Not All Layers Are Created Equal: Learning to Select Layers for Reliable Out-of-Distribution Detection

Ignacio M. De la Jara[1,3] Cristian Rodriguez-Opazo[2] Damith Ranasinghe[1,3]
[1]Adelaide University
[2]Australian National University
[3]Crossing Lab
ignacio.mezadelajara@adelaide.edu.au

Abstract

Intermediate CLIP representations can improve out-of-distribution (OOD) detection, but existing methods apply the same fixed intermediate-layer fusion rule to every input. We instead study whether layer selection can be learned. Voyager *keeps CLIP and its scores frozen and trains a lightweight router to select, for each image, one sparse expert defined by a final-layer-anchored subset of depths. The router uses few-shot ID examples and WebContEX, a web-retrieved weak OOD context built from the ID label set without target-OOD images. On ImageNet-1K,* Voyager *obtains* ***18.86*** *average FPR@95 and* ***94.49*** *AUROC, improving over Local-Prompt by* 8.8 *FPR points and fixed intermediate-layer fusion by* 11.8 *points. Controlled comparisons separate the roles of context and routing: under the same router recipe, replacing WebContEX with ID-derived context increases FPR from* 18.86 *to* 26.71*; with WebContEX and expert scores fixed, learned routing lowers FPR by* 10.1 *points relative to the strongest score-based selector. Across Pascal-VOC, ImageNet-100, near-OOD subsets, and additional CLIP backbones,* Voyager *consistently reduces aggregate FPR relative to matched fixed-layer or static-fusion baselines, demonstrating transfer across ID domains, near-OOD shifts, and backbones. Importantly, the deployed five-router ensemble trains in under three minutes with less than 1 GB of GPU memory, making* Voyager *lightweight and practical. Code:* https://anonymous.4open.science/r/Voyager/.

1. Introduction

Out-of-distribution (OOD) detection is essential when vision systems encounter inputs outside their intended label space. Vision-language models such as CLIP [34] are attractive in this setting because they provide strong zero-shot image-text confidence scores without task-specific backbone training [9, 28, 30, 38]. Most CLIP OOD detectors, however, score only the final representation and discard the intermediate representations that produced it. Evidence useful under one distribution shift can yield misleading confidence under another. We therefore ask *which layers should be trusted for each input*, rather than applying the same fusion rule to every image.

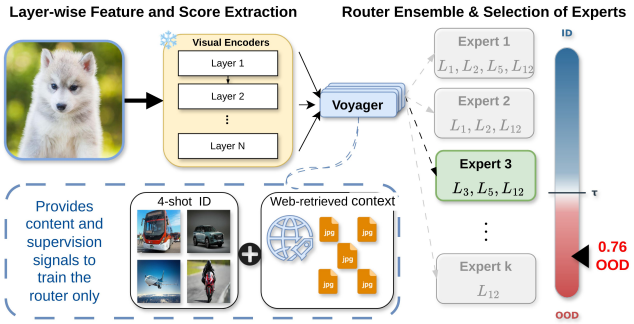


Figure 1. **Main idea.** Voyager treats CLIP depth as an input-conditioned choice. A lightweight router, trained with few-shot ID data and weak OOD context (web retrieved context), selects a sparse layer-subset expert for each image.

Final-layer confidence is a strong baseline. Intermediate representations have long been useful for OOD detection in discriminative networks [21], and recent CLIP studies find complementary, shift-sensitive evidence across depth [17, 19, 36]. In particular, Mysteries of the Deep (MoD) [19] improves on final-layer MCM by fusing a fixed subset of layers. Yet no single subset is uniformly best across inputs, OOD families, and backbones (Sec. 3). Consequently, *we learn layers to trust for each input and exploit layer-complementarity for OOD detection.*

Prompt-learning methods such as CoOp, LoCoOp, IDLike, and Local-Prompt [2, 30, 41, 44] adapt CLIP using few ID examples, sometimes synthesizing negative context from ID data. We instead keep the backbone, text prompts, and layer-wise detector scores frozen and introduce Voy-

ager for layer selection. Voyager builds a dictionary of sparse, final-layer-anchored experts, where each expert averages scores from one layer subset, and learns a small image-conditioned *router* over this dictionary.

Learning the router still requires a supervision signal. Thus, we introduce WebContEX, which automatically retrieves WordNet-filtered Wikimedia Commons images using only the ID label names. These images provide a binary, weak OOD context during router training; they are never used as a test-time score, and target-OOD examples are never used to construct them. Our ablations show that WebContEX and routing are both important. WebContEX improves substantially over ID-derived context (18.86 versus 26.71 FPR@95), while a controlled comparison with the same WebContEX pool shows that learned routing outperforms the strongest score-based selector by 10.1 points. We therefore claim that routing is an effective way to convert weak external context into layer selection, not that the supervision source is immaterial. Figure 1 illustrates our approach and Fig. 4 details the architecture. We summarize our key contributions below:

- We formulate frozen-CLIP OOD detection as *input-conditioned layer selection* over sparse, final-layer-anchored experts.
- We introduce **Voyager**, a standalone, lightweight learned router for selection and **WebContEX**, automatically retrieved weak OOD context, and quantify the contribution of each.
- We evaluate across three ID benchmarks, near-OOD subsets, and four CLIP backbones. Further, we include variance, scalability, same-context baselines, and explicit negative results to define the method's scope.

2. Related Work

Out-of-distribution Detection. OOD detection identifies inputs that deviate from the training distribution, a key challenge for safe deployment. Early approaches rely on confidence scores such as MSP [13] and ODIN [24], and statistical distances like Mahalanobis scoring [21] or Energy-based scoring [27]. Outlier Exposure [14] improves robustness by training models to assign low confidence to a small set of external non-ID samples, effectively regularizing decision boundaries and enhancing OOD detection. Additional advances explore residual features [39], hyperspherical embeddings [29], and activation shaping [8]. Foundational analyses examine learnability and scoring principles [3, 10, 32], while modality-specific taxonomies [1] contextualize anomaly types.

Vision–language Models for OOD Detection. Large vision–language models such as CLIP [35] enable training-free OOD detection via image–text similarity. Extensions include concept matching [28, 31], prompt tuning [22, 30], and synthetic outlier generation [2, 4]. Prompt-learning methods benefit only marginally from external outliers, limiting OOD generalization. Frozen-parameter methods such as SeTAR [23] avoid prompt tuning but still require labeled ID data to select their modification.

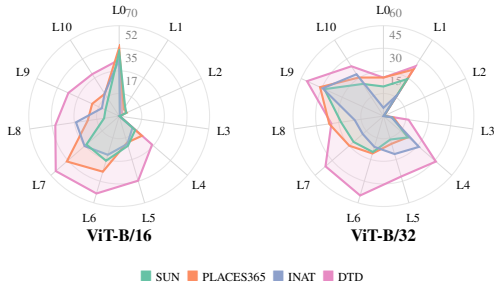


Figure 2. **OOD evidence diversity.** Retrospective layer-selection frequency by OOD dataset using all final-layer-anchored subsets with at most s=9 layers.

Intermediate-layer Representations. Early distance-based detectors used intermediate features [21], but most modern methods rely on final-layer outputs. Recent work shows that intermediate CLIP layers encode transferable, shift-stable signals [17, 36], often with additional supervision. Lin et al. [26] introduce multi-exit energy scoring, while Fayyad et al. [11] attach supervised auxiliary heads. Guglielmo and Masana [12] rank layers to select a single discriminative depth rather than fuse them. *Mysteries of the Deep* [19] shows that ID-selected intermediate-layer aggregation improves frozen-CLIP OOD detection, but relies on a fixed fusion rule that cannot adapt to each input.

3. Motivation: Why Static Fusion Fails

OOD detection with CLIP commonly uses MCM [28]—an image embedding is compared with prompt-based text prototypes, a softmax is applied over ID classes, and the maximum confidence is used as the ID score. Recent work shows that applying this idea beyond the final layer can improve detection because CLIP's intermediate representations retain low-level, semantic, and shift-specific cues [19]. The remaining question is whether a *fixed* layer subset can fully exploit this diversity.

Question: Is one layer subset enough? If the best intermediate evidence changes across inputs and OOD shifts, static fusion can improve over the final layer but still leave substantial performance unused.

3.1. OOD Evidence Diversity

We first ask whether one fixed layer subset can represent heterogeneous shifts. For each input, an analysis-only retrospective selector searches all K=1,981 final-layer-anchored subsets containing at most s=9 of the L=12 lay-

ers (the expert family defined in Sec. 4.1) and selects the subset with the strongest ID–OOD separation. Figure 2 summarizes the selected layers by OOD dataset. ViT-B/16 often assigns mass to early and mid-depth layers, especially for texture-centric shifts such as DTD, while scene-oriented shifts distribute mass more smoothly. ViT-B/32 has a different profile, with stronger concentration on late layers. Thus the most useful evidence changes with anomaly type and architecture; one global subset cannot match all of these depth profiles.

3.2. Margin-Based Separability

Different layer preferences matter only if fixing one subset weakens the ID–OOD decision boundary. We therefore analyze MCM margin distributions. For each sample, let $m(x) = \tau - S(x)$, where τ is an ID boundary estimated from in-distribution data and $S(x)$ is the score; larger positive margins indicate stronger OOD separation.

Figure 3 (left) compares four detectors: *Last* is final-layer MCM; *MoD* is the published fixed fusion rule [19]; *Static* retrospectively chooses the single best subset in the same s=9 family and applies it to every input; and *Retro.* chooses the best subset per input using labels unavailable at deployment. MoD and the strongest static subset shift mass toward positive margins, but the per-input selector shifts it further. Hence even the retrospectively best global subset leaves samples closer to the decision boundary, exposing the limitation of fixed selection.

3.3. How Sparse Can Layer Selections Be?

Fixed selection might fail simply because its subset contains too few layers. If so, allowing more layers should let the best fixed subset close the gap. We test this explanation by varying a maximum subset size s. Every candidate includes the final CLIP layer as a semantic anchor and at most $s-1$ intermediate layers, yielding the family $\mathcal{C}_s$. Because s is a ceiling, a candidate may use fewer than s layers.

Figure 3 (right) reports results for ViT-B/16. As s grows the deployed router's FPR@95 reduces to 18.86 at s=9 while the best fixed expert plateaus near 33.5. With the expert pool and WebContEX held fixed, this gap measures the contribution possible from per-input selection over choosing one global expert at various levels of sparsity (the complete per-dataset retrospective breakdown is in App. N, Fig. N.2).

Motivational insight: Selection matters beyond fusion. Intermediate layers are useful, but fixed fusion constrains their effective use. Margins identified in the retrospective diagnostics motivate learning to select, rather than designing another fixed fusion heuristic.

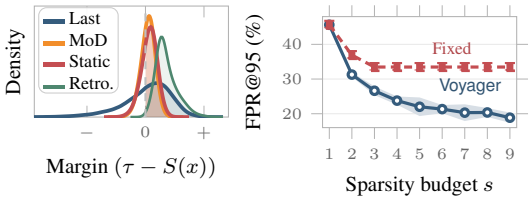


Figure 3. **Why fixed selection fails.** *Left:* Margin distributions for the final layer, MoD, the best fixed subset, and per-input retrospective selection over the same s=9 family. *Right:* Increasing s improves Voyager while the best fixed subset plateaus.

4. Methodology

The preceding analysis suggests that useful OOD evidence exists across depth, but that no fixed layer subset is reliable for every input. We therefore keep the pretrained evidence unchanged and learn only *which* depths to trust. Voyager uses a few-shot ID set $\mathcal{D}_{\text{ID}}$ and weak context $\mathcal{D}_{\text{ctx}}$ to train a router over frozen layer experts; the backbone, text prototypes, and layer-wise OOD scores are never optimized, and no target-OOD label or test image is used.

4.1. Frozen Layer Experts

To isolate routing from representation learning, we preserve the underlying OOD detector. At each depth ℓ, a frozen ID-confidence score $S_\ell(x)$ compares the normalized image representation with the ID text prototypes and summarizes how strongly the image matches the ID taxonomy: higher values are more ID-like. We use MCM or its global–local extension GL-MCM as specified in Sec. 6; Appendix Sec. D.1 gives the score and expert definitions.

A single intermediate layer provides incomplete evidence, while averaging all layers can dilute specialized cues. We therefore construct sparse candidates containing the final layer—a stable semantic anchor and at most $s-1$ intermediate layers. With L layers this gives $K = \sum_{r=0}^{s-1} \binom{L-1}{r}$ candidates. If L_i is expert i's subset, then

$$E_i(x) = \frac{1}{|L_i|} \sum_{\ell \in L_i} S_\ell(x). \tag{1}$$

The unweighted mean makes each expert purely a choice of layers, avoiding fusion parameters that could obscure the contribution of routing.

4.2. Learning Which Expert to Use

Raw expert confidences are not directly comparable because different layer subsets have different ID ranges. We align them at the evaluation operating point: the few-shot ID set defines expert i's 95%-retention boundary $b_i = Q_{0.05}(\{E_i(x) : x \in \mathcal{D}_{\text{ID}}\})$ and scale σ_i, the standard deviation of its fitting-set scores. Here $Q_{0.05}$ is the empirical 5th percentile and $\epsilon > 0$ stabilizes the denominator.

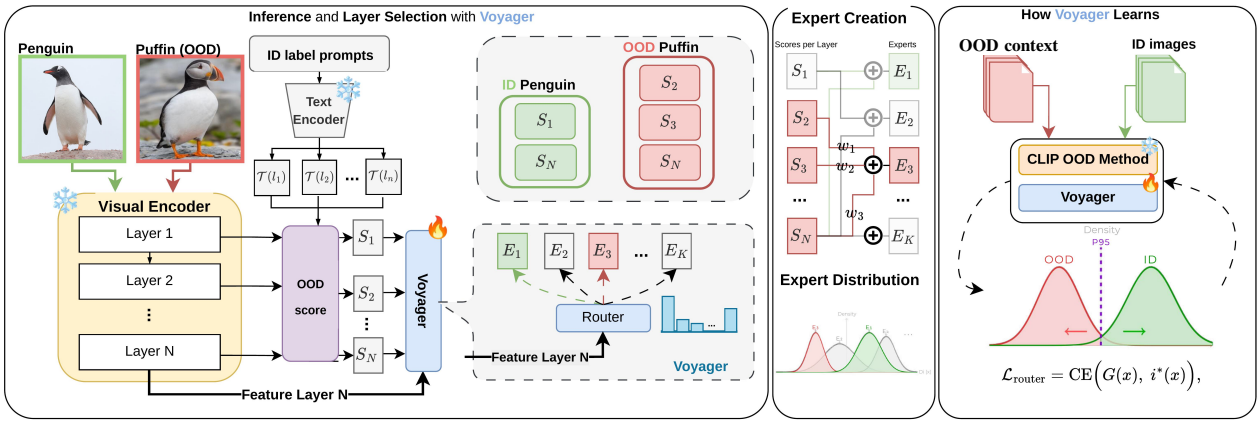


Figure 4. **Voyager method overview.** Frozen CLIP produces layer-wise OOD scores. A lightweight router selects one sparse, final-layer-anchored expert for each image; that expert's mean layer score is the detector output.

Thus $D_i(x) = (E_i(x) - b_i)/(\sigma_i + \epsilon)$ has a common meaning: positive is ID-like and negative falls below the expert's boundary.

This creates supervision without target-OOD data: ID samples rank experts by $D_i(x)$, while WebContEX samples rank them by $-D_i(x)$. Overlapping subsets often have nearly identical margins, making a one-hot choice unstable. A temperature-controlled softmax over the top k experts instead defines $q(x)$, assigning zero mass outside that set (Eq. (7)). This remains selective while sharing mass among plausible routes.

The router $G_\theta(x)$ predicts $q(x)$ from mean-pooled frozen layer features. It never receives expert scores, so it learns visual cues associated with trustworthy depths rather than a hand-crafted score rule. Only this small MLP is optimized, with $\mathcal{L} = \mathrm{CE}(q, G_\theta) + \lambda_{\mathrm{bal}}\mathcal{L}_{\mathrm{bal}}$. The two terms act at different levels: cross-entropy follows each image's soft expert target, while the weak batch-level term $\mathcal{L}_{\mathrm{bal}} = \sum_i f_i p_i$ discourages globally sending all inputs to one expert. For a batch B, $p_i = |B|^{-1} \sum_{x \in B} [G_\theta(x)]_i$ is mean router probability and f_i is the detached fraction of hard routes assigned to expert i. Gradients flow through p_i; no derivative through the hard assignment is needed. We use $\lambda_{\mathrm{bal}} = 0.01$. The separation value from the earlier draft has no router gradient and is excluded; Appendix Sec. H audits this choice.

4.3. Inference

At test time, a hard route preserves the intended interpretation—one input-conditioned layer subset supplies the score:

$$\widehat{i}(x) = \arg\max_i [G_\theta(x)]_i, \qquad \mathrm{Score}(x) = E_{\widehat{i}(x)}(x). \quad (2)$$

Few-shot draws can support slightly different but valid routes, so the deployed detector averages a small ensemble of independently seeded routers. All members share one frozen CLIP forward pass and expert table; only the inexpensive MLP decisions are repeated. No WebContEX image, context sample, or target-OOD example is required at inference.

5. WebContEX: Retrieval-Based Weak OOD Supervision

Voyager needs weak OOD context only to learn which frozen expert to route to. We introduce **WebContEX**, which constructs this context from ID class names and public web images. The procedure is taxonomy-agnostic: replacing the ID label set changes the semantic exclusion criterion, while retrieval and router supervision remain unchanged, and no target-OOD instance is required. ImageNet-100 and Pascal-VOC 1-shot reuse the same ImageNet-derived pool and rescore it against the new ID text prototypes. The Pascal-VOC val row instead uses a matched-size, globally unique refresh drawn with the same construction and disjoint from the deployed pool.

Concretely, WebContEX maps the ID labels and candidate web concepts to WordNet, then rejects direct overlaps and concepts that are too close to the ID taxonomy. Its retrieval stage can use the resulting queries with a searchable image repository. We instantiate the deployed method with Wikimedia Commons because it provides a large, diverse public pool; all headline WebContEX results use this instance. WebContEX samples the retained concepts evenly, removes low-quality or invalid images, and deduplicates each retrieval shard. The images receive only a binary proxy-OOD label: their retrieved concept names are not used as classification targets. WebContEX therefore teaches the router *which frozen layer expert to trust* without tuning prompts, modifying CLIP, or adding any test-time component.

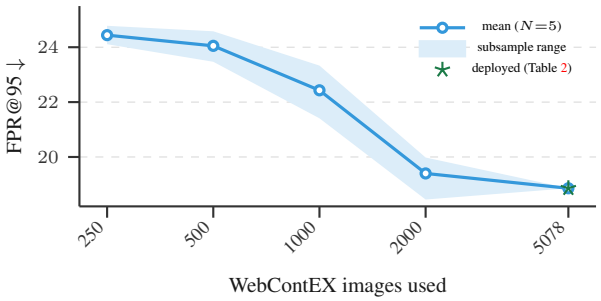


Figure 6. **WebContEX supervision-size sweep (ViT-B/16).** ImageNet-1K FPR@95 as the deployed router (N=5, s=9, top-32) is trained on increasing subsamples of the WebContEX pool (mean over four nested subsamples; shading is min–max).

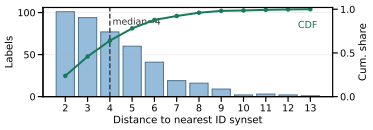
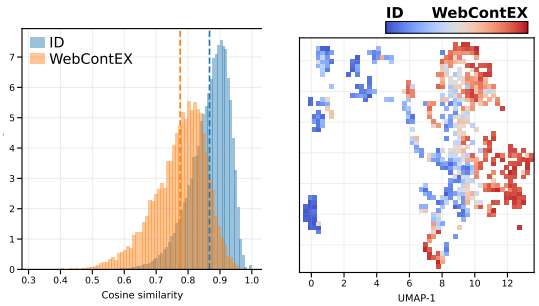


Figure 5. **WebContEX label filtering.** Retained labels are at least two WordNet hops from the ImageNet-1K taxonomy.

Why does WebContEX help? The proxy is useful because it is semantically separated from the selected ID taxonomy, yet remains close enough to expose difficult regions of the representation space. WordNet filtering removes direct ID overlap without assuming access to a target OOD set (Figure 5). In CLIP space, the retrieved samples lie near the ID manifold but shift away from exact ID matches, creating ambiguous inputs for which different layers can provide different evidence (Figure 7). This structure translates into a consistent empirical gain: increasing the context from 250 to 5,078 cache entries lowers FPR@95 from 24.43 to 18.86 (Figure 6). Thus WebContEX provides ID-conditioned, boundary-relevant routing context without requiring knowledge of the unknown evaluation shifts.

Proxy-source check. Table 1 isolates the effect of the supervision source while keeping the router, context size, and training recipe fixed. Beyond Wikimedia retrieval, we test three off-the-shelf external sources (Oxford Flowers-102, Food-101, and Stanford Cars) and two constructions using only ID data (local crops and synthetic features). WebContEX consistently outperforms both the best external proxy dataset and ID-only supervision. Notably, local crops from the few-shot ID images—similar to the supervision used by Local-Prompt—still improve Voyager over static fusion, showing that ID-derived context alone can learn useful routing, although retrieved WebContEX supervision remains substantially more effective. Full per-source results are reported in Appendix Table 10.



(a) Retrieval similarity (b) Local structure (UMAP)

Figure 7. **WebContEX retrieval analysis (ViT-B/16).** WebContEX samples lie near the ID manifold but are less similar than ID matches. UMAP colors indicate ID-dominated (blue), WebContEX-dominated (red), and mixed neighborhoods (pale).

Table 1. **Proxy-source check.** Using the same router, context size, and training recipe with only proxy source changes. Complete per-dataset results are in Table 10.

Proxy source	Avg. FPR↓	Gap
WebContEX	**18.86**	**–**
Best external dataset	23.74	+4.88
ID-only local crops (Local-Prompt-style [2, 41])	26.71	+7.85
Generic ID-only synthetic	27.38	+8.52
MoD static fusion	30.70	+11.84

6. Experimental Settings

Evaluation protocol. We evaluate ImageNet-1K, Pascal-VOC, and ImageNet-100 against their standard held-out OOD datasets. ImageNet-1K follows MoS [16], with SSB-Hard and NINCO from OpenOOD v1.5 for near-OOD evaluation [42]; Pascal-VOC follows the published frozen-CLIP protocol [19, 23]. Results use 1 or 4 ID shots as indicated.

Base detector. Unless otherwise stated, Voyager uses CLIP ViT-B/16 [34] and applies GL-MCM [31] at every layer.

Training. Router fitting uses few-shot ID features and the 5,078-entry WebContEX cache (Sec. 5). ImageNet-100 and VOC one-shot rescore the deployed pool; VOC full-validation uses a source-disjoint refresh. Table 8 records the split roles.

Router configuration. The default uses $s = 9$, a top-32 target with temperature 2, $\lambda_{\text{bal}} = 0.01$, and five fixed seeds. Router training and calibration never use target OOD data. Backbone-specific dictionary sizes are documented in Appendix Sec. D.1 and Table 9; Appendix Sec. L retains a separate prompt-view compatibility check.

Metrics. We report AUROC and FPR@95 measuring the false-positive rate on OOD samples when the ID true-positive rate is fixed at 95% (lower is better).

Table 2. **ImageNet-1K OOD detection (ViT-B/16).** FPR@95↓/AUROC↑; Avg. is over four shifts. Bold/underline: best/second-best per shot setting; ⋆: our rerun.

Method	iNat.	SUN	Places	DTD	Avg.
Zero-shot, training free					
MCM [28]	30.91/94.61	37.59/92.57	44.69/89.77	57.77/86.11	42.74/90.77
GL-MCM [31]	17.42/96.44	**30.75/93.44**	37.62/90.63	55.20/85.54	35.25/91.51
MoD [19]	**15.98/96.90**	45.58/89.69	**35.71/92.72**	**25.51/94.84**	**30.70/93.54**
One-shot					
CoOp [44]	18.95/95.52	29.58/92.90	38.72/89.64	48.03/85.87	33.82/90.98
LoCoOp [30]	21.67/95.69	**22.98/95.07**	31.41/92.10	49.79/87.85	31.46/92.68
ID-Like [2]	14.57/97.35	44.02/91.08	41.74/91.15	**26.77/94.38**	31.78/**93.49**
Local-Prompt⋆ [41]	18.52/96.46	34.79/92.93	41.11/90.60	54.45/87.66	37.22/91.91
Voyager	**4.38/98.48**	25.64/94.21	**24.64/92.36**	32.85/88.74	**21.88**/93.45
Four-shot					
CoOp [44]	14.60/96.62	28.48/94.98	36.49/89.98	43.13/88.03	30.67/91.82
LoCoOp [30]	16.05/96.86	23.44/**95.94**	32.87/91.98	42.28/90.19	28.66/93.74
ID-Like [2]	8.98/98.19	42.03/91.64	44.00/90.57	**25.27/94.32**	30.07/93.68
Local-Prompt [41]	9.65/97.87	**20.40**/95.57	29.39/92.67	51.20/88.08	27.66/93.53
Voyager	**3.93/98.78**	21.65/95.13	**21.88/93.48**	27.96/90.58	**18.86/94.49**

7. Main Results

7.1. OOD Detection Across Datasets and Backbones

Few-shot fitting scope. Core few-shot entries use each method's original released configuration and a separate fit per ID benchmark, shot count, and backbone. That model is frozen across all OOD columns; no per-target retraining is performed.

ImageNet-1K: different shifts favor different signals. Table 2 prioritizes the complete native one- and four-shot prompt grid. At four shots, prompt-learning FPR ranges from 27.66 to 30.67: Local-Prompt is strongest on the scene shifts, ID-Like on iNaturalist and DTD, and LoCoOp is more balanced. Thus each prompt strengthens a different part of the OOD signal. For essential frozen controls, MCM/GL-MCM are zero-shot and MoD uses ID data to choose one fixed layer fusion; GL-MCM is strongest on SUN, while MoD is strongest on the other three shifts. No fixed representation is uniformly best.

Routing combines these depth-dependent cues more effectively. Standalone Voyager reaches 18.86 average FPR@95, improving the strongest four-shot prompt baseline by 8.80 points. Classical post-hoc scores appear in Appendix Table 21; CSP comparisons and prompt-view controls appear in Table 17. These broaden the comparison beyond the core prompt-learning grid.

Pascal-VOC: competitive overall, with complementary shift strengths. Table 3 changes the ID task to a multi-label taxonomy. Voyager ranks second overall at 16.62/96.41, trailing LoCoOp by 2.39 FPR@95 and 0.26 AUROC; we therefore do not claim the best Pascal-VOC result. The average, however, hides complementary behavior: Voyager has lower FPR on four of six shifts—iNaturalist (0.31 vs. 1.18), SUN (10.67 vs. 13.31), Places (8.62 vs. 16.14), and ImageNet-22K (14.22 vs. 14.73)—whereas LoCoOp is markedly stronger on DTD and COCO. Voyager also improves GL-MCM and MoD by 12.82 and 10.35 points, and the recent Local-Prompt by 21.31 points.

This reversal cautions against judging a learned detector on one dataset alone. Local-Prompt is the strongest prompt baseline on ImageNet-1K (27.66), but reaches 23.39 on ImageNet-100 and 37.93 on Pascal-VOC; Voyager obtains 18.86, 12.46, and 16.62, ranking first, first, and second, respectively. We interpret this as more consistent transfer across taxonomies, not uniform dominance: texture-heavy DTD and category-overlapping COCO remain limitations. Because the proxy and ensemble-size comparison included the reported Pascal target suite, this configuration is target-selected rather than strictly target-blind.

Table 3. **Pascal-VOC OOD detection (ViT-B/16).** FPR@95↓/AUROC↑; Avg. is over six shifts. Bold/underline: best/second-best; ⋆: our rerun.

Method	iNat.	SUN	Places	DTD	IN-22K	COCO	Avg.
Training-free (MCM/GL-MCM zero-shot; SeTAR/MoD val-fitted)							
MCM [28]	10.51/97.93	30.45/94.25	36.11/91.86	53.21/91.77	53.82/91.12	57.10/89.02	40.20/92.66
GL-MCM [31]	4.33/98.81	22.94/94.63	26.20/93.11	41.61/92.88	37.88/93.17	43.70/90.71	29.44/93.88
MoD [19]	2.19/98.92	19.70/95.77	19.53/95.11	44.08/92.06	34.70/91.15	41.60/88.09	26.97/93.52
Prompt learning (native full-validation fit, seed 1)							
CoOp⋆ [44]	5.07/98.40	18.85/96.50	21.32/94.88	17.62/96.44	17.59/96.41	37.80/90.75	19.71/95.56
LoCoOp⋆ [30]	1.18/99.40	13.31/97.31	16.14/95.77	**11.83/97.27**	14.73/96.97	**28.20/93.32**	**14.23/96.67**
ID-Like⋆ [2]	28.32/95.80	29.81/94.64	24.66/94.68	65.80/87.68	64.06/80.12	62.40/84.91	45.84/89.64
Local-Prompt⋆ [41]	6.52/98.38	26.39/94.78	30.47/92.74	53.46/91.18	57.25/91.42	53.50/89.25	37.93/92.96
Few-shot routed detectors							
Voyager 1-shot	0.89/99.44	15.83/96.26	20.84/95.51	32.80/93.56	19.84/96.17	44.40/88.80	22.43/94.96
Voyager val	**0.31/99.81**	**10.67/97.67**	**8.62/97.91**	23.81/95.38	**14.22/97.37**	42.10/90.30	16.62/96.41

ImageNet-100: the pattern persists under a smaller taxonomy. In Table 4, LoCoOp is again the strongest prompt rerun at 15.18, while CoOp only matches MoD and ID-Like and Local-Prompt are worse. Voyager improves to 12.46 by capturing stronger iNaturalist, Places, and DTD signals, although LoCoOp remains better on SUN. Prompt learning is therefore useful when its bias matches the shift, but the standalone routed detector gives the strongest macro result in this core comparison.

Table 4. **ImageNet-100 OOD detection (ViT-B/16, 4-shot).** FPR@95↓/AUROC↑; Avg. is over four shifts. Bold/underline: best/second-best; ⋆: our rerun.

Method	iNat.	SUN	Places	DTD	Avg.
MCM [28]	18.13/96.77	36.45/94.54	34.52/94.36	41.22/92.25	32.58/94.48
GL-MCM [31]	17.26/96.90	34.52/94.77	33.69/94.54	38.33/93.38	30.95/94.90
MoD [19]	13.72/97.40	37.71/93.68	25.41/95.71	12.87/**97.44**	22.43/96.06
CoOp⋆ [44]	15.58/97.02	22.89/96.14	25.31/94.92	25.39/95.20	22.29/95.82
LoCoOp$^{\star}_{\mathrm{GL}}$ [30]	6.68/98.49	**11.30/97.58**	16.57/96.31	26.17/94.75	15.18/96.78
ID-Like⋆ [2]	4.59/98.87	46.27/91.22	50.91/89.43	16.49/96.87	29.56/94.10
Local-Prompt⋆ [41]	6.83/98.08	25.02/95.85	25.77/95.53	35.92/93.82	23.39/95.82
Voyager	**1.94/98.98**	19.75/96.14	**15.90/96.53**	**12.23**/97.30	**12.46/97.24**

Additional backbones: gains depend on depth structure. The narrow RN50x4 and ViT-B/32 FPR gains in Table 5, together with their AUROC regressions, show when routing has little room to help. Appendix Fig. N.1 finds RN50x4 expert complementarity mostly below C=0.07, versus maxima near 0.26 for B/16 and B/32; its four convolutional stages therefore expose fewer distinct signals. Alignment is another risk: a matched audit degrades from 45.40/88.82 with CLIP's pretrained downsampling to 56.66/84.81 with

Table 5. **Backbone transfer on ImageNet-1K (4-shot).** FPR@95↓/AUROC↑. The core comparison excludes prompt-fusion extensions; bold marks the best metric per backbone.

Method	iNat.	SUN	Places	DTD	Avg.
RN50x4 (MCM routed scores)					
MCM	44.00/91.59	35.13/92.83	44.30/89.38	57.22/85.99	45.16/89.95
MoD	**41.79/92.00**	**31.83/93.48**	**40.46/90.36**	52.38/87.37	41.61/**90.80**
Voyager	44.21/89.23	37.87/92.03	44.45/88.57	**39.15/89.61**	**41.42**/89.86
ViT-B/32 (GL-MCM routed scores)					
MCM	33.85/93.62	40.99/91.56	46.71/89.25	60.90/85.03	45.61/89.87
MoD	**12.02/97.64**	28.98/**93.37**	35.69/**91.61**	39.36/**91.07**	29.01/**93.42**
Voyager	21.41/95.58	**26.56**/93.17	**34.54**/90.79	**31.95**/90.00	**28.62**/92.39
ViT-L/14 (MCM routed scores)					
MCM	46.19/91.42	32.58/93.34	37.02/91.86	54.95/86.19	42.68/90.70
MoD	51.82/89.43	47.86/88.97	27.84/**93.67**	56.26/84.29	45.94/89.09
Voyager	**16.19/96.85**	**23.65/94.16**	**26.30**/92.52	**44.49/87.52**	**27.66/92.76**

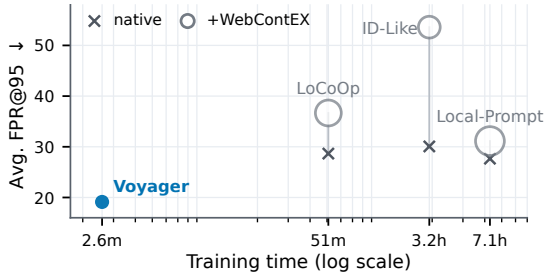


Figure 8. **Detection–Cost Pareto.** *Circles*: Matched 4-shot ImageNet-1K/WebContEX. *Crosses*: Native-FPR. Bubble area denotes peak GPU memory; Voyager uses $N{=}8$.

a frozen random $1{\times}1$ projection. The reported RN50x4 row uses the pretrained path, so low complementarity better explains its remaining limitation. Still, Voyager lowers FPR on all three additional backbones. Most clearly, on the 24-block ViT-L/14 it improves MoD by 18.28 FPR and 3.67 AUROC points. The RN result therefore motivates architecture-aware intermediate-stage experts rather than reusing Transformer-style fusion unchanged.

Shared-context efficiency. Figure 8 controls WebContEX access and cost. With identical data, the $N{=}8$ Voyager audit trains in 2.6 minutes/0.9 GB at 19.12 FPR@95; LoCoOp/ID-Like/Local-Prompt need 50.9 minutes–7.1 hours/13.1–22.9 GB and yield 36.67/53.60/31.13, worse than their native 28.66/30.07/27.66. Unlike these few-shot prompt learners, Voyager is designed to learn from external weak OOD context: holding the router and recipe fixed, WebContEX improves over ID-local crops from 26.71 to 18.86 (Table 1). Prompt methods adapt an ID-derived negative construction; Voyager instead keeps CLIP and its prompts frozen and uses external images only to learn which depth to trust. Thus the difference is the ability to use context, not merely access to it; $N{=}5$ is cheaper.

7.2. Near-OOD Detection

SSB-Hard [37] and NINCO from OpenOOD v1.5 [42] provide complementary ImageNet-1K near-OOD tests. Table 6 compares fixed scores, all four native 4-shot prompt baselines, and standalone Voyager. The strongest prompt method, Local-Prompt (79.22/72.06), trails fixed CSP (75.71/74.90); Voyager gives the lowest FPR (61.87), while CSP retains the best AUROC (74.90).

Table 6. **OpenOOD v1.5 near-OOD (ImageNet-1K ID, ViT-B/16).** FPR@95↓/AUROC↑; *Near*: avg. of *NINCO* & *SSB-Hard*.

Method	SSB-Hard	NINCO	Near
MCM	89.45/63.71	79.62/73.56	84.53/68.64
GL-MCM	85.33/67.82	72.92/75.72	79.13/71.77
CSP	82.81/**71.82**	68.61/77.98	75.71/**74.90**
CoOp	89.07/63.85	77.85/71.43	83.46/67.64
LoCoOp	86.59/67.58	76.15/71.66	81.37/69.62
ID-Like	91.47/49.85	74.40/75.41	82.93/62.63
Local-Prompt	85.75/67.91	72.69/76.20	79.22/72.06
Voyager	**71.39**/64.19	**52.35/81.15**	**61.87**/72.67
CSP-prompt Voyager	82.20/71.32	66.23/77.24	74.22/74.28
Dual-Prompt CSP Voyager	77.42/67.42	60.94/80.95	69.18/74.18

7.3. Controlled Design Analysis

Is learned selection necessary? Table 7 fixes the experts, scores, and WebContEX. Moving from the final layer (45.64 FPR) to one globally selected expert (34.60) shows that the sparse family is useful, but fixed selection cannot adapt. A per-input maximum-score rule reaches 28.93, whereas the feature-conditioned router reaches 18.75. This additional 10.18-point gain isolates learned per-image trust rather than mere access to intermediate scores; the analysis-only oracle at 7.06 shows remaining headroom.

Table 7. **Selector ablation.** Fixed experts, scores, and WebContEX; only selection changes. ImageNet-1K & average FPR@95.

Selector	Selection signal	FPR@95↓
Last-layer	Fixed final-layer expert	45.64
Fixed context	One global expert (WebContEX)	34.60
Best score rule	Per-input max expert score	28.93
Learned Voyager router	**Image feature → expert**	**18.75**
Label-aware oracle	Analysis-only upper bound	7.06

Which design choices matter? Figure 9 gives a consistent picture. The final-layer anchor reaches 18.86, versus 23.51 for the best intermediate anchor, so auxiliary layers work best as corrections to a stable semantic reference. Increasing the subset budget accumulates complementary cues (31.28 at $s{=}2$, 22.01 at $s{=}5$, and 18.86 at $s{=}9$). Capacity is not the explanation: one/three/five hidden layers give 20.56/18.86/19.95. Instead, mean-pooled cross-layer features (18.86) beat last-layer, max-pooled, and concatenated inputs (21.07/20.21/20.00). The final panel shows gains saturating near five routers, motivating the stability analysis below.

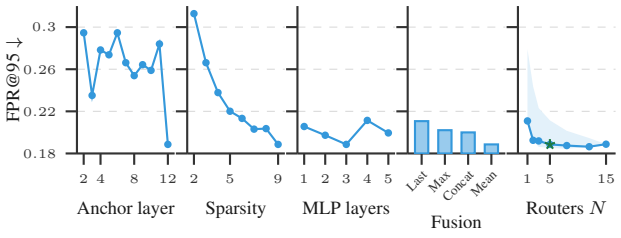


Figure 9. **Ablation study (FPR@95 ↓).** Across anchor layer, subset size s, router depth, input aggregation, and ensemble size.

7.4. Why Multiple Routers?

Near-tied experts make a single router seed-sensitive, so we average small heads that share frozen features and scores and output selected scalar scores, not votes [20, 33]. Across 15 seeds, individual FPR spans 21.09–27.90. Exact routes agree only 5–7%, yet their layer sets overlap (Jaccard 0.47–0.59) and scores correlate (0.63–0.74). Averaging improves $21.09 \rightarrow 19.25 \rightarrow 18.86$ for $N{=}1, 2, 5$ and then saturates; five disjoint ensembles give 19.91 ± 0.79, well below the 8.80-point main gain but not a paired significance test. A fixed-draw control still improves $21.09 \rightarrow 20.48$, isolating optimization diversity. Appendix Fig. J.1 and Table 15 give the full audit.

7.5. What Layers Does Voyager Select?

We audit the deployed GL-MCM routes over all 50,000 ImageNet-1K validation and 35,640 original-suite OOD images: 428,200 decisions across five routers, without a new model or selection step. Target labels only group routes after inference; no target image or label fits or selects a router. Block 12 is mandatory and omitted.

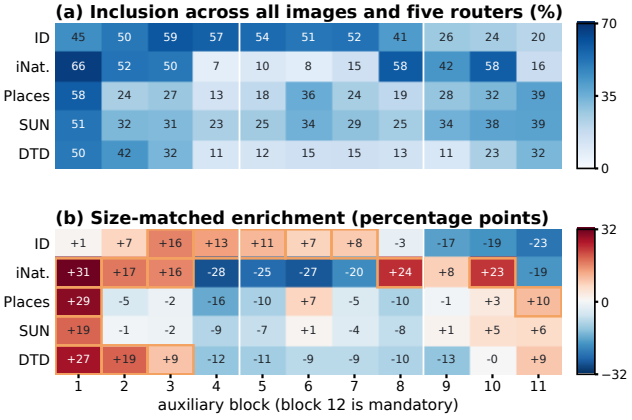


Figure 10. **Layer distribution of standalone Voyager.** (a) Inclusion over every ImageNet-1K validation and original-suite OOD image with five routers. (b) Per-router inclusion minus its dataset mean over the 11 exchangeable auxiliary blocks, controlling for route size. Orange outlines mark positive enrichment for all five routers. Block 12 is mandatory and omitted.

Figure 10(a) shows shift-dependent depth and sparsity. ID routes average 5.80 blocks and enrich blocks 3–7; iNaturalist combines blocks 1–3 with 8/10. DTD has the smallest routes (3.58) and enriches 1–3. Places and SUN have aligned centered profiles ($r{=}0.97$), favoring block 1 while retaining more late evidence than DTD; their routes average 4.16 and 4.60 blocks.

Panel (b) subtracts each router's dataset mean so larger experts do not appear important merely by size. Given low exact-route agreement but moderate layer-set and score agreement, we interpret reproducible depth profiles and route sizes rather than one expert identity. The Appendix Shapley check reinforces that selection frequency describes the policy, not causal layer importance.

Where does the evidence change spatially? Figure 11 shows clean-ID-normalized crop scores for post-hoc successes: MCM/MoD retain local ID-like plant or texture evidence, whereas Voyager produces broader OOD evidence and crosses the image threshold. These examples are spatial, not aggregate or causal; Appendix Sec. Q provides full-set and target-blind route distributions.

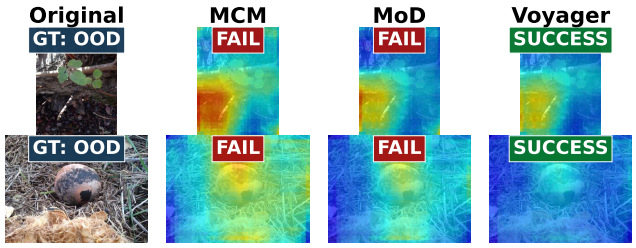


Figure 11. **Where routing changes the OOD evidence.** Sliding-window maps for post-hoc success cases. Scores are threshold-normalized; red/blue denotes ID-/OOD-like evidence.

Conclusion

We introduced Voyager, a lightweight framework that learns which frozen VLM layers to trust for each image. This per-image selection is our central contribution: it exploits complementary layer biases without altering the backbone or using WebContEX at inference. Controlled comparisons separate WebContEX's weak routing context from the gains of adaptive selection over fixed experts, static fusion, score rules, and matched prompt-learning baselines. Across three ID benchmarks and multiple CLIP backbones, the core detector shows where routing transfers and where it does not. A full-set, five-router audit further shows reproducible dataset-level changes in selected depth and route size, while explicitly limiting the interpretation to the learned routing policy rather than causal layer importance.

References

[1] Udit Arora, William Huang, and He He. Types of out-of-distribution texts and how to detect them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10687–10701. Association for Computational Linguistics, 2021. 2

[2] Yichen Bai, Zongbo Han, Changqing Zhang, Bing Cao, Xiaoheng Jiang, and Qinghua Hu. Id-like prompt learning for few-shot out-of-distribution detection, 2024. 1, 2, 5, 6, 20

[3] Julian Bitterwolf, Alexander Meinke, Maximilian Augustin, and Matthias Hein. Breaking down out-of-distribution detection: Many methods based on ood training data estimate a combination of the same core quantities. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pages 2041–2074. PMLR, 2022. 2

[4] Chentao Cao, Zhun Zhong, Zhanke Zhou, Yang Liu, Tongliang Liu, and Bo Han. Envisioning outlier exposure by large language models for out-of-distribution detection, 2024. 2

[5] Mengyuan Chen, Junyu Gao, and Changsheng Xu. Conjugated semantic pool improves ood detection with pre-trained vision-language models, 2024. 22

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014. 12

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255. Ieee, 2009. 11, 12

[8] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection, 2023. 2

[9] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot out-of-distribution detection based on the pre-trained model clip. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6568–6576, June 2022. 1

[10] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[11] Jamil Fayyad, Kashish Gupta, Navid Mahdian, Dominique Gruyer, and Homayoun Najjaran. Exploiting classifier inter-level features for efficient out-of-distribution detection. *Image Vision Comput.*, 142(C), Feb. 2024. 2

[12] Guglielmo, Gianluca and Masana, Marc. Leveraging intermediate representations for better out-of-distribution detection. In *Proceedings of the 28th Computer Vision Winter Workshop (CVWW)*, pages 53–61. Verlag der Technischen Universität Graz, 2025. 2

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2018. 2

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure, 2019. 2

[15] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 12

[16] Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space, 2021. 5, 11, 12

[17] Fran Jelenić, Josip Jukić, Martin Tutek, Mate Puljiz, and Jan Šnajder. Out-of-distribution detection by leveraging between-layer transformation smoothness, 2024. 1, 2

[18] Xue Jiang, Feng Liu, Zhen Fang, Hong Chen, Tongliang Liu, Feng Zheng, and Bo Han. Negative label guided ood detection with pretrained vision-language models, 2024. 22

[19] I. M. De la Jara, C. Rodriguez-Opazo, D. Teney, D. Ranasinghe, and E. Abbasnejad. Mysteries of the deep: Role of intermediate representations in out of distribution detection, 2025. 1, 2, 3, 5, 6, 11

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 8

[21] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018. 1, 2

[22] Tianqi Li, Guansong Pang, Xiao Bai, Wenjun Miao, and Jin Zheng. Learning transferable negative prompts for out-of-distribution detection, 2024. 2

[23] Yixia Li, Boya Xiong, Guanhua Chen, and Yun Chen. Setar: Out-of-distribution detection with selective low-rank approximation, 2024. 2, 5, 11

[24] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 12

[26] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection, 2021. 2

[27] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Tao. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2020. 2

[28] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations, 2022. 1, 2, 6, 11, 23

[29] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection?, 2023. 2

[30] Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. Locoop: Few-shot out-of-distribution detection via prompt learning, 2023. 1, 2, 6, 20, 22

[31] Atsuyuki Miyai, Qing Yu, Go Irie, and Kiyoharu Aizawa. Gl-mcm: Global and local maximum concept matching for zero-shot out-of-distribution detection, 2025. 2, 5, 6

[32] Peyman Morteza and Yixuan Li. Provable guarantees for understanding out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAI)*, 2022. 2

[33] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019. 8

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 5

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[36] Arnas Uselis and Seong Joon Oh. Intermediate layer classifiers for ood generalization, 2025. 1, 2

[37] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations (ICLR)*, 2022. 7

[38] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no, 2023. 1

[39] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching, 2022. 2

[40] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, pages 3485–3492. IEEE, 2010. 12

[41] Fanhu Zeng, Zhen Cheng, Fei Zhu, Hongxin Wei, and Xu-Yao Zhang. Local-prompt: Extensible local prompts for few-shot out-of-distribution detection, 2025. 1, 5, 6, 20

[42] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. Openood v1.5: Enhanced benchmark for out-of-distribution detection, 2024. 5, 7

[43] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2017. 12

[44] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, July 2022. 1, 6

A. Appendix Overview and Organization

This appendix provides additional material that complements the main paper and documents both supporting and negative evidence for Voyager. In particular, it expands the discussion of datasets, implementation details, limitations, source dependence, few-shot robustness, retrieval-based OOD supervision, theoretical motivation, layer-importance diagnostics, and interpretability.

For clarity, the appendix is organized as follows:

- **Sec. B** details the in-distribution, evaluation, and proxy OOD datasets used throughout the paper.
- **Sec. C** describes *WebContEX*, a retrieval-based strategy for constructing proxy OOD context from Wikimedia Commons.
- **Sec. D** records the default router architecture and score-flow used in the main ImageNet-1K experiments.
- **Sec. D.1** gives the complete mathematical definition of the expert dictionary, ID-boundary normalization, routing target, loss, and ensemble.
- **Sec. E** reports inference runtime and latency overhead.
- **Sec. F** ablates the selection mechanism, showing the learned router beats every score-based selection rule and the static baselines.
- **Sec. G** ablates the routing-target sharpness (hard top-1, top-k, and fully soft) used by the deployed detector.
- **Sec. H** audits every loss term and tests an expert-overlap-aware target.
- **Sec. I** reports how Voyager scales with the number of ID shots, showing that 1–2 shots are sufficient.
- **Sec. J** reports the full ensemble-size curve and route-stability audit supporting the main-paper ablation.
- **Sec. K** details how the prompt-learning baselines are re-trained under the same WebContEX context as Voyager, and the harness used to measure training time and memory.
- **Sec. L** defines the two-view extension and reports its component and fusion-sensitivity audits.
- **Sec. M** reports near-OOD, fixed-recipe transfer, and classical-score checks, including limitations.
- **Sec. N** analyzes expert complementarity and the routing opportunity across sparsity levels and backbones.
- **Sec. O** provides complementary theoretical justification for the value of intermediate layers and dynamic routing.
- **Sec. P** provides a post-hoc layer-importance diagnostic for the fixed Table 2 detector.
- **Sec. Q** presents full-set and target-blind per-image layer-route distributions for standalone Voyager.
- **Sec. R** discusses the main limitations of Voyager, with emphasis on the dependence on proxy OOD supervision.

Taken together, these sections isolate the roles of proxy context and learned selection, document few-shot and boundary sensitivity, and show that complementary intermediate evidence creates the opportunity exploited by the routing policy.

B. Dataset description

In-Distribution Dataset

ImageNet-1K. ImageNet-1K [7] is the primary in-distribution (ID) dataset for the standard CLIP OOD suite. Following the few-shot setting, we sample a small number of images per class (1-shot and 4-shot) from the ImageNet training set to train the Voyager router. These samples are used only for router supervision; the CLIP backbone remains frozen.

For evaluation, we follow the standard ImageNet-based OOD protocol [16, 28] and use the full ImageNet-1K validation split, consisting of 50,000 images across 1,000 categories. All OOD detection results reported in the main paper are computed using this zero-shot evaluation setting.

Pascal-VOC. We also evaluate on the multi-label Pascal-VOC ID benchmark used in prior frozen-CLIP work [19, 23]. The ID fitting pool is the published validation split (94 images over 14 classes), while ID evaluation uses the disjoint VOC test split (906 images). The Voyager (val) row in Table 3 uses all 94 validation images; the 1-shot row uses one image per class.

ID split separation. For every ID benchmark, router-training images are distinct from evaluation images. On ImageNet-1K, the router receives 4-shot samples from the training split and evaluation uses the validation split. On Pascal-VOC, the Voyager (val) row fits on the published validation pool and evaluates on disjoint VOC test images, matching the validation-fitted MoD protocol. Table 8 summarizes the exact roles and counts.

Table 8. **Data split accounting.** ID train/fitting images are disjoint from ID evaluation images. Core transfer and appendix compatibility rows reuse the same ImageNet-derived WebContEX pool, while Pascal-VOC val uses a matched-size, source-disjoint balanced refresh. No reported target-OOD evaluation image is used for router training or proxy construction.

Benchmark	Role	Split/source	# images	Use in this paper
ImageNet-1K	ID router train	ImageNet train	4,000	4-shot ID supervision and ID-boundary estimation for the router; sampled from the training split only.
ImageNet-1K	Router / fusion context	WebContEX	5,078	Cached images retrieved from ImageNet-1K ID names; context for router fitting only.
ImageNet-1K	ID evaluation	ImageNet val	50,000	ID side of final FPR@95/AUROC evaluation; never used for router fitting.
ImageNet-1K	OOD evaluation	iNat./SUN/Places/DTD	10,000/10,000/10,000/5,640	Held-out OOD benchmarks used only for reporting final metrics.
ImageNet-100	ID router train	ImageNet-100 train subset	400	4-shot ID supervision for the subset stress test in Table 4.
ImageNet-100	Router / fusion context	WebContEX	5,078	Same ImageNet-derived cache entries, rescored against ImageNet-100 text prototypes.
ImageNet-100	ID evaluation	ImageNet-100 val subset	5,000	ID side of the ImageNet-100 FPR@95/AUROC evaluation; disjoint from the 4-shot train subset.
ImageNet-100	OOD evaluation	iNat./SUN/Places/DTD	10,000/10,000/10,000/5,640	Held-out OOD benchmarks used only for reporting final metrics.
Pascal-VOC	ID router train	Published val split	94	Full-val Voyager row uses the same ID validation pool used to fit the MoD configuration; the 1-shot row uses 14 images.
Pascal-VOC	Router / fusion context	WebContEX	5,078	The val row uses a globally unique, source-disjoint balanced refresh; the 1-shot and prompt-view rows rescore the deployed ImageNet-derived pool. Neither contains VOC test data.
Pascal-VOC	ID evaluation	VOC test	906	ID side of the Pascal-VOC evaluation; disjoint from the validation pool.
Pascal-VOC	OOD evaluation	SUN/Places/DTD/iNat./COCO/IN-22K	10,000/10,000/5,640/10,000/1,000/10,000	Held-out OOD benchmarks from the published frozen-CLIP protocol; used only for reporting final metrics.

Table 9. **Configurations used in reported router results.** All rows use the same MLP, optimizer, and six epochs; WebContEX has 5,078 rows. Pascal-VOC val uses the matched-size, source-disjoint balanced refresh; the 1-shot and prompt-view rows use the deployed pool. "Full" denotes a soft target over all K experts.

Use	Backbone / score	s	Target	N	ID fit
ImageNet-1K main	ViT-B/16, GL-MCM	9	top-32	5	4-shot
Matched LoCoOp view (IN-1K)	ViT-B/16, LoCoOp-MCM	9	top-32	5	4-shot
CSP auxiliary (12-layer)	ViT-B/16 or B/32, CSP	9	top-32	5	4-shot / full val
WebContEX size sweep	ViT-B/16, GL-MCM	9	top-32	5	4-shot
ImageNet-100	ViT-B/16, GL-MCM	7	full	8	4-shot
Matched LoCoOp view (IN-100)	ViT-B/16, LoCoOp-MCM	7	full	8	4-shot
Pascal-VOC 1-shot	ViT-B/16, GL-MCM	7	full	8	1-shot
Pascal-VOC val	ViT-B/16, MCM	9	top-32	16	full val
Matched LoCoOp view (VOC)	ViT-B/16, LoCoOp-MCM	7	full	8	full val
Near-OOD IN-10/20	ViT-B/16, MCM	7	full	5	4-shot
ViT-B/32	ViT-B/32, GL-MCM	8	top-32	5	4-shot
RN50x4 standard	RN50x4, MCM	2	top-2	8	4-shot
RN50x4 CSP auxiliary	RN50x4, CSP	2	full	5	4-shot
ViT-L/14	ViT-L/14, MCM	4	top-32	5	4-shot
Shared-context diagnostic	ViT-B/16, GL-MCM	9	top-32	8	4-shot
Selector / few-shot diagnostics	ViT-B/16, GL-MCM	9	top-32	8	1–16 shots

Out-of-Distribution Datasets

We adopt the four OOD datasets filtered by the MoS protocol [16], which removes all ImageNet-overlapping categories to ensure a clean distribution shift.

iNaturalist. [15] contains fine-grained flora and fauna categories. We use the 10,000-image filtered version from [16], which excludes any classes that map to ImageNet-1K.

SUN. [40] consists of indoor and outdoor scene images across thousands of semantic categories. Following [16], we use the de-duplicated 10,000-image subset with all ImageNet-overlapping classes removed.

Places. [43] provides large-scale scene categories. We use the 10,000-image subset filtered by [16] to ensure no semantic overlap with ImageNet-1K.

Texture (DTD). [6] includes 5,640 texture images annotated using 47 human-interpretable attributes. This dataset introduces low-level distribution shifts distinct from object-centric ImageNet categories.

Pascal-VOC OOD suite. For Pascal-VOC, we follow the published frozen-CLIP protocol and evaluate against six held-out OOD sets: the four datasets above plus COCO [25] and an ImageNet-22K subset [7]. These OOD images are evaluation-only and are not used to fit Voyager or MoD.

B.1. Weak OOD Supervision for Voyager

Voyager uses a small amount of weak OOD supervision to train the dynamic router; this proxy context is *not* used during evaluation. The supervision is provided by *WebContEX*, a retrieval-based proxy context constructed automatically from Wikimedia Commons using ImageNet-1K label names (no hand-curated outlier data); we detail its construction in Sec. C. Transfer experiments reuse this same image pool and rescore it against the new ID taxonomy.

C. WebContEX (Web Context via Semantic Exclusion): Retrieval-Based Weak OOD Supervision

Constructing a proxy OOD context dataset can be challenging in practice, particularly in domain-specific settings

Algorithm C.1 WebContEX construction. Retrieval-based weak OOD supervision from ID labels only.

Require: ID label set $\mathcal{C}_{\mathrm{ID}}$, candidate web labels $\mathcal{C}_{\mathrm{web}}$, WordNet graph $\mathcal{G}$, image-search retriever $\mathcal{R}$ (Wikimedia Commons here), distance threshold $d_{\min}$=2, per-label image cap m

Ensure: WebContEX context $\mathcal{D}_{\mathrm{ctx}}$ and cached frozen-CLIP features/scores

Map each ID label $c \in \mathcal{C}_{\mathrm{ID}}$ to WordNet synsets $S_{\mathrm{ID}}(c)$.
Initialize retained query set $\mathcal{Q} \leftarrow \emptyset$ and context set $\mathcal{D}_{\mathrm{ctx}} \leftarrow \emptyset$.
for all candidate label $q \in \mathcal{C}_{\mathrm{web}}$ **do**
    Map q to candidate synsets $S(q)$.
    Compute $d(q, \mathcal{C}_{\mathrm{ID}})$, the shortest WordNet path from $S(q)$ to any ID synset.
    if $S(q)$ overlaps an ID synset or $d(q, \mathcal{C}_{\mathrm{ID}}) < d_{\min}$ **then**
        discard q
    else
        add q to $\mathcal{Q}$
    end if
end for
for all retained query $q \in \mathcal{Q}$ **do**
    Retrieve images $\mathcal{I}_q \leftarrow \mathcal{R}(q)$ from the configured repository.
    Keep at most m images that pass resolution, file-integrity, and deduplication checks.
    Add the filtered images to $\mathcal{D}_{\mathrm{ctx}}$ with binary context label y=1.
end for
Cache frozen CLIP features and per-layer scores for $\mathcal{D}_{\mathrm{ctx}}$.
Train the router with domain labels y=0 for few-shot ID images and y=1 for WebContEX context images.
return $\mathcal{D}_{\mathrm{ctx}}$ and cached features/scores.
Constraint: no target-OOD label, target-OOD training image, or evaluation image is an input to WebContEX.

where curated outlier data may not be readily available. To explore a more accessible alternative, we introduce *WebContEX*, a retrieval-based strategy that constructs weak OOD supervision from web images for a given labeled ID distribution. The goal of WebContEX is to demonstrate that the router in Voyager can learn effective routing behavior from retrieval-based weak OOD supervision and is not restricted to curated, hand-collected benchmark outlier sets. Algorithm C.1 gives the full construction procedure.

WebContEX first identifies candidate labels that are semantically separated from the in-distribution classes. To this end, we leverage the WordNet hierarchy and measure the shortest-path distance between candidate synsets and the synsets corresponding to the ID labels supplied by the user or benchmark. Labels whose semantic distance exceeds a small threshold (at least two hops in our experiments) are retained as potential OOD categories. As illustrated in Fig. 5, this heuristic encourages semantic separation while still allowing a mixture of relatively nearby and more distant concepts. Including semantically closer categories is particularly useful for generating challenging near-OOD examples, which provide informative supervision for learning robust routing behavior. Importantly, this procedure is intended to encourage semantic disjointness with the in-distribution classes, although constructing perfectly disjoint OOD proxy sets remains an open problem.

Once the candidate labels are selected, we retrieve images from Wikimedia Commons[1], an open repository containing over 136 million images, and use them to construct the proxy OOD context for router training. Sampling is balanced across the selected labels so that the proxy context is not dominated by a small number of visual concepts. Example retrieved images are shown in Fig. C.1.

C.1. Why retrieval-based augmentation works

To understand why WebContEX provides useful supervision for training the router, we analyze the representation similarity between retrieved samples and in-distribution data. Main-paper Fig. 7a shows the cosine similarity between image embeddings and their closest ID neighbors (ImageNet-1K in our main experiments). While ID samples naturally exhibit very high similarity to other ID examples, WebContEX images remain relatively close to the ID manifold but are consistently shifted toward lower similarity values. This indicates that retrieved images tend to form *near-OOD* examples: they share semantic or visual structure with the in-distribution classes while still deviating from them. Such samples provide informative supervision, as they expose the model to ambiguous regions where the boundary between ID and OOD is difficult to determine.

We further visualize the embedding space using the UMAP projection in main-paper Fig. 7b. The map reveals regions where ID and WebContEX samples overlap as well as regions dominated by OOD examples. These mixed regions correspond to areas of high uncertainty where distinguishing ID from OOD requires more nuanced representation cues. Training the router with WebContEX therefore encourages the model to learn routing policies that are sensitive to these local shifts in the representation space, allowing it to better exploit intermediate-layer signals when encountering distribution shifts at test time.

WebContEX construction details (reproducibility)

We document the exact construction settings used to build the deployed WebContEX pool so the proxy context can be reproduced from the released builder.

Candidate vocabulary. The candidate OOD vocabulary is drawn from WordNet noun synsets (`wordnet.all_synsets(pos="n")`). A synset is retained only if it is semantically disjoint from the

[1] https://commons.wikimedia.org/wiki/Main_Page

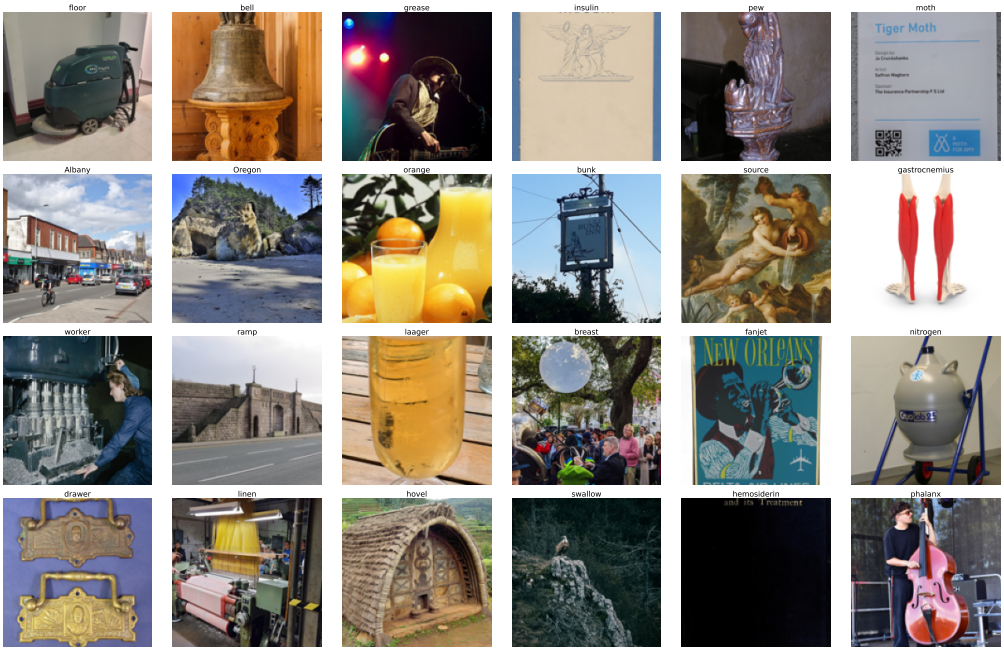


Figure C.1. **Example WebContEX retrievals.** Random samples from the Wikimedia Commons proxy context used only for router supervision.

ID label set: we form the union of the ID synsets and their full WordNet ancestor and descendant closures (hypernyms/instance-hypernyms and hyponyms/instance-hyponyms), and discard any candidate falling in that union. Surviving candidates are further filtered to a minimum WordNet depth of 5, to the allowed lexical categories `noun.animal`, `noun.plant`, and `noun.artifact`, and to a "simple" primary lemma (lowercase alphabetic, no underscores) whose maximum lemma frequency lies in $[5, 80]$ to prefer mid-frequency concrete concepts over hyper-generic or rare ones. Selection is deterministic under a fixed seed (seeded tie-jitter), and the disjointness is verified by an overlap report asserting $|Y_{\text{WebContEX}} \cap Y_{\text{ImageNet}}| = 0$.

Query formation from labels. For each selected synset, search queries are formed from its WordNet lemma names: underscores are replaced by spaces, and only simple alphabetic terms of length ≥ 3 are kept. From each base term we generate up to three query variants—the bare term, "`<term> photo`", and a lexname hint (e.g. "`<term> animal`", "`<term> plant`", "`<term> object`")—capped at 5 queries per synset, with up to 80 candidate results fetched per query from the Wikimedia Commons search API.

Images-per-label cap. Sampling is balanced across labels: each synset has a target of 20 images, and per-synset collection stops once that quota is met. To avoid intra-synset redundancy, candidates are additionally capped at a maximum fraction of the per-synset quota from any single query and from any single aspect-ratio bucket.

Duplicate removal. Within each independently retrieved shard, duplicates are removed at three levels: (i) image URL and Commons page id are deduplicated during candidate collection; (ii) exact duplicates are removed by SHA-256 over the re-encoded JPEG bytes; and (iii) near-duplicates are removed with an 8×8 DCT-based perceptual hash, rejecting any image within a Hamming distance of 8 of an already-accepted image. The released manifest retains the source and hash of every deployed cache entry for reproducibility. These within-shard checks and semantic filters do not establish image-level disjointness from every evaluation dataset; no cross-dataset near-duplicate audit is reported here.

Failed-query and download handling. A search query that raises an API error is logged and skipped, and the synset proceeds with its remaining queries. Each image download is attempted up to 2 times with linear backoff (base 0.4 s) and an 18 s per-request timeout, retrying on transient HTTP responses (429/503/504); downloads that

still fail, or images that fail to decode, are skipped. Synsets that do not reach their quota are recorded as partial and used to backfill the final selection when needed.

Integrity and resolution filtering. Every accepted image must have width and height of at least 256 pixels; SVG/vector files are rejected, EXIF orientation is normalized, transparent (RGBA/LA) images are flattened onto a white background, and images are re-encoded to RGB JPEG (quality 95). A lightweight quality heuristic down-weights candidates whose title or description matches noisy tokens (e.g. logo, icon, diagram, map, flag, chart) and those with extreme aspect ratios, prioritizing photographic content.

Deployed pool. Applying this procedure yields the deployed WebContEX cache of 5,078 Wikimedia Commons entries, materialized as an `ImageFolder`-compatible union and used only at training time. To prevent live-web drift, the revision artifacts include the exact ordered manifest with the Commons URL, query, license, attribution, dimensions, and SHA-256 for every entry. A separately staged deterministic JPEG archive has SHA-256 `ff7e6a38...c71c`; the manifest and packager verify all 5,078 cached paths without relying on live re-retrieval.

C.2. Proxy-context source ablation

Table 10 isolates the effect of the proxy-context *source*, holding the router and recipe fixed (deployed s=9, top-32, N=5); every context is sampled to the WebContEX size. It compares the deployed Wikimedia retrieval with three off-the-shelf external datasets and two proxies constructed entirely from ID data. Three findings emerge. (i) **ID-only local context is already useful:** a local-region crop proxy built from freshly sampled few-shot ID images reaches 26.71 FPR@95 and improves over static MoD (30.70). A generic ID-only feature-synthesis proxy (27.38) shows the same trend. (ii) **The source matters substantially:** WebContEX is 7.85 points stronger than ID-only local context and 4.88 points stronger than the best off-the-shelf external dataset. External retrieval is therefore a major contributor, not a nuisance variable. (iii) **The supervision labels matter:** randomly permuting ID/OOD-context labels collapses performance to 51.36. Together with the same-context selector ablation, these results support complementary contributions from WebContEX and learned routing.

Table 11 tests a stricter alternative explanation: perhaps WebContEX works simply because a binary ID-vs-WebContEX classifier transfers to the benchmark OOD sets. It does not. Direct MLP classifiers trained on the final-layer feature or on the same mean-pooled multilayer feature used by the router both exceed 54 FPR@95, while Voyager remains at 18.86. Thus WebContEX does not transfer effectively as a direct proxy-domain classifier in this setup; it is more useful as supervision for choosing among frozen layer experts.

Table 10. **Proxy-context source ablation (ImageNet-1K ID, ViT-B/16, 4-shot).** Each cell reports FPR@95↓/AUROC↑; Avg. is the macro-average across OOD datasets. Every context is sampled to the WebContEX size (5,078) and trained with the identical router (s=9, top-32, N=5); only the outlier-context *source* changes. Voyager (deployed) uses WebContEX; the generic ID-only synthetic row uses context generated from the few-shot ID features (best of mixup/extrapolation/noise); the ID-only local-crops row uses small local image regions of freshly-sampled few-shot ID images (ID-Like/Local-Prompt-style). **Bold** marks the best per column among the routing proxies; the lower block is reference (static MoD; and a shuffled-label control that destroys the ID/OOD supervision).

Proxy-context source	iNat.	SUN	Places	DTD	Avg.
Retrieved / generated context (routing)					
WebContEX (deployed)	**3.93/98.78**	**21.65/95.13**	21.88/93.48	27.96/90.58	**18.86/94.49**
ID-only local crops (ID-Like)	22.01/94.53	33.95/93.22	25.06/93.44	**25.82/92.45**	26.71/93.41
Generic ID-only synthetic	17.37/95.91	33.66/93.41	25.98/93.42	32.52/90.26	27.38/93.25
External proxy datasets (sampled to 5,078)					
Oxford Flowers-102	15.97/96.13	29.79/94.05	**20.83/94.45**	28.39/91.23	23.74/93.96
Food-101	15.32/96.28	30.55/94.09	22.47/94.02	29.10/91.25	24.36/93.91
Stanford Cars	18.79/95.25	29.54/94.05	22.68/93.79	31.99/89.38	25.75/93.12
Reference (no routing / broken supervision)					
MoD static fusion	15.98/96.90	45.58/89.69	35.71/92.72	25.51/94.84	30.70/93.54
Shuffled labels (control)	32.06/93.20	52.83/88.38	61.07/84.36	59.47/82.32	51.36/87.06

Table 11. **Direct proxy-classifier control.** Binary ID-vs-WebContEX MLPs use the same 4-shot ID data and full WebContEX pool, but their classifier probability is used directly as the OOD score instead of routing over frozen experts. Avg. FPR@95↓ on ImageNet-1K.

Detector	Avg. FPR@95↓
Direct MLP, final-layer feature	54.38
Direct MLP, mean-pooled multilayer feature	56.96
Voyager router, same WebContEX context	**18.86**

D. Default Architecture and Implementation Details

Figure D.1 expands the default Voyager detector used for the main ImageNet-1K results. The architecture is intentionally split into a frozen scoring path and a small trainable routing path. The CLIP image encoder, text encoder, prompts, layer-wise GL-MCM scores, ID thresholds, and expert table are all fixed before router training; only the router MLP parameters are learned.

Frozen scoring path. For the default ViT-B/16 setting, CLIP provides L=12 visual layers. We compute a GL-MCM confidence score at each layer and construct every sparse expert as a mean over a layer subset that always includes the final layer $\ell^\star$=11. With sparsity budget s=9, the

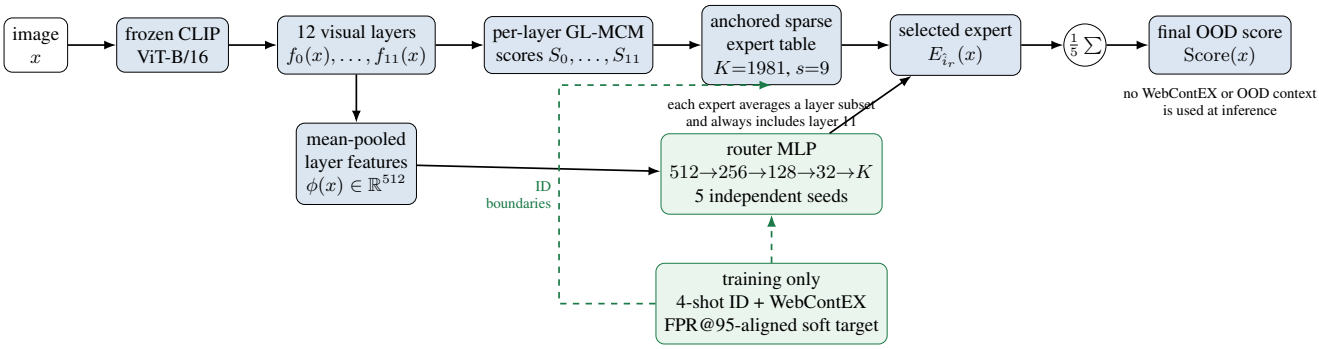


Figure D.1. **Default Voyager architecture.** The frozen scoring path produces layer-wise GL-MCM scores and a fixed dictionary of final-layer-anchored sparse experts. A small router MLP, trained only from 4-shot ID and WebContEX context, selects one expert per router; the deployed detector averages the five selected expert scores.

candidate family contains

$$K = \sum_{r=0}^{8} \binom{11}{r} = 1981$$

experts. Each expert has its own ID boundary estimated from the few-shot ID features, and expert scores are simple frozen reductions of the cached layer scores.

Router and ensemble. The router input is the mean-pooled layer feature vector $\phi(x) \in \mathbb{R}^{512}$. The default MLP has hidden widths $256, 128, 32$ and outputs a distribution over the K experts. It is trained for 6 epochs with AdamW, learning rate 10^{-2}, weight decay 10^{-4}, batch size 64, and the top-k FPR@95-aligned soft target from Eq. 7 (deployed k=32; ablated in Sec. G). The deployed detector trains five routers with independent seeds and averages their selected expert scores, as in Eq. 10. This ensemble changes only the scalar score aggregation; it does not require additional CLIP forward passes.

Training versus inference. WebContEX and the 4-shot ID set are used only to train the router and estimate ID boundaries. At test time, Voyager receives a single image, runs frozen CLIP once, computes the frozen expert table, selects one expert per router, and returns the average selected score. No WebContEX image, target-domain OOD example, or proxy-dataset score is used during inference.

Reproducibility and environment. All reported numbers use PyTorch 2.10 (CUDA 12.8) on a single GPU, with the five fixed router seeds $\{0, \ldots, 4\}$ and the cached 4-shot ID draws. Because the router is trained on *cached* per-layer features and GL-MCM scores (the frozen CLIP forward is computed once and stored), the full training-and-evaluation pipeline is deterministic and reproduces run-to-run bit-for-bit within a fixed environment; we release the environment specification and seeds with the code. As is typical of GPU training, the hard $\arg\max$ over the K=1981 near-tied expert candidates makes the *absolute* FPR@95 mildly sensitive (well under one point) to the CUDA/cuDNN version, so we fix and report the environment to reproduce the headline numbers exactly; the routing conclusions and rankings are unaffected.

Architecture insight: The learned part is small. The trainable component is only the router ensemble. The backbone, prompts, per-layer GL-MCM scores, expert definitions, and ID boundaries remain fixed, so Voyager acts as a lightweight selection layer over an already computed frozen-CLIP score table.

D.1. Full Method Definition

This section expands the intuitive description in Sec. 4 and fixes every mathematical convention used by the deployed detector.

Layer scores and expert dictionary. Let $f_\ell(x) \in \mathbb{R}^d$ and $t_c \in \mathbb{R}^d$ be normalized image and text representations. At depth ℓ,

$$z_{\ell,c}(x) = \gamma \langle f_\ell(x), t_c \rangle, \qquad S_\ell(x) = \max_c \frac{\exp z_{\ell,c}(x)}{\sum_{c'} \exp z_{\ell,c'}(x)}, \tag{3}$$

where $\gamma = 1/T$ is the fixed inverse temperature of the underlying MCM/GL-MCM detector. With final layer $\ell^\star = L - 1$ and sparsity budget s, the anchored dictionary is

$$\Lambda(L, s) = \{A \cup \{\ell^\star\} : A \subseteq \{0, \ldots, L-2\},\ |A| \le s - 1\},$$
$$K = \sum_{r=0}^{s-1} \binom{L-1}{r}. \tag{4}$$

Each $L_i \in \Lambda(L, s)$ returns the mean in Eq. (1). The deployed ViT-B/16 model uses $L = 12$, $s = 9$, and $K = 1981$.

Clean-ID boundary normalization. For each expert, the few-shot ID fitting split defines

$$\begin{aligned} b_i &= Q_{0.05}(\{E_i(x) : x \in \mathcal{D}_{\text{ID}}\}), \\ \sigma_i &= \text{Std}(\{E_i(x) : x \in \mathcal{D}_{\text{ID}}\}), \\ D_i(x) &= \frac{E_i(x) - b_i}{\sigma_i + \epsilon}. \end{aligned} \tag{5}$$

Here $\epsilon > 0$ is a numerical stabilizer. Positive distance is ID-like; negative distance falls below that expert's 95%-retention boundary. We use the standard linearly interpolated empirical quantile and never use ID evaluation or target-OOD data to estimate it. The estimate is naturally noisier for very small pools such as Pascal-VOC one-shot, so the disjoint full-validation VOC setting is primary.

Routing target. Training samples have domain label $y(x) = 0$ for ID and $y(x) = 1$ for WebContEX context. We rank experts by

$$M_i(x) = \begin{cases} D_i(x), & y(x) = 0, \\ -D_i(x), & y(x) = 1. \end{cases} \tag{6}$$

Let $\mathcal{N}_k(x)$ contain the k largest policy values and $Z_k(x) = \sum_{j \in \mathcal{N}_k(x)} \exp(M_j(x)/\tau_{\text{tgt}})$. The soft target is

$$q_i(x) = \begin{cases} \exp(M_i(x)/\tau_{\text{tgt}})/Z_k(x), & i \in \mathcal{N}_k(x), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The deployed setting uses $k = 32$ and $\tau_{\text{tgt}} = 2$. The router takes the fixed mean-pooled layer feature $\phi(x)$ and outputs $G_\theta(x) \in \Delta^{K-1}$. Its route loss is

$$\mathcal{L}_{\text{route}} = -\mathbb{E}_x \sum_{i=1}^{K} q_i(x) \log[G_\theta(x)]_i. \tag{8}$$

Balancing and ensemble. For a batch B, define mean router mass $p_i = |B|^{-1} \sum_{x \in B} [G_\theta(x)]_i$ and detached hard-assignment frequency $f_i = |B|^{-1} \sum_{x \in B} \mathbf{1}\{\arg\max_j [G_\theta(x)]_j = i\}$. Then $\mathcal{L}_{\text{bal}} = \sum_i f_i p_i$, and the deployed objective is

$$\mathcal{L} = \mathcal{L}_{\text{route}} + 0.01 \mathcal{L}_{\text{bal}}. \tag{9}$$

Because f_i is detached, $\nabla_\theta \mathcal{L}_{\text{bal}} = \sum_i f_i \nabla_\theta p_i$; the differentiable probability term supplies the gradient. There is no loss-weight schedule. For router r, $\widehat{i}_r(x) = \arg\max_i [G_{\theta_r}(x)]_i$; the five-router detector returns

$$\text{Score}_{\text{ens}}(x) = \frac{1}{R} \sum_{r=1}^{R} E_{\widehat{i}_r(x)}(x), \qquad R = 5. \tag{10}$$

The members share the frozen backbone, layer scores, and expert table, so no additional CLIP forward pass is required. Table 9 records benchmark-specific settings, while Secs. G, H, and J provide the corresponding ablations.

Table 12. **Single-image inference cost (ViT-B/16, one GPU).** End-to-end latency (raw image to OOD score, *no* cached features), throughput, and peak allocated GPU memory over 300 runs. Voyager adds only 0.10 ms over fixed 12-layer GL-MCM and no measurable peak-memory increase.

Detector	Latency (ms)↓	Throughput↑	Peak GPU↓
MCM	3.21	312	0.39 GB
GL-MCM fixed	4.08	245	0.43 GB
Voyager deployed	**4.18**	**239**	**0.43** GB

E. Inference Runtime and Latency

Table 12 reports the single-image inference cost of Voyager on ViT-B/16, measured end-to-end from the raw image with no cached features. The deployed detector runs in 4.18 ms (239 images/s) with a 0.43 GB peak GPU footprint. Compared with a fixed 12-layer GL-MCM detector, the input-conditioned router adds only 0.10 ms (+2.5%) and no measurable peak-memory increase, because the heavy CLIP forward and per-layer scores are shared across experts.

F. Selection Mechanism: Learned Routing vs. Rules

A central question is whether Voyager's gain comes from *input-conditioned routing* or merely from access to the sparse expert family and its scores. Figure F.1 isolates this: every bar uses the same anchored sparse experts, the same frozen GL-MCM scores, and the same WebContEX context; only the per-input selection mechanism changes. We compare (i) static experts (the last layer, and the single best expert chosen on the context), (ii) score-based selection rules that pick one expert per input from a heuristic on the expert scores (random, min/max expert score, and the largest absolute margin from the per-expert ID boundary), (iii) the learned router, and (iv) the per-input label-aware oracle.

The learned router reaches 18.8 FPR@95, beating the strongest rule (max-score, 28.9) by roughly ten points and the static last-layer baseline by more than twenty-five. Crucially, the router does not consume the expert scores—it selects from the image feature alone—so these score-based rules serve as representative non-learned baselines, not upper bounds over all non-learned selectors. That they all trail the router shows the benefit is the learned, input-conditioned policy, not the scores themselves. The oracle (7.1) marks the remaining headroom of the expert family under perfect per-input selection. (Here the router is the extended-budget N=8 ensemble, evaluated on cached features; the deployed detector uses N=5.)

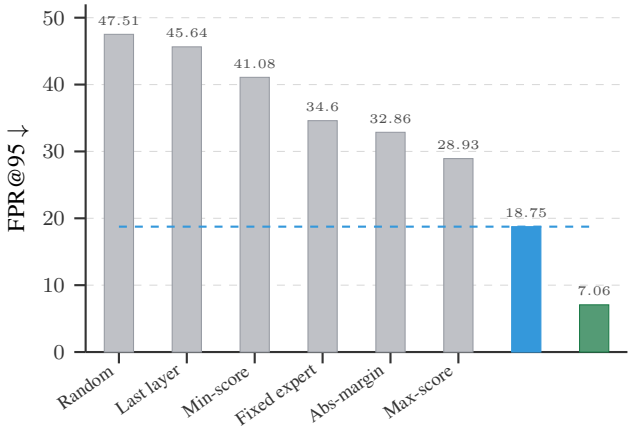


Figure F.1. **Selection mechanism, not the score (ViT-B/16, WebContEX).** All bars use the same anchored sparse expert family and the same context; only the per-input *selection mechanism* changes. **Gray**: score-based selection rules (pick an expert per input by a heuristic on the expert scores) and the static last-layer/fixed-context experts. **Blue**: the learned router (dashed line). **Green**: the per-input oracle (label-aware headroom). The learned gate beats the best rule (max-score, 28.9) by ∼10 FPR@95 and is far below the static baselines, establishing a routing contribution under fixed scores and context rather than merely reading off a max/min/margin of the expert scores. Configuration: N=8 routers, 4-shot ID, top-32 target.

Routing insight: The gate is the mechanism. Given identical experts, scores, and context, a learned input-conditioned gate beats every score-based selection rule by ∼10 FPR@95, the fixed-context expert by ∼16, and the last-layer baseline by >25. The improvement is the routing policy, not access to the expert scores.

G. Routing Target Ablation

The router is trained with a soft, FPR@95-aligned target rather than a hard one-hot choice of the single best expert. Figure G.1 ablates the target sharpness by varying the number of target experts k while keeping the rest of the recipe fixed (three independent N=8 ensembles per setting). A hard top-1 target is robustly worst: at the FPR@95 boundary many experts are near-tied, so committing the target to a single expert is brittle. Relaxing the target helps, but a fully soft target over all 1981 experts spreads its mass too widely. The top-32 target gives the best result in this diagnostic, beating the full soft target by ∼0.8 FPR@95 (19.7 vs. 20.5) with non-overlapping error bars. We therefore use top-32 in the deployed recipe.

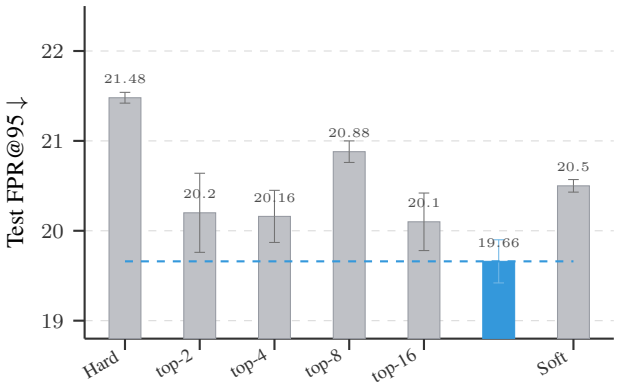


Figure G.1. **Routing target ablation (ViT-B/16, WebContEX).** Mean FPR@95 over three independent N=8 ensembles. A hard top-1 target is worst (21.48), while the top-32 target (dashed line, 19.66) beats the full soft target (20.50) by about 0.8 FPR@95.

Table 13. **Router-loss audit (ImageNet-1K, ViT-B/16).** All rows use the deployed top-32, N=5 recipe. The hard-gather separation value has no router gradient; a genuinely differentiable version is harmful.

Objective	FPR@95↓	AUROC↑
$\mathcal{L}_{\text{route}}$	19.61	94.40
$\mathcal{L}_{\text{route}} + 0.01\mathcal{L}_{\text{bal}}$ (deployed)	**18.86**	**94.49**
$\mathcal{L}_{\text{route}} + \mathcal{L}_{\text{sep}}^{\text{soft}}$	56.75	83.29
$\mathcal{L}_{\text{route}} + \mathcal{L}_{\text{sep}}^{\text{soft}} + 0.01\mathcal{L}_{\text{bal}}$	60.78	82.35

Target insight: Soft, but not too soft. A hard one-hot target is brittle because many experts are near-tied at the FPR@95 boundary, while a fully soft target over all experts spreads mass too widely. A soft target restricted to the top-32 near-optimal experts is strongest in this diagnostic.

H. Loss and Expert-Structure Audit

An implementation audit found that the separation loss described in the previous draft had been evaluated on the frozen expert score selected by a hard $\arg\max$. Because the gather index is discrete and the gathered score is frozen, this value has no gradient to the router. Consequently, the deployed 18.86 detector was trained by $\mathcal{L}_{\text{route}} + 0.01\mathcal{L}_{\text{bal}}$, which is now the objective stated in Eq. 9. Table 13 reproduces the deployed result after explicitly setting the inactive coefficient to zero. Removing balancing costs 0.75 FPR points. Replacing the hard value with the genuinely differentiable expected normalized distance $\sum_i [G_\theta(x)]_i D_i(x)$ makes performance much worse, so we report it as a negative ablation rather than adding it to the method.

Experts overlap because their layer subsets share the final anchor and often several intermediate layers. The

Table 14. **Expert-overlap-aware target.** We add λ_J times the Jaccard similarity to the best expert inside the top-32 target logits. Explicit overlap weighting does not improve the deployed target.

λ_J	0	0.5	1	2
FPR@95↓	**18.86**	19.93	19.79	19.59
AUROC↑	**94.49**	94.26	94.22	94.05

deployed target handles near-equivalence through performance: it distributes mass among the 32 experts with the best normalized boundary metric. Table 14 tests a more explicit hierarchy by adding the Jaccard similarity between each candidate subset and the best subset to the top-32 target logit. All nonzero strengths are slightly worse. Thus explicit overlap weighting is unnecessary in this experiment, although the shared structure remains useful for interpreting route stability (Table 15).

I. Few-shot Scaling of Voyager

Figure I.1 reports how Voyager scales with the number of ID shots per class used to train the router, with the WebContEX context held fixed. Performance is strongest at **1–2 shots** and degrades slightly as more ID data is added. The router is trained using ID-versus-context domain supervision to predict an expert-selection policy, so additional ID supervision can let it over-fit the boundary between ID and the WebContEX proxy, which transfers less well to the held-out evaluation shifts. This confirms that Voyager needs only minimal labeled ID data, consistent with the few-shot setting used throughout the paper, and mirrors the proxy-size behavior in Sec. C: a moderate amount of supervision is enough, and more is not always better. This diagnostic uses the extended-budget N=8 ensemble; the main paper reports the deployed N=5 detector at the standard 4-shot setting (Table 2, 18.86 FPR@95), so the absolute values should be read through the configuration map in Table 9.

Few-shot insight: A little ID data is enough. Voyager reaches its best ImageNet-1K FPR@95 with only 1–2 ID shots per class; adding more shots does not help and can mildly hurt, because the router over-fits the ID-vs-proxy boundary rather than the true OOD decision boundary.

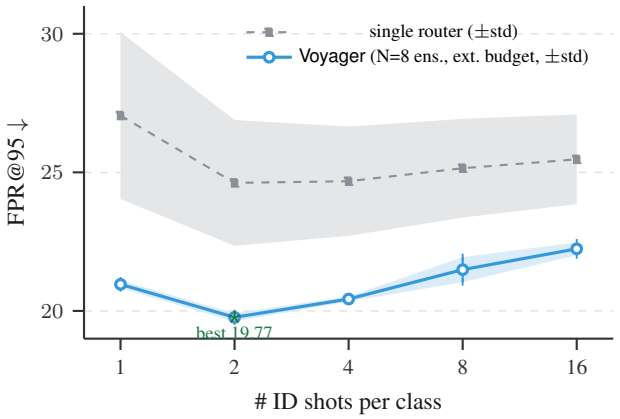


Figure I.1. **Few-shot scaling of Voyager (ViT-B/16, WebContEX).** ImageNet-1K FPR@95 vs. the number of ID shots per class. Gray: a single router (mean, with the ±std band over seeds); blue: the extended-budget N=8 ensemble (error bars/band are ±1 std over 3 independent ensembles; the deployed detector uses N=5). The ensemble both *lowers* FPR@95 by ∼4–6 points and *collapses* the seed variance (std ∼0.1–0.5 vs. ∼2–3 for a single router). Performance is best at only **1–2 shots**; more ID data lets the router over-fit the ID-vs-WebContEX boundary, which transfers worse to the evaluation shifts. Configuration: N=8 routers, top-32 target; ID shots vary along the x-axis.

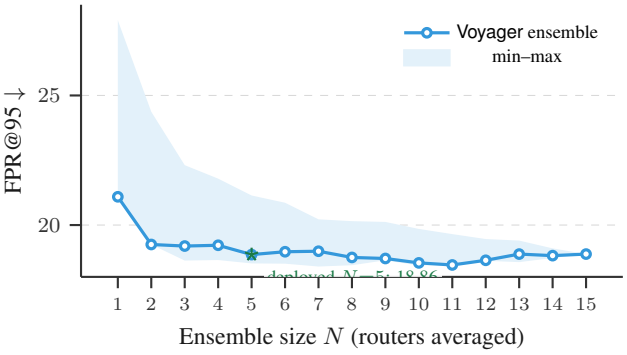


Figure J.1. **Ensemble-size variance reduction (ViT-B/16, 4-shot, top-32 target).** ImageNet-1K FPR@95 of the Voyager ensemble as routers are added (seeds $0..N-1$); shading is the min–max over random N-router subsets of a 15-router pool. Averaging both lowers FPR@95 and shrinks the seed-to-seed spread, with diminishing returns past $N\approx5$. The deployed N=5 detector (star) reaches **18.86**, matching the main results (Table 2).

J. Ensemble Size and Variance Reduction

Figure J.1 measures how score-level router averaging trades ensemble size for stability, holding the deployed recipe fixed (4-shot ID, top-32 target, WebContEX context). In the 15-seed audit pool, the seed changes both the cached 4-shot draw and router optimization randomness. A single router's FPR@95 ranges from 21.09 to 27.90 (std 1.7). Averaging the per-input scores of N routers shrinks this spread monotonically (std 0.5 at N=5, <0.2 by N=12) and lowers FPR, which plateaus past $N\approx5$. Five disjoint five-seed ensembles reach 19.91±0.79 FPR@95; the deployed seeds-0–4 ensemble, fixed without target-OOD selection, reaches 18.86 (Table 2). In a stricter control where all five routers share the identical seed-0 four-shot draw, averaging still improves 21.09 → 20.48 FPR. Thus optimization diversity helps by itself, while part of the full deployed gain comes from averaging across few-shot draws. The ensemble re-

Table 15. **Route stability across deployed seeds 0–4.** Values average all ten router pairs. Exact expert IDs vary, but selected layer sets overlap and their scores remain correlated.

Evaluation set	Exact route	Set Jaccard	Score corr.	Unique/seed
ImageNet-1K ID	0.061	0.590	0.694	617–785
iNaturalist	0.074	0.576	0.630	320–496
SUN	0.058	0.505	0.690	375–470
Places	0.052	0.468	0.696	280–402
DTD	0.073	0.490	0.737	288–412

mains inexpensive because the frozen backbone and per-layer scores are shared and only the small MLP heads are replicated. This is the mechanism behind deterministic test-time inference: the fixed five-router ensemble yields one scalar score per input.

Ensemble insight: Cheap averaging buys stability. Score-level averaging of a few routers converts the seed variance of a single gate into a stable, near-deterministic detector. Five routers capture most of the gain; the cost is negligible because only the tiny router heads are replicated over the shared frozen-CLIP score table.

The five deployed members individually span 21.09–25.54 FPR@95 and 92.93–93.96 AUROC, while their score average reaches 18.86/94.49. Exact expert-index agreement is only 5–7%, but this overstates instability in a highly overlapping dictionary: mean selected-set Jaccard is 0.47–0.59, and selected-score correlation is 0.63–0.74. Each router also uses hundreds of experts rather than collapsing to one. The members therefore learn distinct but overlapping routes, and score averaging converts that diversity into the performance gain.

K. Training Baselines under a Shared WebContEX Context

To separate routing from the supervision source, we retrain the prompt-learning baselines using the exact same weak OOD context as Voyager. Every method receives: (i) the same frozen CLIP ViT-B/16 backbone; (ii) the same 4-shot ImageNet-1K ID supervision; (iii) the same 5,078-entry WebContEX cache; and (iv) the same four-set evaluation protocol. Each method retains its native objective and score, so this is a controlled context comparison, not an assertion that the optimization problems are identical.

How each baseline consumes the WebContEX context. The prompt-learning baselines were not originally designed to ingest an arbitrary external outlier pool; each instead derives its negatives internally. We therefore adapt each method *minimally*, leaving its loss and architecture intact and substituting only the source of outliers:

- **LoCoOp** [30] normally mines OOD regions from the background of the few-shot ID images. We replace this internally mined OOD-region set with the WebContEX pool through its external-context branch (16 context tokens, 50 epochs, outlier weight λ=0.25, outlier batch size 4), keeping its global+local (GL-MCM) test-time scoring.
- **ID-Like** [2] normally *synthesizes* ID-like outliers via learnable crops of the ID images. We instead supply WebContEX images as the external negative set (4-shot ID, 3 epochs, ID batch size 32, outlier batch size 32). Its crop-based outlier construction makes each optimization step expensive, dominating its training cost.
- **Local-Prompt** [41] normally contrasts positive prompts against learned negative prompts; we attach the WebContEX pool as the negative-region context (16 context tokens, class-specific contexts, 30 epochs, 8 positive / 1 negative prompts, 300 negative-prompt bank). We reduce its batch size from 256 to 32 so the added outlier batch fits in memory.

We retain each published loss, loss coefficient, learning-rate schedule, and epoch count; only the explicit WebContEX input hook and the memory-required Local-Prompt batch-size change are made. These settings are taken from the native recipes and are not tuned on any target OOD set. The same-context results use one prescribed baseline run rather than selecting the best target-OOD seed; they should therefore be read as mechanism controls, while Table 2 remains the published-baseline comparison.

Cost-measurement protocol. All methods are measured with a single identical harness: end-to-end wall-clock training time (single seed), peak GPU memory (sampled from `nvidia-smi` at 3 s intervals), and peak system RAM (resident set of the training processes). Voyager is measured the same way, timing the training of its extended-budget N=8 router ensemble on the cached frozen features under the same WebContEX context (the deployed N=5 ensemble is cheaper still). This places every point of the efficiency–accuracy comparison on a common footing rather than mixing measured and reported costs.

Reproducibility notes. We record three practical considerations for exact reproduction. (i) The WebContEX pool is materialized as an `ImageFolder`-compatible union of the retrieval batches; the per-folder labels in the union are unused, since the pool serves only as an undifferentiated outlier set. (ii) For LoCoOp, the post-training ID-accuracy check in the underlying Dassl trainer expects a single logit tensor and is incompatible with LoCoOp's global+local tuple output; this does not affect the saved prompt checkpoint

or the OOD evaluation, which we run as a separate step. (iii) For Local-Prompt, the default batch size (256) does not fit on a 24 GB GPU once the WebContEX outlier batch is added; we reduce it to 32 (which still uses ∼22.9 GB) and enable expandable CUDA memory segments to avoid fragmentation. This changes only the batch size, not the optimization recipe.

For the native ImageNet-1K one-shot Local-Prompt row in Table 2, we use its released seed-1 split: exactly one image for each of 1,000 classes, 30 epochs, training top-50, and evaluation top-10, without WebContEX or target-OOD selection. Its effective batch of 256 also exceeds 24 GB because gradients traverse the frozen text transformer for 1,000 class-specific prompts. We preserve every native optimizer step by accumulating sample-count-weighted losses over two 128-image microbatches inside each of the eight prompt updates, and use activation checkpointing to recompute frozen text activations. Thus the objective and optimizer schedule are unchanged, up to floating-point reduction order; the split manifest, checkpoint/log hashes, and adapter are included with the artifacts.

For the separate ImageNet-100 comparison in Table 4, we materialize the exact 100 WNID classes as symlinks, verify that their sorted loader order equals the prompt-label order, and fit every prompt method on the same four seed-1 images per class. These are native runs without WebContEX: CoOp is implemented by setting LoCoOp's local-loss weight to zero, LoCoOp uses λ=0.25 and top-50 local patches, ID-Like uses its three-epoch recipe, and Local-Prompt uses 30 epochs and its 300-negative-prompt bank. No target-OOD image selects a checkpoint, score, or hyperparameter.

For the Pascal-VOC prompt comparison in Table 3, all four methods instead fit the complete 94-image validation split and evaluate the same disjoint 906-image test split and six OOD sets. We verify the same 14-class loader order in every implementation and retain each native recipe: CoOp and LoCoOp use 50 epochs, ID-Like uses 3, and Local-Prompt uses 30 with its 300-prompt negative bank. LoCoOp's released ImageNet local selection uses 200 of 1,000 classes; because 200 is invalid for 14 classes, we prescribe top-3, the nearest integer scaling (21%), before evaluation. Target OOD data are not used to choose this value, any checkpoint, or any score. The IN-22K loader uses the same deterministic first 10,000 sorted paths as the Voyager evaluation.

Training cost spans more than two orders of magnitude. Table 16 reports the measured cost of every method under identical supervision, and the spread is large. Training *time* ranges from ∼2.6 minutes for Voyager to ∼50.9 minutes (LoCoOp), ∼3.2 hours (ID-Like), and ∼7 hours (Local-Prompt)—a factor of roughly 160× between the cheapest and most expensive method. Peak *GPU memory* ranges from under 1 GB for Voyager to 13–23 GB for the prompt-learning baselines. Peak *system RAM* shows a similar gap (2.8 GB vs. 28–35 GB), because Voyager trains on a small cache of frozen per-layer scores while the prompt-learning methods stream the full ImageNet-1K training set through their data loaders.

Table 16. **Shared WebContEX training cost.** ViT-B/16, 4-shot ID, WebContEX (5,078 cache entries). Avg. FPR@95 is averaged over iNaturalist, SUN, Places, and Texture; the N=8 Voyager row is the shared-cost diagnostic, not the deployed Table 2 configuration.

Method	Train time	Peak GPU	Peak RAM	Avg. FPR@95↓
LoCoOp	50.9 min	18.5 GB	28.3 GB	36.67
ID-Like	∼3.2 h	13.1 GB	35.3 GB	53.60
Local-Prompt	7.1 h	22.9 GB	11.7 GB	31.13
Voyager (N=8)	**∼2.6 min**	**0.9 GB**	**2.8 GB**	**19.12**

The reason is structural: prompt learning optimizes *through* the frozen CLIP encoder, so every step must retain the activations needed to back-propagate the full image—and, for the methods studied here, several inner forward passes and an extra outlier batch. Voyager instead trains a small router on cached frozen scores and never runs a CLIP forward or backward pass during training. As a concrete illustration, Local-Prompt's default batch size of 256 does not fit on a 24 GB GPU at all once the WebContEX outlier batch is added; even reduced to 32 it occupies ∼22.9 GB (97% of the card) and takes about seven hours, whereas Voyager trains an extended-budget N=8 router ensemble in under three minutes using less than one gigabyte (the deployed detector uses N=5 and is cheaper still).

Result. The prompt adaptations do not outperform Voyager under the shared context. LoCoOp, ID-Like, and Local-Prompt reach 36.67, 53.60, and 31.13 FPR@95, whereas the shared-cost Voyager diagnostic reaches 19.12 with an extended-budget N=8 router ensemble. This is separate from the deployed Table 2 detector, which uses N=5 and reaches 18.86. The experiment shows that, given the same WebContEX pool, our routing formulation converts that context into a stronger detector than these adaptations. It does not show that access to external context is irrelevant: the proxy-source ablation establishes the opposite.

Appendix insight: Cheaper *and* better under the same context. Given the same retrieval-based WebContEX pool, Voyager is the strongest of the evaluated adaptations and the cheapest to train. This isolates how the methods use a common context; the separate proxy-source ablation measures the contribution of the context itself.

Table 17. **Prompt-view extensions (ViT-B/16).** Macro FPR@95↓/AUROC↑. These diagnostics are separate from the core detector tables. Panel (a) matches each LoCoOp-prompt route to its benchmark's standard route; panel (b) changes the semantic score by adding grouped negative labels.

Method	ImageNet-1K	ImageNet-100	Pascal-VOC
(a) Matched LoCoOp prompt view			
Standard-prompt Voyager	18.86/94.49	12.46/97.24	19.75/95.45
LoCoOp-prompt Voyager	42.43/86.78	13.61/97.11	14.89/96.42
Equal standard+LoCoOp	20.88/93.51	**8.38/98.13**	**13.35/97.18**
(b) Orthogonal CSP/negative-label extension			
CSP	17.35/95.82	6.49/98.54	15.42/96.75
CSP-prompt Voyager	14.73/96.26	3.87/98.77	14.53/96.64
Equal standard+CSP	**11.67/96.47**	**3.31/98.94**	**9.51/97.50**

L. Prompt-View Compatibility Check

DualPrompt is not part of Voyager and is not used for any main claim. We retain one compact compatibility table so that the revision does not hide the mixed result. The LoCoOp view uses native seed-1 prompts learned from ID data only; the CSP view uses the fixed grouped negative-label score [5, 18, 30]. Each view is routed independently with the same WebContEX pool and benchmark-specific standard router recipe.

Before equal fusion, each branch is standardized by its own clean-ID 5th-percentile boundary and standard deviation. The 50/50 weight is fixed without target-OOD tuning, and no target image or label is used for prompt fitting, routing, calibration, or weighting. Standard+LoCoOp is mixed: it regresses on ImageNet-1K but improves ImageNet-100 and Pascal-VOC. Standard+CSP improves these three far-OOD macro results, but it introduces a separate negative-label method and therefore remains an appendix compatibility result rather than evidence for the routing contribution.

M. Robustness Stress Tests

Near-OOD. Table 18 evaluates the discriminative ImageNet-10/ImageNet-20 split in both directions. Voyager lowers FPR@95 by 2.0 and 6.6 points relative to MCM, but AUROC is essentially unchanged in the first direction and decreases by 0.50 in the second. This supports a near-OOD FPR benefit without claiming uniform metric improvement.

Fixed hyperparameter transfer and boundary sensitivity. Table 19 separates two questions that the protocol-specific headline rows conflate. Carrying the complete ImageNet-1K configuration without retuning remains strong on ImageNet-100 (13.25/96.98) but loses 7.41 FPR points relative to the VOC-specific configuration. Thus the architecture and optimizer transfer, but the reported expert budget, target support, ensemble size, and proxy realization were benchmark-specific. The VOC 1-shot quantile sweep in Table 20 further varies only the empirical ID boundary. FPR spans 24.50–28.90 while AUROC changes by 1.33 points, confirming measurable sensitivity in a 14-image fitting pool. The full 94-image validation fit with the same fixed recipe reaches 24.03/95.44.

Table 18. **Near-OOD ImageNet subset stress test (ViT-B/16, 4-shot).** Each cell is FPR@95↓/AUROC↑. Both methods use global MCM scores in our common evaluation pipeline.

Method	IN-10 ID / IN-20 OOD	IN-20 ID / IN-10 OOD
MCM	6.80/98.90	24.80/97.22
Voyager	**4.80**/98.88	**18.20**/96.72

Table 19. **Benchmark-specific versus fixed router recipe.** The fixed recipe transfers ImageNet-1K's s=9, top-32, N=5 settings without target-OOD tuning.

ID benchmark	Protocol-specific	Fixed recipe
	FPR@95↓/AUROC↑	
ImageNet-1K	18.86/94.49	18.86/94.49
ImageNet-100	**12.46/97.24**	13.25/96.98
Pascal-VOC	**16.62/96.41**	24.03/95.44

Table 20. **VOC 1-shot boundary sensitivity.** The fixed ImageNet-1K recipe is used throughout; each cell is macro FPR@95/AUROC over six shifts.

Q	0.01	0.025	0.05	0.10	0.20
FPR@95↓	26.42	**24.50**	25.93	25.20	28.90
AUROC↑	**94.85**	94.67	94.51	94.22	93.52

Table 21. **Classical post-hoc scores on frozen ViT-B/16 features.** No method sees OOD fitting data; kNN and Mahalanobis use the same four ID shots. Values are four-shift macro averages.

Score	FPR@95↓	AUROC↑
Maximum logit	67.04	84.59
MSP (T=0.01)	60.01	82.94
Negative entropy (T=0.01)	**48.95**	**87.53**
Energy (T=0.01)	77.77	81.24
4-shot kNN	97.59	67.01
4-shot Mahalanobis	98.87	67.99

Traditional post-hoc scores. Table 21 broadens the main frozen-VLM comparison with maximum-logit, MSP, energy, entropy, kNN, and shrinkage-Mahalanobis scores on the same final-layer features. These conventional scores do not transfer well to the frozen CLIP four-shift protocol; the strongest shown is negative entropy at 48.95/87.53. We report this result for scope rather than treating methods designed around trained discriminative classifiers as directly

matched competitors.

N. Complementarity Between Experts

We analyze the complementarity between experts generated at different sparsity levels, using ImageNet-1K as the in-distribution dataset and SUN, Places, iNaturalist, and Textures as OOD benchmarks. The **complementarity score** between two experts is defined as:

$$C = \frac{\text{Unique Detections}}{\text{Total Detections}} = \frac{\text{only}_1 + \text{only}_2}{\text{only}_1 + \text{only}_2 + \text{both}}, \quad (11)$$

where only_1 and only_2 denote samples detected exclusively by one expert, and both denotes samples detected by both. A value of $C = 0$ indicates identical behavior, whereas $C = 1$ corresponds to fully complementary experts.

To characterize expert diversity, we evaluate complementarity under a **retrospective per-instance selector**, where the best expert is chosen independently for each instance using labels that are unavailable at deployment. This is an analysis diagnostic rather than a practical detector. Figure N.1 reports the pairwise complementarity matrices across sparsity levels for four CLIP backbones.

Several consistent trends emerge. First, experts associated with *distant sparsity levels* exhibit the strongest complementarity. For example, ViT-B/16 and ViT-B/32 reach complementarity scores up to $C \approx 0.26$ between the most distant sparsity configurations, indicating that these experts frequently detect different subsets of OOD samples. In contrast, nearby sparsity levels show substantially lower complementarity (e.g., $C \approx 0.05$–0.10), suggesting partial redundancy.

Second, transformer backbones exhibit substantially richer complementarity patterns than convolutional ones. ViT-B/16 and ViT-B/32 display the highest diversity across experts, while ViT-L/14 shows slightly reduced complementarity (maximum $C \approx 0.21$), suggesting that deeper transformer models produce somewhat more homogeneous expert behavior. In contrast, RN50x4 exhibits consistently low complementarity (typically $C \leq 0.07$), indicating that experts derived from different sparsity levels capture more similar signals.

RN50x4 feature alignment. Convolutional stages also differ in channel count and spatial resolution, so mapping them into CLIP's shared embedding space can itself discard information. We isolate this factor with the same score, router recipe, WebContEX context, and eight seeds, changing only the stage-alignment operator. CLIP's pretrained residual downsampling reaches $45.40/88.82$; propagating through the later transition blocks gives $51.51/87.21$, while frozen, untrained 1×1 projections degrade to $56.66/84.81$. Thus naive random alignment is harmful. The RN50x4 result in the main paper uses the pretrained alignment path, so its remaining limitation is consistent with the low complementarity above rather than an untrained projection artifact. More generally, convolutional encoders may require architecture-aware experts that preserve multiscale stage structure instead of directly transplanting the Transformer subset construction.

Retrospective sparsity sweep. Figure N.2 reports the main sparsity diagnostic deferred from the paper body. Increasing the sparsity budget enlarges the candidate expert family and consistently lowers retrospective FPR@95 relative to the last-layer baseline, confirming that useful intermediate evidence is not exhausted by a single fixed subset.

O. Theoretical Justification: The Value of Intermediate Layers

This section is intended as a lightweight motivation for dynamic routing, not as a formal guarantee of FPR@95. The gap Δ below is an expected-score separation proxy that explains when adding an intermediate layer can improve an anchored expert; the empirical sections measure the actual thresholded OOD behavior.

Layer-wise Confidence and Separation. Let $f_\ell(x)$ be the normalized representation at layer $\ell \in \{0, \dots, L-1\}$ of a frozen CLIP-style encoder and $\{t_c\}_{c=1}^{C}$ be text prototypes. We define the maximum-concept matching (MCM) confidence [28] at temperature $\tau > 0$ as:

$$S_\ell(x;\tau) = \max_c \frac{\exp(\langle f_\ell(x), t_c\rangle/\tau)}{\sum_{c'} \exp(\langle f_\ell(x), t_{c'}\rangle/\tau)}. \quad (12)$$

To quantify OOD detection performance, we define the *separation contribution* of layer ℓ as the expected difference between in-distribution ($\mathbb{P}_{\text{ID}}$) and OOD ($\mathbb{P}_{\text{OOD}}$) scores:

$$\Delta(\ell) = \mathbb{E}_{x\sim\mathbb{P}_{\text{ID}}}[S_\ell(x;\tau)] - \mathbb{E}_{x\sim\mathbb{P}_{\text{OOD}}}[S_\ell(x;\tau)]. \quad (13)$$

Anchored Expert Scores. We anchor our selection to the final layer $\ell^\star = L-1$, which contains the strongest semantic signal. For any auxiliary subset $A \subseteq \{0, \dots, L-2\}$ where $|A| \leq s-1$, the anchored expert score and its corresponding population separation gap $\Delta(A)$ are:

$$S_A^{\text{anch}}(x;\tau) = \frac{S_{\ell^\star} + \sum_{\ell\in A} S_\ell}{1+|A|}, \quad \Delta(A) = \frac{\Delta(\ell^\star) + \sum_{\ell\in A}\Delta(\ell)}{1+|A|}. \quad (14)$$

While a *static* method fixes A for all x, our *dynamic* approach selects $A(x)$ per-input. Larger $\Delta(A)$ implies better threshold separability for OOD detection.

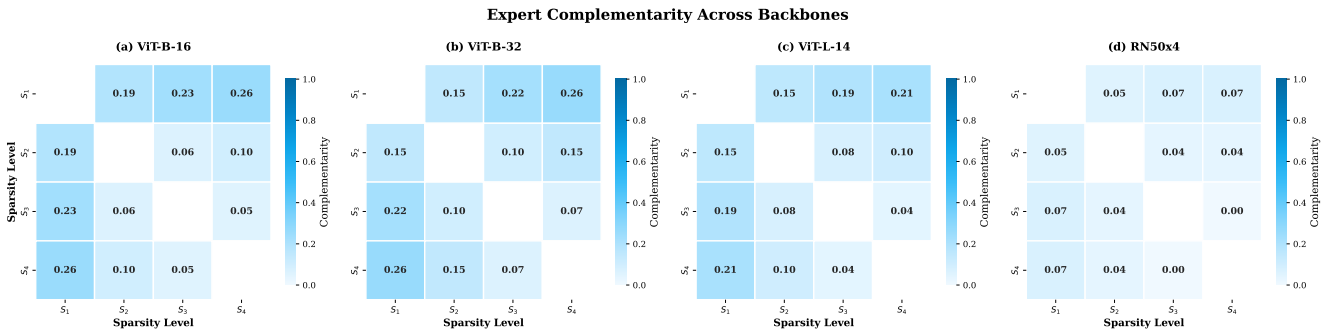


Figure N.1. **Expert complementarity across CLIP backbones.** Pairwise complementarity matrices for ViT-B/16, ViT-B/32, ViT-L/14, and RN50x4. Transformer experts expose richer diversity, whereas RN50x4 remains comparatively redundant.

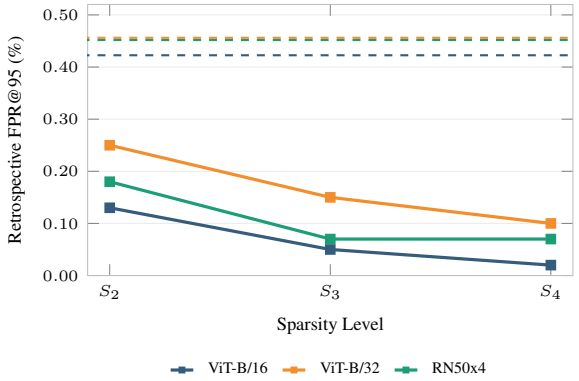


Figure N.2. **Sparsity diagnostic.** Comparison across architectures. Dashed lines: last-layer reference; solid lines: retrospective per-instance selection, used only to measure routing opportunity.

Theorem 1 (Exact Improvement Condition). *For any $A \subseteq \{0, \ldots, L-2\}$ and $\ell \notin A$, adding layer ℓ improves the separation gap if and only if its individual contribution exceeds the current anchored average:*

$$\Delta(A \cup \{\ell\}) > \Delta(A) \iff \Delta(\ell) > \Delta(A). \tag{15}$$

Proof. Let $B = \Delta(\ell^\star) + \sum_{j \in A} \Delta(j)$. By (14), we have $\Delta(A) = \frac{B}{1+|A|}$ and $\Delta(A \cup \{\ell\}) = \frac{B+\Delta(\ell)}{2+|A|}$. Then:

$$\begin{aligned} \frac{B+\Delta(\ell)}{2+|A|} > \frac{B}{1+|A|} &\iff (1+|A|)(B+\Delta(\ell)) > B(2+|A|) \\ &\iff (1+|A|)\Delta(\ell) > B \\ &\iff \Delta(\ell) > \frac{B}{1+|A|} = \Delta(A). \end{aligned}$$

This completes the proof. □

Implications: why intermediate layers help. Equation (15) formalizes the key mechanism: intermediate layers are beneficial *only* when they provide higher ID–OOD separation than the current anchored expert. Under distribution shift, some intermediate depths can increase separability (large $\Delta(\ell)$) while others can be neutral or harmful (small or negative $\Delta(\ell)$). Consequently, any *static* subset A inevitably underperforms on shifts where its chosen layers do not satisfy (15). Dynamic routing addresses this by selecting, per input, auxiliary layers whose contributions are predicted to be beneficial using minimal outlier context.

Combinatorial Sparsity Motivation and a Theoretical Plot Equation (4) shows that selecting up to s layers from L induces a combinatorial hypothesis class of size $K = \sum_{r=1}^{s} \binom{L-1}{r-1}$, even under the mandatory inclusion constraint of the last layer. This motivates dynamic routing as an *amortized* selection mechanism: a static method must commit to a single subset (or a fixed fusion rule) globally, whereas routing learns a mapping $x \mapsto \widehat{i}(x)$ that selects among many sparse candidates without exhaustive search.

P. Standalone Voyager Layer-Selection Validity Check

Main Fig. 10 audits the deployed standalone GL-MCM Voyager with all ImageNet-1K validation and original-suite OOD images and all five router seeds. For auxiliary block ℓ and image x, its inclusion rate is

$$r_\ell(x) = \frac{1}{5} \sum_{n=1}^{5} \mathbf{1}[\ell \in A_n(x)], \tag{16}$$

the fraction of deployed routers whose selected expert contains that block. Layer 12 is mandatory and omitted. Aggregating r_ℓ uses all 50,000 ID and 35,640 OOD images. The size-matched view subtracts, separately for each router and dataset, its mean inclusion across the 11 exchangeable auxiliary blocks. A positive value therefore means that a block is selected more than expected for that router's observed route size. Orange outlines require a positive sign for every seed, a stricter stability condition than a large pooled mean.

As a separate validity check, we compute post-hoc sparsity-constrained Shapley values for the same fixed recipe (ViT-B/16, WebContEX, 4-shot ID, s=9, top-32,

Table 22. **Do routed layers match measured layer value?** Router inclusion is averaged over seeds 0–4; Shapley blocks maximize marginal ID–OOD separation. Blocks are one-indexed and exclude the mandatory final block 12. Agreement is Spearman correlation across the 11 auxiliary blocks.

Shift	Visual cue	Most routed blocks (rate)	Largest Shapley blocks (ϕ)	ρ
DTD	texture	1 (0.49), 2 (0.41), 11 (0.32)	5 (0.088), 3 (0.070), 6 (0.037)	−0.10
iNaturalist	fine-grained natural	1 (0.66), 8 (0.59), 10 (0.56)	8 (0.049), 10 (0.042), 1 (0.032)	0.86
Places	scene/layout	1 (0.57), 11 (0.39), 6 (0.34)	3 (0.040), 5 (0.036), 4 (0.010)	−0.44
SUN	scene/layout	1 (0.53), 10 (0.38), 11 (0.38)	3 (0.036), 7 (0.008), 2 (0.005)	−0.24

seeds 0–4). Shapley estimation uses a fixed seed-2027 subsample of 400 ID and 400 images per OOD dataset; the full-set route frequencies remain independent of this approximation. The final layer is held as the mandatory anchor. Equation (16) measures what the router selects, whereas Shapley values estimate each block's marginal mean-separation contribution. Neither diagnostic is used for model selection.

Selection is not causal attribution. The full-set distribution explains the learned routing policy. Correlated, overlapping experts can make a block frequent without making its individual marginal contribution positive.

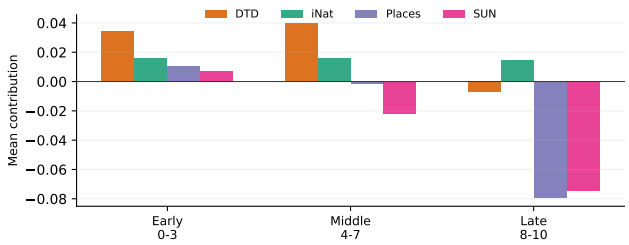


Figure P.1. **Intermediate-layer contribution by OOD family.** Bars report the mean Shapley contribution of early, middle, and late intermediate layers to the Table 2 Voyager separation objective. Positive values indicate improved ID–OOD separation relative to the final-layer anchor; negative values indicate harmful auxiliary evidence.

Table 22 directly contrasts full-set route frequency with the approximate separation objective. The fine-grained iNaturalist shift gives the clearest correspondence: blocks 1, 8, and 10 are among both the most frequently routed and most valuable. DTD and the two scene shifts show the limitation: their routed experts vary among correlated subsets, so frequent blocks need not have the largest individual Shapley value. Together with the low exact-route agreement but moderate set overlap in Table 15, this supports depth-group and distribution-level explanations rather than a literal causal label for one selected expert.

Q. Interpretability: Full-Set and Per-Image Routes

Main Fig. 10 reports the aggregate depth policy of standalone Voyager. Figure Q.1 expands that audit with image-level routes, using the same five GL-MCM routers, ViT-B/16 backbone, WebContEX context, and sparsity budget s=9. For image x, each plotted ordinate is the inclusion rate $r_\ell(x)$ from Eq. (16): the fraction of the five routers whose selected expert contains auxiliary block ℓ. Hence values occur in increments of 0.2, and a value of one denotes unanimous inclusion. Block 12 is mandatory and omitted.

The top row exhausts each original-suite OOD dataset; the remaining rows use three examples per dataset drawn uniformly without replacement from the frozen dataset order with seed 2027, before inspecting routes or predictions. This selection is target-blind and is not used for model selection. The page therefore separates population-level routing tendencies from illustrative per-image variation without introducing the dual-prompt or CSP branch.

R. Limitations

Voyager is a few-shot detector that depends on both labeled ID examples and proxy OOD context. WebContEX is automatic, but it is still external data: replacing it with ID-derived local context degrades average FPR@95 from 18.86 to 26.71. The same-context selector ablation separately shows that learned routing contributes beyond the proxy, improving from 28.9 for the strongest score rule to 18.8. These controls support complementary roles for data and routing, not a data-free explanation.

The gains are taxonomy- and shift-dependent. On Pascal-VOC, native LoCoOp reaches 14.23/96.67 and remains stronger than standalone Voyager at 16.62/96.41. Near-OOD FPR improves, but one direction loses 0.50 AUROC, and the VOC quantile experiment shows that empirical boundaries are noisy in extremely small ID pools. The reported model evaluation is confined to CLIP backbones; transfer to other VLM families and specialized ID domains remains to be established. Future work should study domain-adaptive proxies, shrinkage-based boundaries, hierarchical expert sharing, and proxy-free routing.

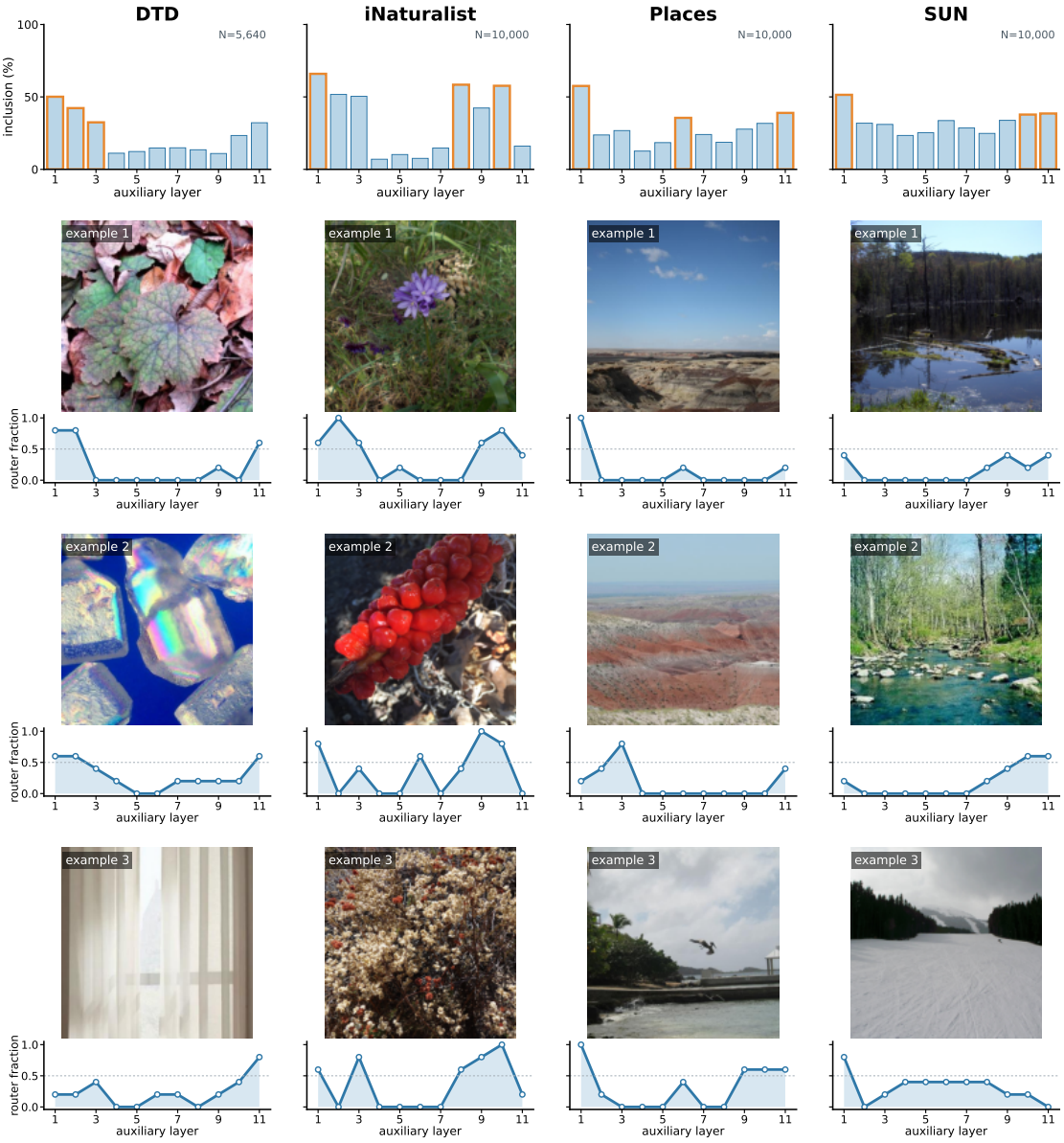


Figure Q.1. **Standalone Voyager route distributions.** The top row is the mean auxiliary-block inclusion rate over every image and all five routers; orange outlines mark the three most frequently included blocks. Each example curve gives the fraction of the same five routers selecting each block for one target-blind image. Block 12 is mandatory and omitted.

Dataset-level structure. DTD concentrates on early blocks 1–3 (50%, 42%, 32%) while retaining block 11 (32%). iNaturalist combines early block 1 (66%) with late blocks 8 and 10 (58% each). Places and SUN both favor block 1 (58% and 51%) but preserve more middle/late evidence: Places next emphasizes blocks 11 and 6, whereas SUN emphasizes 11 and 10. Thus the complete-data profiles reproduce Main Fig. 10: texture, fine-grained natural, and scene shifts induce distinct depth mixtures rather than one universal subset.

Per-image variation and scope. Aggregate peaks do not imply a fixed route. DTD examples 1–2 favor blocks 1–2, while example 3 shifts toward block 11. Places ranges from near-exclusive block-1 agreement to block-3 emphasis and a mixed early/late route; SUN includes both late-heavy and broad middle-depth cases. iNaturalist repeatedly uses early and late blocks, but their relative weights change by image. These curves describe the router policy, not why an object is OOD: correlated experts can realize similar scores with different subsets, and inclusion frequency is neither semantic segmentation nor causal layer attribution.