about 7.7× less than DINOv3 ViT-7B, and the full pipeline's to 39.1 ZFLOPs, about 6.4× less.

C Ablation studies

C.1 Comparison with projected feature distillation

Our alignment loss is intended to keep image-to-prototype assignments consistent with the semantic meaning of the prototype bank. Although the language-derived centroids are fixed, cross-view self-distillation can learn consistent assignments without respecting the caption groups those centroids represent. The alignment loss supplies this missing grounding by supervising the image's relative affinities to these groups using the frozen contrastive model.

To assess the benefit of supervising these associations explicitly, we compare LexDINO with direct distillation of the frozen SigLIP2 image encoder's global feature through a learned projector. Both runs retain the Qwen2.5-3B prototype bank and $\mathcal{L}_{\mathrm{LexDINO}}$. In the feature-distillation variant, a cosine feature loss through the projector replaces $\mathcal{L}_{\mathrm{align}}$.

The two models achieve similar ImageNet k-NN accuracy (83.9 versus 83.7), but differ on ADE20K and especially in VLM transfer (Table 11). Cross-space alignment improves ADE20K mIoU from 46.3 to 47.3, while the OCR average rises from 40.4 to 46.7. Across all twelve VLM tasks, the average increases from 54.1 to 57.3. In this controlled comparison, grounding the prototype assignments is more effective for multimodal transfer in particular than matching the frozen encoder's global feature.

Table 11: **Cross-space alignment versus SigLIP2 feature distillation.** Both use the same frozen SigLIP2 router and retain the Qwen2.5-3B prototype bank and $\mathcal{L}_{\text{LexDINO}}$. Feature distillation replaces $\mathcal{L}_{\text{align}}$ with cosine matching of SigLIP2's global feature through a learned projector. Evaluation follows Tables 1 and 2.

Pretraining method	ImageNet k-NN ↑	ADE20K ↑	Non-OCR	OCR	Overall
LexDINO	**83.9**	**47.3**	**64.9**	**46.7**	**57.3**
SigLIP2 feature distillation	83.7	46.3	63.9	40.4	54.1

	Non-OCR								OCR					
Method	AI2D	GQA	MME-P	MMStar	POPE	RWQA	SQA-I	Avg.	TextVQA	ChartQA	DocVQA	InfoVQA	OCRBench	Avg.
LexDINO	**66.3**	**59.2**	65.8	**43.1**	**88.2**	**56.5**	**75.0**	**64.9**	**41.6**	**65.3**	**59.1**	**32.7**	**34.5**	**46.7**
SigLIP2 feature distillation	64.0	**59.2**	**67.2**	41.7	87.9	54.6	72.9	63.9	36.7	59.4	50.2	28.8	27.0	40.4

C.2 Choice of prototype space

We next vary the representation used to construct the fixed prototype bank. Table 12 compares Qwen2.5-1.5B, the reference Qwen2.5-3B, Qwen3-Embedding-4B (Zhang et al., 2025b), and SigLIP2 text features. All variants use the same caption corpus, K=65,536 prototypes, and visual training recipe, including the alignment settings, target-generation procedure, and iBOT configuration. The projection heads have the same depth and hidden widths, with only the output dimension adjusted to the native embedding dimension of each representation.

For the SigLIP2 variant, we retain the Qwen2.5-3B caption assignments and pool SigLIP2 text features within those groups, without reclustering. Centering and whitening are used only to construct the prediction bank $\boldsymbol{W}$. The routing bank $\boldsymbol{C}$ remains as defined in Eq. 3, without these transformations, so image–text similarities are still computed in SigLIP2's original joint space. The banks therefore share caption-group indices, but not identical vectors. Relative to the Qwen2.5-3B reference, this changes the prototype representation while preserving the caption groups, routing bank, and alignment targets.

The benefit of LLM-derived prototypes is clearest in VLM transfer. All three Qwen banks achieve higher OCR and overall averages than the SigLIP2 variant, despite similar ImageNet recognition accuracy. Qwen2.5-3B improves over the SigLIP2 bank on every OCR benchmark, raising the OCR average from 40.0 to 46.7 and the overall average from 54.1 to 57.3. Representing the same caption groups with LLM-derived prototypes therefore improves multimodal transfer over using the grounding model's own text representations, a benefit that ImageNet accuracy alone does not reveal.

The gain is not tied to a single model. A contrastively trained embedding model from a later generation (Qwen3-Embedding-4B) matches the prompted LLMs, and the advantage over DINOv3 persists when the downstream language model (Qwen3-1.7B; Table 14) differs from the one used to build the bank (Qwen2.5-3B). Since the prototype head is discarded after pretraining and the VLM receives only backbone features, the bank shapes the representation during pretraining but is never seen by the language model.

Table 12: **Choice of prototype space.** Controlled ViT-L/16 evaluation on ImageNet-1k, ADE20K, and VLM transfer. SigLIP2 retains the Qwen2.5-3B caption groups and routing targets; VLM groups follow Table 2.

Prototype representation	ImageNet k-NN ↑	ADE20K ↑	Non-OCR avg.	OCR avg.	Overall avg.
Qwen2.5 1.5B	83.7	46.1	64.3	45.5	56.5
Qwen2.5 3B	83.9	**47.3**	**64.9**	**46.7**	**57.3**
Qwen3-Embedding 4B	84.0	46.0	64.5	46.0	56.8
SigLIP2	**84.1**	47.1	64.2	40.0	54.1

	Non-OCR								OCR					
Prototype bank	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
Qwen2.5 1.5B	65.3	59.1	**66.7**	42.2	87.5	55.3	73.9	64.3	40.7	64.1	58.9	30.6	33.3	45.5
Qwen2.5 3B	**66.3**	59.2	65.8	**43.1**	**88.2**	**56.5**	**75.0**	**64.9**	**41.6**	**65.3**	59.1	**32.7**	**34.5**	**46.7**
Qwen3-Embedding 4B	65.7	**61.0**	66.0	41.5	88.1	55.6	73.8	64.5	39.8	64.3	**60.2**	31.8	33.8	46.0
SigLIP2	65.5	59.0	66.0	41.6	87.9	55.4	74.2	64.2	36.1	60.4	49.2	29.6	24.9	40.0

C.3 Transfer across language models

The main experiments use Qwen2.5-1.5B. We repeat the ViT-L/16 comparison with Qwen2.5-3B and Qwen3-1.7B (Yang et al., 2025) to test whether the relative performance of the vision towers depends on the downstream language model.

Qwen2.5-3B. Table 13 compares `LexDINO` with SigLIP2 and the official DINOv3 checkpoint.

Table 13: **VLM transfer with Qwen2.5-3B.** All vision towers are ViT-L/16; reporting follows Table 3.

	Non-OCR								OCR					
Encoder	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
SigLIP2 L/16	**78.8**	63.5	**73.9**	**57.7**	88.7	63.3	87.4	73.3	**75.2**	79.7	**89.2**	**65.4**	**73.5**	**76.6**
DINOv3 L/16	77.4	63.6	69.6	52.7	88.9	61.7	84.8	71.2	65.5	77.8	84.5	55.4	61.5	68.9
`LexDINO` L/16	78.4	**64.5**	73.2	56.7	**89.3**	**65.4**	**89.8**	**73.9**	74.4	**80.2**	88.9	64.6	70.8	75.8

Qwen3-1.7B.

Table 14 compares our ViT-L encoder with SigLIP2 and the official DINOv3 checkpoint under the same twelve-task protocol.

Table 14: **VLM transfer with Qwen3-1.7B.** ViT-L/16 vision towers evaluated with the twelve-task protocol of Table 3.

	Non-OCR								OCR					
Encoder	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
SigLIP2 L/16	74.1	62.1	68.5	49.0	88.5	62.4	**84.6**	69.9	**74.7**	77.3	86.2	58.8	**69.9**	**73.4**
DINOv3 L/16	74.2	62.2	70.6	48.5	**89.3**	63.7	81.8	70.0	67.1	**77.8**	86.7	58.8	64.4	71.0
`LexDINO` L/16	**75.0**	**62.4**	**72.8**	**50.1**	88.9	**65.9**	83.4	**71.2**	71.5	77.6	**87.3**	**60.0**	68.3	72.9

Across both language models, `LexDINO` obtains the highest non-OCR average. On the OCR benchmarks, it remains within 0.8 and 0.5 points of SigLIP2 with Qwen2.5-3B and Qwen3-1.7B, respectively, while outperforming DINOv3 in both settings.

C.4 VLM transfer with ViT-B/16

We also evaluate our distilled ViT-B/16 student as a VLM vision tower. Table 15 compares it with DINOv3 and SigLIP2 at the same backbone size.

Table 15: **VLM transfer with ViT-B/16.** Reporting follows Table 3.

	Non-OCR								OCR					
Encoder	**AI2D**	**GQA**	**MME-P**	**MMStar**	**POPE**	**RWQA**	**SQA-I**	**Avg.**	**TextVQA**	**ChartQA**	**DocVQA**	**InfoVQA**	**OCRBench**	**Avg.**
SigLIP2 B/16	**73.2**	**61.9**	68.3	**49.6**	88.3	58.0	**84.0**	69.0	**69.3**	74.7	84.3	**57.0**	**65.7**	**70.2**
DINOv3 B/16	70.9	60.6	67.5	47.7	87.8	58.6	78.6	67.4	51.5	70.2	72.8	41.3	45.2	56.2
LexDINO B/16	72.9	61.7	**70.0**	48.5	**88.7**	**60.3**	82.8	**69.3**	65.8	**75.2**	**84.5**	53.9	62.9	68.5

Table 16: **Controlled comparison across backbone sizes.** Within each row, DINOv3 and LexDINO are trained from scratch on ImageNet-1k with matched pretraining settings.

Backbone	**DINOv3**		**LexDINO**		**Gain**	
	k-**NN**	**ADE20K**	k-**NN**	**ADE20K**	k-**NN**	**ADE20K**
ViT-S/16	71.8	30.7	78.3	35.7	+6.5	+5.0
ViT-B/16	78.4	39.6	81.9	41.8	+3.5	+2.2
ViT-L/16	81.9	44.0	83.9	47.3	+2.0	+3.3

Compared with DINOv3, LexDINO improves the non-OCR and OCR averages by 1.9 and 12.3 points. It also slightly exceeds SigLIP2 on the non-OCR average, while trailing it by 1.7 points on OCR.

C.5 Controlled comparison across backbone sizes

We repeat the controlled ImageNet-1k comparison of Section 4.2 with ViT-S/16, ViT-B/16, and ViT-L/16 backbones.

LexDINO improves ImageNet k-NN accuracy and ADE20K mIoU at all three sizes (Table 16). Its k-NN advantage over DINOv3 narrows from 6.5 points with ViT-S/16 to 2.0 points with ViT-L/16, whereas the ADE20K improvement remains above two points throughout. The benefit is therefore not specific to ViT-L/16 and is particularly strong for recognition with the smaller backbones.

C.6 Construction of alignment targets

We examine two choices in constructing the alignment targets: whether targets should be cached per image or recomputed for each crop, and whether alignment should use all prototype groups or only the strongest matches from the frozen contrastive model (Table 17). Offline targets are computed once from a canonical view of each image and reused across its global crops, while online targets are recomputed for the crop seen by the student. The top-64 rows isolate the effect of recomputing the target, while the online rows isolate the effect of support size.

Table 17: **Construction of alignment targets.** Offline and online targets with sparse or full-bank support in the controlled ImageNet-1k ViT-L/16 setup.

Target generation	**Support**	k-**NN** ↑	**ADE20K mIoU** ↑
Offline	top 64	83.8	46.5
Online	top 64	**83.9**	**47.3**
Online	all 65,536	83.5	46.0

With top-64 support, recomputing the target for each crop raises ADE20K mIoU from 46.5 to 47.3, while ImageNet k-NN accuracy changes only from 83.8 to 83.9. The effect is therefore concentrated in segmentation. With online target generation fixed, restricting alignment to the 64 strongest matches also outperforms full-bank alignment on both probes. In this controlled comparison, crop-specific sparse targets give the strongest results on both probes.

C.7 Number of prototypes

We next ask whether a larger prototype vocabulary improves the representation. Table 18 compares $K = 65{,}536$ with $K = 131{,}072$ while keeping online per-crop targets and top-64 support fixed. Here, K is the size of the bank, whereas $k = 64$ is the number of groups retained in each alignment target.

Doubling K leaves ImageNet k-NN accuracy essentially unchanged and lowers ADE20K mIoU by 0.4 points. A larger bank therefore offers no benefit on either visual probe in this controlled setting.

Table 18: **Effect of prototype-bank size.** Both runs use online per-crop targets with top-64 support.

Prototypes K	**ImageNet k-NN ↑**	**ADE20K mIoU ↑**
65,536 (reference)	83.9	**47.3**
131,072	**84.0**	46.9

C.8 Sinkhorn normalisation

We compare standard and log-space implementations of Sinkhorn–Knopp in the controlled ImageNet-1k setting (Table 19). The two variants give nearly identical results, with log-space normalisation improving both probes by 0.1 points. Thus, the numerical implementation has little effect on the final representation in this comparison. We nevertheless use the log-space form for large-scale training, where we found it more numerically stable.

Table 19: **Standard versus log-space Sinkhorn–Knopp** in the controlled ImageNet-1k setting.

Normalisation	**ImageNet k-NN ↑**	**ADE20K mIoU ↑**
Standard	83.8	47.2
Log-space	**83.9**	**47.3**

C.9 Caption data for prototype construction

Table 20 compares Qwen2.5-3B prototype banks built from our default CC12M + LAION-COCO mixture and from DataComp captions, using offline alignment targets in both cases. The two banks differ by 0.1 points on ImageNet k-NN and 0.3 mIoU on ADE20K, indicating modest sensitivity to the caption source in this setting.

Table 20: **Caption data for prototype construction.** Qwen2.5-3B banks with offline alignment targets in the controlled ImageNet-1k setting.

Caption corpus	**ImageNet k-NN ↑**	**ADE20K mIoU ↑**
CC12M + LAION-COCO	**83.8**	**46.5**
DataComp	83.7	46.2

D Additional related work

Fixed and language-derived prototypes: Fixed classifier weights and prescribed hyperspherical prototypes have been studied in supervised recognition, where labels identify the target row (Hoffer et al., 2018; Mettes et al., 2019). Language embeddings have likewise served as supervised targets in recognition and semantic segmentation (Frome et al., 2013; Li et al., 2022). Most closely, Wen et al. (2024) use CLIP embeddings of known class names as fixed prototypes for supervised long-tailed recognition; their separate DINO experiment retains learned visual prototypes and only subsamples them. SwAV also ablates a fixed random bank (Caron et al., 2020). These methods either use labels

to select externally defined prototypes or retain prototypes with no prescribed semantics; they do not ground an external semantic index to unlabelled images during self-supervised pretraining.

Post-hoc vision–language alignment: dino.txt aligns a text encoder to a frozen DINOv2 backbone, while LLM2CLIP adapts an LLM as the text encoder of a pretrained CLIP model (Jose et al., 2025; Huang et al., 2026). Both learn cross-modal alignment from paired image–caption data after visual pretraining; neither uses language to structure DINO's self-distillation objective.

E Analysis

We test whether LLM-defined caption groups remain coherent and retrievable in the frozen contrastive model's feature space, then examine how image-to-prototype assignments approach their final grouping during visual training.

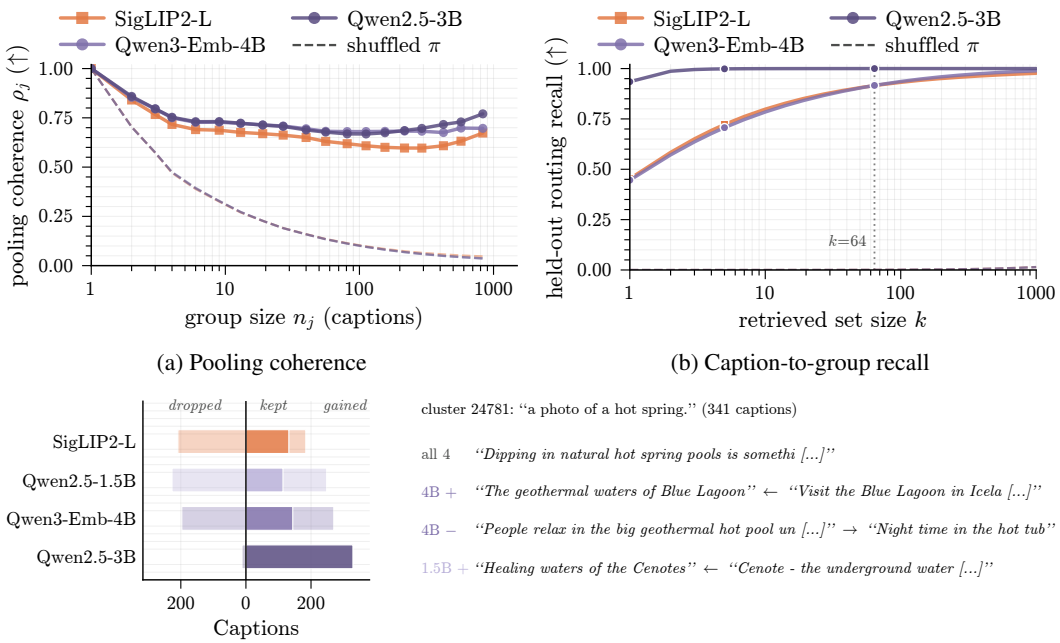
E.1 Shared caption groups across feature spaces

The routing bank pools SigLIP2 text features according to caption groups formed in Qwen2.5-3B space (Eq. 3). This construction does not require the two encoders to induce the same partition. We examine whether the Qwen-defined groups remain geometrically coherent after pooling in the routing space, and whether they can be recovered among the k groups used by the alignment target.

First, the groups remain coherent across text encoders. For a group j with n_j captions, we measure the resultant length $\rho_j = \|\sum_{i:\pi(i)=j} \boldsymbol{b}_i\|/n_j$. It approaches one when the member features point in a common direction and decreases when they cancel. Across all three feature spaces, the observed groups are substantially more coherent than size-matched shuffled groups (Fig. 3a). This includes SigLIP2-L, although its features did not define the partition.

The pooled rows also preserve group identity under retrieval. We recompute the query's group mean without that caption, score the query against the bank, and ask whether its assigned group is retrieved. At the k=64 support used by the alignment loss, recall is above 91% in both SigLIP2-L and Qwen3-Embedding-4B, compared with 0.07% after shuffling (Fig. 3b). Top-1 recall in these two spaces is only about 45%, so the assigned group often remains among the closest candidates even when it is not the nearest match. The hot-spring example in Fig. 3c illustrates this local ambiguity: captions exchanged by the group concern nearby concepts such as geothermal lagoons, hot tubs, and cenotes.

These results show that Qwen-defined groups remain coherent and retrievable in SigLIP2 space, supporting supervision through their shared group indices without requiring identical nearest-centroid partitions.



(a) Pooling coherence

(b) Caption-to-group recall

(c) One group in detail

Figure 3: **LLM-defined caption groups remain usable across feature spaces.** (a) Median pooling coherence by group size, with size-matched shuffled groups shown as dashed curves. (b) Leave-one-out recall of a caption's assigned group; the rule marks the k=64 support used for alignment. (c) Membership changes for one group after reassigning captions to their nearest group centroid in each feature space.

E.2 Assignment stability

We next examine how image-to-prototype assignments approach their final grouping during training.

At each checkpoint, we assign 50,000 centre-cropped ImageNet-1k validation images using the EMA teacher's argmax over its raw, pre-Sinkhorn logits, then compare the resulting partition with the final partition of the same run. Agreement is measured by normalised mutual information, adjusted for chance using a shuffled copy of the final assignments: $(\mathrm{NMI} - \mathrm{NMI}_{\mathrm{shuffle}})/(1 - \mathrm{NMI}_{\mathrm{shuffle}})$. Because NMI compares image groupings rather than prototype indices, the same measure applies to both fixed and learned banks. It captures when a run approaches its eventual grouping, not whether that grouping agrees with semantic labels.

Agreement for the aligned run crosses 0.5 between 15k and 30k iterations, compared with approximately 86k for DINOv3 and 99k for the random fixed-bank run (Fig. 4). The aligned configuration approaches its eventual image grouping earlier than either baseline, a pattern that freezing random prototypes alone does not reproduce.

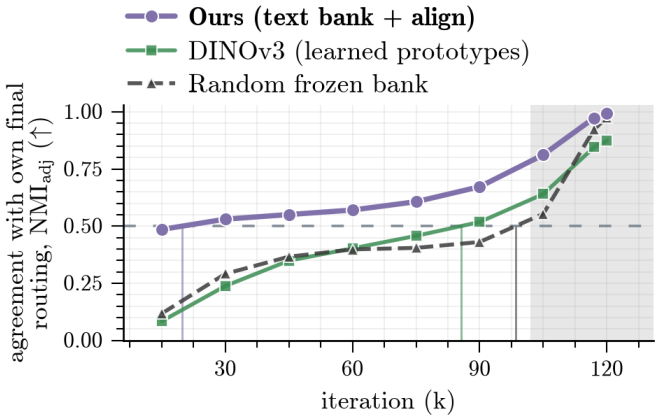


Figure 4: **Image-level assignment stability.** Shuffle-adjusted NMI between each checkpoint's CLS assignment partition and the final partition of the same run, measured on 50k validation images. Vertical rules mark interpolated crossings at 0.5; the final point, equal to one by construction, is omitted. Grey shading marks late learning-rate decay. One run is shown per configuration.

F LIMITATIONS

Pretrained components: `LexDINO` uses no captions paired with its training images. Its multimodal knowledge comes from the frozen router, the image tower of SigLIP2 (Tschannen et al., 2025), which has public weights but was trained on the non-public WebLI corpus of paired web images and text. Neither SigLIP2 nor the LLM is needed at inference, although the encoder may inherit biases from both models. Appendix B reports compute with and without a lower-bound estimate of SigLIP2's pretraining, and excludes LLM pretraining.

Run-to-run variation: The controlled ImageNet-1k results average three runs, while each ablation is a single run, so ablation differences of a few tenths of a point could be inconclusive. The three LLM-derived banks, each from a separate pretraining run, lie within 1.2 OCR points of each other, less than a fifth of the 6.3- and 6.7-point OCR gaps in Tables 11 and 12.

Training data at scale: Our large-scale image mixture differs from DINOv3's LVD-1689M, so comparisons with released models reflect both method and data. It is, however, considerably simpler to build. Unlike LVD-142M and LVD-1689M, it involves no similarity-based retrieval or clustering-based balancing, and its largest component is a random, uncurated sample of roughly 300M DataComp images (Gadre et al., 2023) (Section 3.5). Our results therefore do not rely on such curation, which is known to benefit self-supervised pretraining (Oquab et al., 2024). The controlled ImageNet-1k experiments fix the training images, backbone, and schedule to isolate the method's effect.

Data provenance: The caption corpus includes LAION-COCO, derived from LAION-2B-en, and the image mixture includes uncurated DataComp images. The captions serve only as text for constructing the prototype banks, but the vocabulary and the encoder may still reflect the content and biases of these sources.