VEFR: VISUAL EVIDENCE FUSION ROUTING FOR BUDGETED SKILL SELECTION IN FINE-GRAINED VISUAL RECOGNITION WITH MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained recognition requires distinguishing subtle visual differences, an MLLM-based system can approach the same image through several recognition strategies. These strategies differ in cost and have complementary strengths across classes and individual images. We formulate recognition as budgeted skill routing: selecting an executable recognition strategy before observing its output or incurring its execution cost. We propose Visual Evidence Fusion Routing (VEFR), which estimates skill competence from two sources of visual evidence. Class-conditioned statistics capture recurring skill preferences, while correctness records from visually similar profiling images provide query-specific evidence. A soft visual class distribution weights the class statistics, and a validation-selected fixed weight combines the two estimates without training a separate routing network. We separate competence estimation from deployment control, allowing different prioritization of accuracy–cost without recalculating the evidence. Across eight image recognition benchmarks, VEFR achieves better accuracy compared with fixed skill and skill routing methods while consuming much fewer MLLM tokens. These results show that visual evidence can guide skill selection to improve the accuracy–cost trade-off.

1 INTRODUCTION

Fine-grained visual recognition distinguishes closely related categories, such as bird species, car models, and aircraft variants (Wah et al., 2011; Krause et al., 2013; Maji et al., 2013). Recognition often depends on small differences in plumage, shape, or local markings. Meanwhile, pose, viewpoint, and background can vary substantially within the same category. These subtle distinctions remain challenging for multimodal large language models (MLLMs), despite their broad visual knowledge (Kim & Ji, 2024).

An MLLM-based recognition system can approach the same image in several ways, e.g. visual retrieval, direct prediction, category-description prompting, explicit reasoning, and visual in-context learning (VICL) (Zhang et al., 2022; Conti et al., 2025; Yang et al., 2026; Xiong et al., 2026). We call these executable procedures *recognition skills*. They draw on different sources of evidence and incur different inference costs. Applying one skill to every image overlooks their complementary strengths, while executing all skills can incur unnecessary cost. This raises a question beyond what label to predict: *which skill should be used for image be recognized?*

The answer can vary across both classes and individual images. Figure 1(b) shows that the most accurate skill differs across selected bird classes. Figure 1(c) further illustrates that images from the same class can have different successful skills. The gap between the strongest fixed skill and a hindsight oracle provides broader evidence of this complementarity (Table 1). For some images that the strongest fixed skill misclassifies, another skill already produces the correct answer. The challenge is to predict which skill is likely to succeed before observing its output or incurring its execution cost.

Existing routing methods provide useful foundations for this problem. Concurrent SkillOrchestra (Wang et al., 2026) infers skill demands from language-agent interaction states and uses a learned competence handbook for multi-turn orchestration. RouteLLM (Ong et al., 2025) learns routing

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

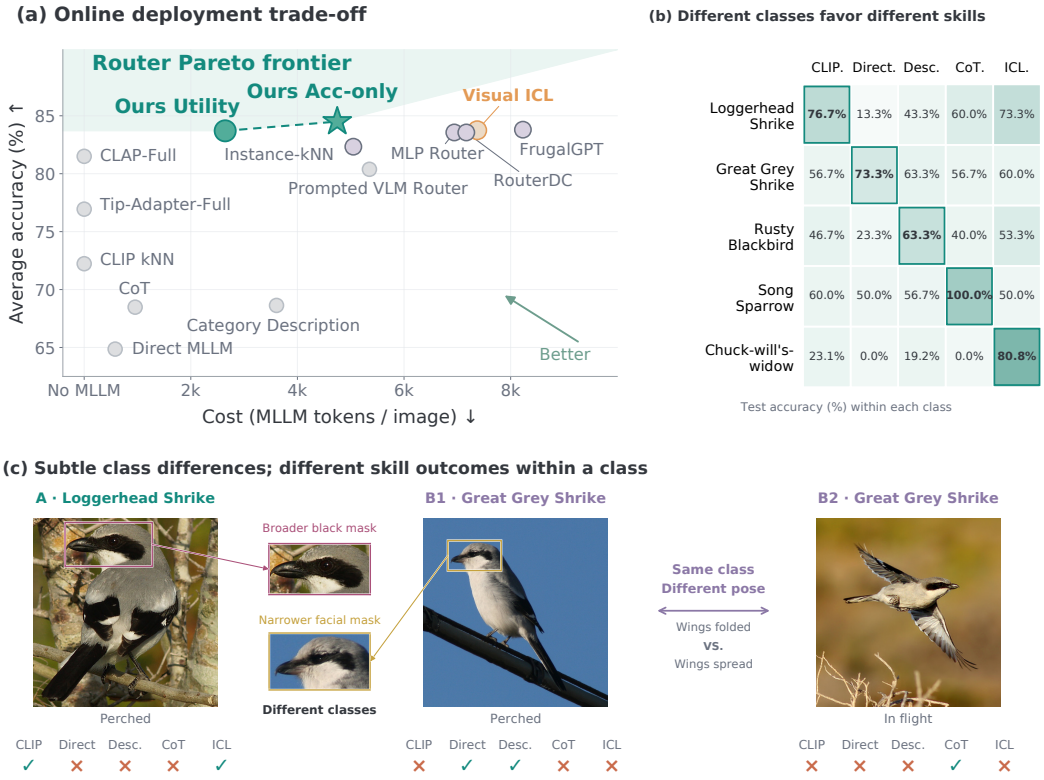


Figure 1: **Skill complementarity and accuracy–cost trade-offs.** (a) Accuracy versus MLLM token usage; the dashed line connects non-dominated evaluated routing configurations. (b) Per-class skill accuracy; shading shows deviation from the class mean and outlines mark the best skill. (c) Subtle between-class differences and within-class variation in skill outcomes (✓: correct; ×: incorrect). Examples do not establish a causal effect of pose.

models from preference data, while RouterDC (Chen et al., 2024c) trains a query encoder and model embeddings through contrastive learning. These language-oriented formulations do not directly model the visual cues that affect recognition-skill success. For fine-grained recognition, identical textual instructions can accompany images that benefit from different skills. Routing therefore requires evidence that distinguishes individual images. We investigate whether visual similarity and cached recognition outcomes can provide this evidence without training a separate routing network.

We propose *Visual Evidence Fusion Routing (VEFR)*, which estimates skill competence by combining class-conditioned and local visual evidence. Class-conditioned profiling statistics capture recurring skill preferences. A soft visual class distribution weights these statistics, avoiding commitment to a single predicted category. Correctness records from visually similar profiling images provide local evidence that can capture variation within a class. We combine the two competence estimates for skill selection. The resulting *Fixed Hybrid* estimator uses cached outcomes directly and requires no separate routing network.

Realistic deployment requires different accuracy–cost trade-offs. VEFR therefore separates competence estimation from skill selection. The same frozen competence scores support accuracy-first selection, per-query utility optimization, and batch-budget allocation. The cost-aware policies use mean skill costs from offline profiling to penalize expensive skills or allocate them jointly under an estimated token budget. We evaluate realized costs separately. Changing the deployment objective or budget requires no retraining of the competence estimator.

Across eight image recognition benchmarks, VEFR improves the accuracy–cost trade-off, as shown in Figure 1(a). Accuracy-first selection achieves 84.50% macro-average accuracy, exceeding VICL, the strongest fixed skill, by 0.75 percentage points while using 35.6% fewer MLLM tokens. Utility-based selection further reduces token usage by 64.1% relative to VICL while maintaining nearly the same accuracy (83.70% vs. 83.75%).

Our contributions are threefold:

- We formulate fine-grained MLLM recognition as budgeted routing over heterogeneous executable recognition skills, selecting a skill for each image before execution.
- We propose VEFR, which combines class-conditioned skill statistics with correctness records from visually similar profiling images to estimate competence without training a separate routing network.
- We separate competence estimation from deployment control. One frozen estimator supports accuracy-first selection, per-query utility optimization, and batch-budget allocation, yielding improved accuracy–cost trade-offs across eight benchmarks.

2 Related Work

Fine-Grained Visual Recognition: Fine-grained visual recognition (FGVR) distinguishes subordinate categories such as bird species Wah et al. (2011), car models Krause et al. (2013), aircraft variants Maji et al. (2013), and animal breeds Parkhi et al. (2012). Subtle inter-class differences and variation in pose, viewpoint, scale, illumination, and background make it challenging Nilsback & Zisserman (2008); Xiao et al. (2010); Hou et al. (2017); Van Horn et al. (2021). Part-based R-CNN Zhang et al. (2014) models parts explicitly; RA-CNN Fu et al. (2017), MA-CNN Zheng et al. (2017), and NTS-Net Yang et al. (2018) discover informative regions with weaker supervision. Bilinear CNNs Lin et al. (2015), PMG Du et al. (2020), and TransFG He et al. (2022) use high-order interactions, multi-granularity features, and discriminative transformer tokens, respectively. Their fixed classifiers and inference pipelines, however, limit adaptation to individual images.

VLMs and MLLMs for Fine-Grained Recognition: CLIP Radford et al. (2021) enables open-vocabulary recognition through image–text alignment. Adaptation methods learn prompts or use few-shot visual features Zhou et al. (2022b;a); Khattak et al. (2023); Zhang et al. (2022). FGVR-specific methods add textual supervision, LLM-discovered attributes, region-grounded descriptions, or class-aware prompt weights Kim & Ji (2024); Chiquier et al. (2024); Halbe et al. (2026); Dong et al. (2026). MLLMs such as BLIP-2 Li et al. (2023), LLaVA Liu et al. (2023), InstructBLIP Dai et al. (2023), and MiniGPT-4 Zhu et al. (2024) support recognition through prompts, descriptions, and reasoning, but modality gaps, weak category alignment, and unreliable fine-grained reasoning persist Kim & Ji (2024); He et al. (2025; 2026). Other methods use VQA, few-shot reasoning, candidate filtering, or answer extraction Atabuzzaman et al. (2025); Liu et al. (2025); Wei (2024); Lawrence et al. (2026). Most apply a fixed strategy across queries or execute several at added token cost. We instead route each image to a recognition skill based on accuracy and cost.

Retrieval- and Agent-Augmented Vision-Language Reasoning: Retrieval provides knowledge, demonstrations, and multimodal evidence in KAT Gui et al. (2022), PICa Yang et al. (2022), and related methods Chen et al. (2022); Hu et al. (2023b;a). Visual in-context learning (ICL) improves demonstrations through counterfactual retrieval, coreset optimization, or compressed task representations Xiong et al. (2026); Chen et al. (2025); Huang et al. (2024). Multiple image–text examples increase token consumption, while multimodal task vectors require model-internal access. Agent- and tool-augmented systems decompose tasks and invoke external models, APIs, or programs Wu et al. (2023); Yang et al. (2023); Surís et al. (2023); Lu et al. (2023); Liu et al. (2024). GEPA Agrawal et al. (2026), ACE Zhang et al. (2026), APO-Pair Goncalves et al. (2026), and AutoSEP Hong et al. (2026) optimize prompts or context through feedback. For FGVR, SCAN Yang et al. (2026) uses System-2-style attribute reasoning, while ReFine-RFT Zhu et al. (2026) controls reasoning length. Such methods may require iterative updates, planning, or repeated calls. Our router uses class-level accuracy and cached local competence to choose among ICL, prompting, description generation, and reasoning, reserving costly skills for images likely to benefit.

Skill-Aware Model and Agent Routing: Concurrent SkillOrchestra Wang et al. (2026) combines a reusable language-skill handbook with inferred demand, agent competence, and cost for multi-turn orchestration. We address black-box MLLM fine-grained recognition by estimating competence for fixed procedures from a visual class posterior and cached neighbor correctness, separately from online utility and exact batch-budget control. Transfer protocols also differ: SkillOrchestra reuses its handbook across orchestrators; RouteLLM Ong et al. (2025) retains its router while replacing the model pair; and GraphRouter Feng et al. (2025) adds models through limited interactions. Our

executor-transfer study separates reuse of frozen visual competence evidence from adaptation with explicitly budgeted target outcomes.

3 VISUAL EVIDENCE FUSION ROUTING

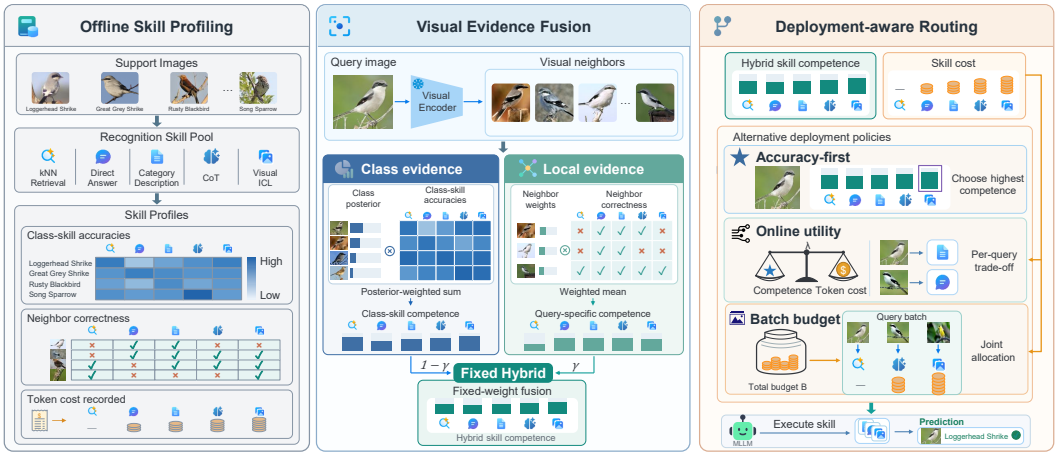


Figure 2: **Visual Evidence Fusion Routing (VEFR).** Offline profiling records each skill's correctness and cost. Darker blue indicates higher class–skill accuracy. At inference, fixed-weight fusion combines class-conditioned and local competence estimates. The same frozen estimator supports accuracy-first selection, per-query utility, and batch-budget allocation.

3.1 RECOGNITION AS BUDGETED SKILL SELECTION

Problem formulation. We study fine-grained recognition as a skill selection problem. Given a query image x_i, a fine-grained label space $\mathcal{C}$, and a set of recognition skills $\mathcal{S}$, a router selects one skill before execution. Each skill $s \in \mathcal{S}$ provides an alternative recognition procedure for predicting the fine-grained category in $\mathcal{C}$ and incurs a realized execution cost $C_{i,s}$. Let $y_{i,s} \in \{0, 1\}$ indicate whether the prediction from skill s is correct. The ideal but unknown query-specific competence of skill s is defined as

$$\pi_{i,s} = \Pr(y_{i,s} = 1 \mid x_i). \tag{1}$$

The router estimates this quantity and selects a skill according to the deployment objective. We estimate competence from a labeled training subset, referred to as the *profiling set*, which records skill outcomes and execution costs on representative images. These records provide the basis for constructing class-level and local evidence for skill selection.

Offline skill profiling. We execute each skill offline on the profiling images and record its correctness and realized cost. The resulting profiling set is

$$\mathcal{P} = \{(x_j, c_j, \mathbf{y}_j, \mathbf{C}_j)\}_{j=1}^{N_\mathcal{P}}, \qquad N_\mathcal{P} = |\mathcal{P}|, \tag{2}$$

where $c_j \in \mathcal{C}$ is the ground-truth class of x_j. The vectors $\mathbf{y}_j = (y_{j,s})_{s\in\mathcal{S}}$ and $\mathbf{C}_j = (C_{j,s})_{s\in\mathcal{S}}$ contain the observed correctness ($y_{j,s} \in \{0, 1\}$) and realized cost of each skill, respectively.

For class $c \in \mathcal{C}$, let $\mathcal{P}_c = \{j \in \{1, \ldots, N_\mathcal{P}\} : c_j = c\}$ denote its profiling subset, and let $n_c = |\mathcal{P}_c|$. For $n_c > 0$, the empirical accuracy of skill s on class c is

$$a_{c,s} = \frac{1}{n_c} \sum_{j\in\mathcal{P}_c} y_{j,s}. \tag{3}$$

These class–skill accuracies capture recurring differences in skill competence across classes. They provide the class-conditioned evidence used for skill selection.

Regularization with limited profiling data. The empirical class–skill accuracy $a_{c,s}$ can be noisy when only few profiling examples are available for class c. We therefore shrink it toward the global skill prior g_s estimated from the entire profiling set:

$$\widetilde{a}_{c,s} = \frac{\sum_{j \in \mathcal{P}_c} y_{j,s} + \tau g_s}{n_c + \tau}, \qquad g_s = \frac{1}{N_{\mathcal{P}}} \sum_{j=1}^{N_{\mathcal{P}}} y_{j,s}. \tag{4}$$

Here $\tau \geq 0$ controls the shrinkage strength and represents the number of equivalent profiling examples from the global prior. Setting $\tau = 0$ recovers the empirical accuracy. The resulting $\widetilde{a}_{c,s}$ is used as the regularized class–skill accuracy statistic in the class-conditioned estimate.

3.2 Competence Estimation from Visual Evidence

We estimate skill competence from two complementary forms of visual evidence. The profiling set provides empirical observations of these latent competence values through skill outcomes on representative images. Class-conditioned evidence captures recurring skill preferences across likely classes, while local visual evidence captures query-specific variation from visually similar profiling examples. Figure 2 summarizes the framework.

For each query x_i, a low-cost vision–language model, such as CLIP, first selects a nonempty candidate class set $\mathcal{Q}_i \subseteq \mathcal{C}$. We then retrieve the K nearest profiling images using a frozen image encoder $h(\cdot)$. Let $\mathcal{N}_K(x_i)$ denote the indices of the K nearest profiling neighbors of x_i. We measure similarity by $r_{ij} = \cos(h(x_i), h(x_j))$ and assign each neighbor a positive weight $w_{ij} = \exp(r_{ij}/T)$, where $T > 0$ controls the sharpness of neighbor weighting. When comparing image encoders, we keep the candidate class set fixed. Only neighbor retrieval and similarity weights depend on h.

Class-conditioned evidence. We use the labels of retrieved neighbors to construct a soft distribution over the candidate classes:

$$p(c \mid x_i) = \frac{\mathbb{1}[c \in \mathcal{Q}_i] \sum_{j \in \mathcal{N}_K(x_i)} w_{ij} \mathbb{1}[c_j = c]}{\sum_{j \in \mathcal{N}_K(x_i)} w_{ij} \mathbb{1}[c_j \in \mathcal{Q}_i]}. \tag{5}$$

$\mathbb{1}[\cdot]$ is the indicator function. If no retrieved neighbor belongs to $\mathcal{Q}_i$, we use a uniform distribution over $\mathcal{Q}_i$ and assign zero probability to all other classes. The above distribution only gives non-zero probabilities to candidate classes $\mathcal{Q}_i$.

The class-conditioned competence estimate is

$$\widehat{A}^{\text{cls}}_{i,s} = \sum_{c \in \mathcal{C}} p(c \mid x_i) \widetilde{a}_{c,s}. \tag{6}$$

This estimate averages the class–skill accuracies using the query's soft class distribution. It avoids committing to a single predicted class. It provides an approximation of expected skill success under a class-conditioned competence assumption, where images from the same class share similar skill preferences. Pooling outcomes within each class reduces sensitivity to individual correctness records. However, it does not capture within-class variation, and the resulting competence estimate is therefore not perfectly calibrated. Local evidence relaxes this assumption by capturing within-class variation.

Local visual evidence. Images from the same class can differ in pose, viewpoint, and visible discriminative features. Class-level statistics may overlook such within-class variations. We therefore use correctness records from visually similar profiling images to construct a query-specific competence estimate:

$$\widehat{A}^{\text{loc}}_{i,s} = \frac{\sum_{j \in \mathcal{N}_K(x_i)} w_{ij} y_{j,s}}{\sum_{j \in \mathcal{N}_K(x_i)} w_{ij}}. \tag{7}$$

This estimate shares the same retrieved neighborhood and similarity weights as the class-conditioned evidence, but aggregates skill correctness directly rather than through class statistics. It averages correctness directly over all retrieved neighbors, without restricting them to $\mathcal{Q}_i$, allowing visually similar examples from other candidate or non-candidate classes to provide complementary evidence.

Fixed-weight fusion. We combine the two competence estimates using a fusion weight $\gamma \in [0, 1]$:

$$\widehat{A}^{\mathrm{hyb}}_{i,s} = (1-\gamma)\widehat{A}^{\mathrm{cls}}_{i,s} + \gamma \widehat{A}^{\mathrm{loc}}_{i,s}. \tag{8}$$

Equivalently, this estimate can be written as $\widehat{A}^{\mathrm{cls}}_{i,s} + \gamma(\widehat{A}^{\mathrm{loc}}_{i,s} - \widehat{A}^{\mathrm{cls}}_{i,s})$. Thus, local evidence provides a query-specific correction to the class-conditioned estimate. We select γ on the validation set and keep it fixed at inference. A larger γ gives more weight to local evidence, while $\gamma = 0$ reduces to the class-conditioned estimator. We call this estimator *Fixed Hybrid*. It requires no additional routing network training and directly uses the estimated competence for skill selection.

3.3 Deployment-Aware Skill Selection

Different deployment settings require different accuracy–cost trade-offs. We separate competence estimation from the rule used to select a skill. The same frozen competence estimates support different deployment policies: accuracy-first selection, per-query utility optimization, and batch-budget allocation.

Cost estimation. We estimate each skill's cost from its offline profiling records. Following the use of skill-level cost profiles and model-level expected costs in prior routing work Wang et al. (2026); Hu et al. (2024), we use the mean realized cost as a surrogate for estimated cost:

$$\widehat{C}_s = \frac{1}{N_{\mathcal{P}}} \sum_{j=1}^{N_{\mathcal{P}}} C_{j,s}. \tag{9}$$

This estimate is shared across queries and requires no separate cost regressor. Query-dependent cost estimation is another option explored in routing work Somerstep et al. (2025); Wang et al. (2025). Our deployment rules can accommodate such estimates, but the formulation below uses the skill-level mean $\widehat{C}_s$. Estimated cost guides skill selection and realized cost is measured after execution.

Accuracy-first selection. When accuracy is the only objective, we select the skill with the highest estimated competence:

$$s^*_i = \arg\max_{s \in \mathcal{S}} \widehat{A}^{\mathrm{hyb}}_{i,s}. \tag{10}$$

Per-query utility. For cost-sensitive selection, we subtract a normalized cost penalty from the competence estimate:

$$s^*_i(\lambda) = \arg\max_{s \in \mathcal{S}} \left[\widehat{A}^{\mathrm{hyb}}_{i,s} - \lambda \frac{\widehat{C}_s}{\widehat{C}_{\mathrm{ref}}} \right]. \tag{11}$$

Here $\lambda \geq 0$ controls the accuracy–cost trade-off, and $\widehat{C}_{\mathrm{ref}} = \max_{s \in \mathcal{S}} \widehat{C}_s > 0$ is a shared normalization scale. In our skill pool, this reference is the estimated cost of visual ICL, which is the most expensive skill among all candidates. Larger λ increases the penalty on expensive skills. This rule selects a skill independently for each query and does not impose a hard budget.

Batch-budget allocation. When a batch of queries shares a budget, we allocate skills jointly. This allows the router to spend more on some queries and less on others. Let $\mathcal{B}$ be the set of query indices in the batch and B its estimated-cost budget. The binary variable $z_{i,s}$ indicates whether skill s is selected for query x_i. We solve

$$\begin{aligned} \mathbf{z}^* = \arg\max_{\mathbf{z}} \quad & \sum_{i \in \mathcal{B}} \sum_{s \in \mathcal{S}} z_{i,s} \widehat{A}^{\mathrm{hyb}}_{i,s} \\ \text{s.t.} \quad & \sum_{i \in \mathcal{B}} \sum_{s \in \mathcal{S}} z_{i,s} \widehat{C}_s \leq B, \qquad \sum_{s \in \mathcal{S}} z_{i,s} = 1, \qquad z_{i,s} \in \{0,1\}, \quad \forall i \in \mathcal{B},\ s \in \mathcal{S}. \end{aligned} \tag{12}$$

The constraints select exactly one skill per query and keep the total estimated cost within B. The selected skill is $s^*_i = \arg\max_{s \in \mathcal{S}} z^*_{i,s}$. Eq. 12 is a multiple-choice knapsack problem, which we solve with an integer optimizer with details deferred to the appendix. Its budget constraint applies to estimated cost. We evaluate realized budget compliance separately. Across all three selection rules, the competence estimator remains unchanged. Changing the deployment objective therefore does not require retraining it.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and model setup. We evaluate on 8 fine-grained image categorization benchmarks: CUB Wah et al. (2011), FGVC-Aircraft Maji et al. (2013), Stanford Cars Krause et al. (2013), Oxford-IIIT Pets Parkhi et al. (2012), Stanford Dogs Khosla et al. (2011), Oxford Flowers Nilsback & Zisserman (2008), Food-101 Bossard et al. (2014), and SUN397 Xiao et al. (2010). Our skill pool consists of five executable recognition strategies: CLIP kNN retrieval Zhang et al. (2022), direct MLLM classification Conti et al. (2025), category-description prompting Yang et al. (2026), CoT via NLG2C reasoning Conti et al. (2025) Lawrence et al. (2026), and visual in-context learning (VICL) Xiong et al. (2026). Detailed prompts and candidate-resolution rules are provided in Appendix A.2. We evaluated CLIP ViT-B/L, DINO v2/3 ViT-B and SigLip ViT-B for routing features. We use Qwen3-VL-8B-Instruct for skill execution with additional MLLMs including Qwen3-VL-32B-Instruct, Gemma3-4B, and Gemma3-12B to assess the robustness.

Baseline methods. We evaluate each of the five skills individually. We also include random skill selection Ding et al. (2024) and majority voting over all five skill outputs Chen et al. (2024a). The CLIP-based recognition baselines include zero-shot classification, prompt ensembling Radford et al. (2021), image-feature retrieval Zhang et al. (2022), Tip-Adapter Zhang et al. (2022), and CLAP Silva-Rodríguez et al. (2024). For adaptive skill selection, we compare with an Instance-kNN router Shirkavand et al. (2025), an MLP router Li et al. (2024), RouterDC Chen et al. (2024c), a FrugalGPT-style cascade Chen et al. (2024b), and a prompted VLM router Zhang et al. (2025). All routing methods use the same five-skill pool and are evaluated using the same recorded skill outcomes. This controls for differences in skill execution across routers. Task-specific adaptations and implementation details are provided in Appendix A.

Implementation details. CLIP provides the top-k candidate classes for the MLLM-based skills. The neighborhood size K, similarity temperature T, and fusion weight γ are selected on Validation for each dataset, the full hyperparameter search space is reported in Appendix A. For each dataset, we split the original training set by class into 80% for skill profiling and router training and 20% for Validation. We apply the same split to the trainable router baselines and select hyperparameters on Validation Shirkavand et al. (2025); Li et al. (2024); Chen et al. (2024c;b); Zhang et al. (2025). We then refit the corresponding statistics and learned routers using the full training set with all selected settings fixed. For VICL Xiong et al. (2026), all labeled training samples serve as demonstration candidates. Additional details on profiling, cost calibration, and deployment settings are provided in Appendices D.1.

4.2 ROUTING IMPROVES THE ACCURACY–COST TRADE-OFF

The five recognition skills have complementary strengths. Visual ICL (VICL) is the strongest fixed skill, achieving 83.75% average accuracy at 7,373 MLLM tokens per image. The hindsight oracle, which selects a correct skill whenever one is available, reaches 93.42%. This 9.67-percentage-point gap shows that other skills correctly recognize many images that VICL misclassifies. However, combining their outputs through majority voting achieves only 79.67% accuracy while consuming 12,516 tokens per image. Thus, skill complementarity provides an opportunity for routing, but does not guarantee an improvement through simple aggregation.

Our accuracy-first router, denoted Ours (Accuracy-only), achieves the highest average accuracy among the evaluated deployable methods: 84.50% at 4,747 tokens per image. Compared with VICL, it improves accuracy by 0.75 percentage points and significantly reduces token usage by 35.6%. It also achieves the highest accuracy among deployable methods on 7 of the 8 datasets: CUB, Aircraft, Cars, Pets, Dogs, and Food. The strongest routing baseline, the FrugalGPT-style cascade, reaches 83.80% at 8,230 tokens per image. Our router improves on this result by 0.70 percentage points while using 42.3% fewer tokens. At the reported operating points, it achieves higher average accuracy and lower token usage than every evaluated routing baseline. It also uses fewer MLLM calls and image inputs than these baselines, averaging 0.94 calls and 10.55 image inputs per query.

Table 1: Accuracy and MLLM token cost benchmarking. Avg. Acc. is the dataset macro average; costs are per image. Labeled imgs? indicates use of labeled training images, including retrieval and ICL examples. † marks baselines adapted to the shared skill pool. Bold/underline mark the best/second-best distinct values per column, excluding the oracle.

Method	Labeled imgs?	Overall		Accuracy (%) ↑								Usage ↓	
		Avg. Acc. ↑	Tokens ↓	CUB	Aircraft	Cars	Pets	Dogs	Flowers	Food	SUN	MLLM calls	Image inputs
CLIP baselines													
CLIP Zero-Shot (ICML'21)	No	65.08	–	55.26	24.87	63.47	89.13	64.90	71.56	88.87	62.60	–	–
CLIP Prompt Ensemble (ICML'21)	No	65.71	–	55.42	24.69	64.71	89.56	64.92	72.30	89.28	64.78	–	–
Tip-Adapter-Full (ECCV'22)	Yes	76.94	–	74.91	45.63	76.52	91.33	71.75	93.33	89.31	72.73	–	–
CLAP-Full (CVPR'24)	Yes	81.50	–	79.05	54.10	83.90	93.43	77.78	95.39	90.92	77.41	–	–
CLIP kNN Retrieval (ECCV'22)	Yes	72.23	–	71.09	43.32	73.77	83.67	62.41	83.46	89.44	70.64	–	–
Fixed MLLM skills													
Direct MLLM (ICCV'25)	No	64.86	**581**	45.20	35.79	74.19	82.83	54.45	75.82	83.85	66.72	1.00	**1.00**
Category Description (AAAI'26)	No	68.63	3,608	51.57	39.72	73.08	85.91	61.74	74.87	88.53	73.65	1.00	**1.00**
CoT Reasoning (WACV'26)	No	68.48	956	55.83	42.42	76.67	88.50	63.08	79.51	84.21	57.61	2.00	**1.00**
Visual ICL (CVPR'26)	Yes	83.75	7,373	80.39	62.32	84.93	92.97	78.61	**99.20**	92.35	79.22	1.00	18.00
Multi-skill baselines													
Random Skill Selection† (ICLR'24)	Yes	71.59	2,503	60.82	44.72	76.53	86.78	64.06	82.57	87.68	69.57	1.00	4.20
Skill Ensemble (Majority)† (NeurIPS'24)	Yes	79.67	12,516	75.46	55.06	84.21	92.20	72.69	87.61	92.19	77.98	5.00	21.00
Router baselines													
FrugalGPT Skill Cascade† (TMLR'24)	Yes	83.80	8,230	80.02	62.20	86.49	92.59	78.84	98.52	92.39	79.30	2.38	18.70
Instance-kNN Skill Router† (NeurIPS'25)	Yes	82.33	5,049	80.07	61.18	82.34	91.14	76.91	98.81	90.67	77.49	1.81	10.74
MLP Skill Router† (NeurIPS'24)	Yes	83.56	6,943	80.17	62.15	84.53	92.98	78.43	99.06	92.05	79.12	1.95	15.86
RouterDC† (NeurIPS'24)	Yes	83.56	7,168	80.05	62.12	85.02	93.08	78.51	99.14	92.17	78.41	1.95	17.09
Prompted VLM Router† (NeurIPS'25)	Yes	80.38	5,351	78.29	60.64	82.46	90.30	74.91	91.27	89.97	75.20	2.42	8.29
VEFR (Ours)													
VEFR (Accuracy-first)	Yes	**84.50**	4,747	**81.15**	**64.63**	**87.74**	**93.54**	**79.41**	97.56	**92.48**	**79.46**	0.94	10.55
VEFR (Utility)	Yes	83.70	2,645	80.20	64.51	86.71	92.94	78.60	96.58	91.45	78.60	**0.80**	4.99
VEFR (Batch budget)	Yes	84.03	3,287	80.22	64.54	87.34	93.38	78.72	97.15	92.25	78.66	0.87	6.96
Oracle upper bound													
Oracle Skill Selector	–	93.42	–	91.09	83.02	95.98	98.61	90.44	99.64	97.00	91.54	–	–

The same competence estimator supports lower-cost selection policies. Ours (Batch budget) achieves 84.03% accuracy at 3,287 tokens per image, exceeding VICL by 0.28 percentage points while reducing token usage by 55.4%. Ours (Utility) further reduces usage to 2,645 tokens per image, a 64.1% reduction relative to VICL, with an accuracy of 83.70%. This is only 0.05 percentage points below VICL. These results show that the estimator supports several useful accuracy–cost trade-offs through changes to the selection policy. Section 4.4 examines these trade-offs in detail.

4.3 Class and Local Evidence Are Complementary

We compare competence estimators under full profiling, using the same skill pool and accuracy-first selection rule. Global-only replaces $\widehat{A}_{i,s}^{\mathrm{hyb}}$ with the global skill mean g_s in Eq. 10. It assigns each skill a query-independent competence score. Class-only and Local-only use class-conditioned and local visual evidence, respectively.

Table 2: **Ablation of competence estimation under full profiling.** All variants use the same skill pool and accuracy-first selection rule. Accuracy is the macro average across eight datasets; tokens are per image.

Variant	Class Evidence	Local Evidence	Global prior	Acc. (%) ↑	Tokens/img ↓
Global-only			✓	83.75	7,373
Class-only	✓			84.15	4,783
Local-only		✓		82.84	**3,299**
Class + global prior	✓		✓	84.44	6,195
Class + local (Hybrid)	✓	✓		**84.50**	4,747

Table 2 shows that Class-only provides a strong routing signal, achieving 84.15% average accuracy at 4,783 tokens per image. Local-only achieves lower accuracy (82.84%) but uses fewer tokens (3,299 per image). Combining both sources in Fixed Hybrid increases accuracy to 84.50%, while slightly reducing token usage to 4,747. These results suggest that local visual evidence can refine class-conditioned competence estimates.

To examine whether this benefit comes from query-specific evidence, we replace the local estimate with the global skill prior. Class + global prior achieves 84.44% accuracy at 6,195 tokens per image. Fixed Hybrid achieves similar accuracy with 23.4% fewer tokens. Thus, the clearest benefit of local

evidence in this comparison is lower token usage at comparable accuracy. Additional controls with separately tuned global-prior fusion weights are reported in Appendix E.1.

4.4 One Competence Estimator Supports Multiple Deployment Regimes

We examine three deployment policies using the same frozen Fixed Hybrid competence scores: accuracy-first selection, per-query utility optimization, and batch-budget allocation.

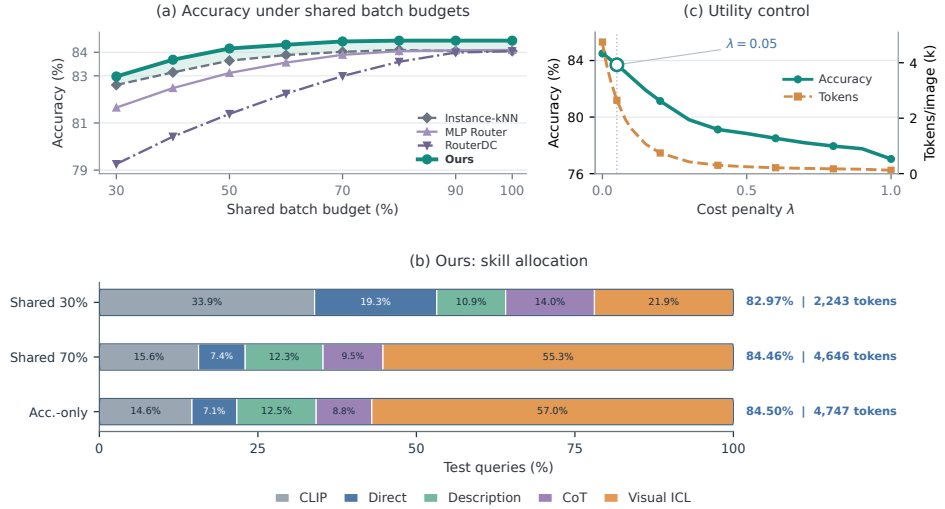


Figure 3: **Deployment with frozen Fixed Hybrid competence scores.** (a) Accuracy under shared batch-budget settings. (b) Skill allocation at two budget settings and under accuracy-first selection. (c) Accuracy and realized tokens per image as the cost penalty λ varies.

Figure 3(a) compares routing methods under shared batch-budget settings. Our method achieves higher average accuracy than Instance-kNN, the MLP router, and RouterDC at every evaluated budget. Its accuracy increases as the budget grows and then levels off. These results show that the frozen competence estimates remain useful across the evaluated budget range.

Figure 3(b) shows how batch-budget allocation changes the skill distribution. At the 30% budget setting, CLIP kNN retrieval handles 33.9% of queries, while visual in-context learning (VICL) handles 21.9%. This allocation achieves 82.97% accuracy at 2,243 tokens per image. At the 70% setting, the VICL share increases to 55.3%, yielding 84.46% accuracy at 4,646 tokens per image. Under accuracy-first selection, VICL handles 57.0% of queries. Thus, even without an explicit cost penalty, the estimator selects other skills for a substantial fraction of queries.

Figure 3(c) examines per-query utility optimization by varying the cost penalty λ. Larger penalties reduce token usage, with a corresponding decline in accuracy. The selected operating point, $\lambda = 0.05$, achieves 83.70% accuracy at 2,645 tokens per image. Compared with accuracy-first selection, it reduces token usage by 44.3% with an accuracy decrease of 0.80 percentage points. Together, these results show that a single frozen competence estimator supports different accuracy–cost trade-offs without retraining.

5 Conclusion

We presented VEFR, a framework that treats fine-grained recognition as a decision about which recognition skill to execute for each image. Its Fixed Hybrid estimator combines class-conditioned skill statistics with local visual evidence from cached correctness records, without training a separate routing network. Separating competence estimation from deployment control allows the same frozen scores to support accuracy-first selection, per-query utility optimization, and batch-budget allocation. Experiments across eight benchmarks show improved accuracy–cost trade-offs, while ablations suggest that local evidence refines class-conditioned estimates and reduces token usage at comparable accuracy. The approach still requires offline skill profiling, and a substantial gap remains between the router and hindsight oracle selection. Future work could reduce profiling requirements and account for uncertainty in competence and cost estimates.

AI USE STATEMENT

We used generative AI tools, including ChatGPT and Codex, to assist with refining research questions and experimental designs, implementing and debugging evaluation scripts, interpreting results, identifying relevant literature, drafting and editing manuscript text, and preparing figures and tables. Verification included executing evaluation scripts, comparing reported values with saved experimental outputs, and consulting original sources for citations. The authors take responsibility for the final manuscript, code, analyses, and claims. The MLLMs evaluated as recognition executors are part of the experimental method and are documented separately in the paper.

ETHICS STATEMENT

This work studies recognition-skill routing on established public image benchmarks and does not introduce new human-subject experiments or collect new personal data. Routing inherits the limitations and biases of its datasets, recognition skills, and pretrained models; improved aggregate accuracy does not eliminate these limitations. Practical deployment should consider recognition errors, subgroup performance, and the data-handling policies of external model services.

REPRODUCIBILITY STATEMENT

Appendix A documents the datasets, data splits, recognition-skill prompts, model configurations, and hyperparameter selection protocol. Appendix D specifies the cost estimation and budget-allocation procedures. Appendices C and E describe the controlled ablations and profiling protocols. Together, these details specify how competence records are constructed, settings are selected without Test supervision, and accuracy and realized token usage are evaluated. We will publicly release the implementation and evaluation scripts upon acceptance.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Md Atabuzzaman, Andrew Zhang, and Chris Thomas. Zero-shot fine-grained image classification using large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23569–23582, 2025.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101: Mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461, 2014. doi: 10.1007/978-3-319-10599-4_29.

Huiyi Chen, Jiawei Peng, Kaihua Tang, Xin Geng, and Xu Yang. Enhancing multimodal in-context learning for image classification through coreset optimization. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5130–5139, 2025.

Lingjiao Chen, Jared Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more llm calls all you need? towards the scaling properties of compound ai systems. *Advances in Neural Information Processing Systems*, 37:45767–45790, 2024a.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024b.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024c.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5558–5570, 2022.

Mia Chiquier, Utkarsh Mall, and Carl Vondrick. Evolving interpretable visual classifiers with large language models. In *European Conference on Computer Vision*, pp. 183–201. Springer, 2024.

Alessandro Conti, Massimiliano Mancini, Enrico Fini, Yiming Wang, Paolo Rota, and Elisa Ricci. On large multimodal models as open-world image classifiers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16388–16398. IEEE, 2025.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, volume 2024, pp. 41348–41366, 2024.

Ruijiang Dong, Zesheng Ye, Jianzhong Qi, Lei Feng, Feng Liu, Gang Niu, and Masashi Sugiyama. Carprt: Class-aware zero-shot prompt reweighting for vision-language model. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *European conference on computer vision*, pp. 153–168. Springer, 2020.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *International Conference on Learning Representations*, 2025.

Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4438–4446, 2017.

Lucas Goncalves, Robert A Barton, Vidit Bansal, and Karim Bouyarmane. Agentic prompt optimization with visual contrastive reasoning for fine-grained classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3101–3110, 2026.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander G Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 956–968, 2022.

Shaunak Halbe, Junjiao Tian, KJ Joseph, James Seale Smith, Katherine Stevo, Vineeth N Balasubramanian, and Zsolt Kira. Grounding descriptions in images informs zero-shot visual recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5672–5681, 2026.

Hulingxiao He, Geng Li, Zijun Geng, Jinglin Xu, and Yuxin Peng. Analyzing and boosting the power of fine-grained visual recognition for multi-modal large language models. In *The Thirteen International Conference on Learning Representations*, 2025.

Hulingxiao He, Zijun Geng, and Yuxin Peng. Fine-r1: Make multi-modal llms excel in fine-grained visual recognition by chain-of-thought reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ju He, Jie-Neng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. Transfg: A transformer architecture for fine-grained recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 852–860, 2022.

Yunqi Hong, Sohyun An, Andrew Bai, Neil Lin, and Cho-Jui Hsieh. Unlabeled data improves fine-grained image zero-shot classification with multimodal llms. *Advances in Neural Information Processing Systems*, 38:74304–74326, 2026.

Saihui Hou, Yushan Feng, and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *Proceedings of the IEEE international conference on computer vision*, pp. 541–549, 2017.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided image captioning for vqa with gpt-3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2963–2975, 2023a.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 23369–23379, 2023b.

Brandon Huang, Chancharik Mitra, Assaf Arbelle, Leonid Karlinsky, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *Advances in Neural Information Processing Systems*, 37:22124–22153, 2024.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19113–19122, 2023.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2011. URL `https://people.csail.mit.edu/khosla/papers/fgvc2011.pdf`.

Jeonghwan Kim and Heng Ji. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6187–6207, 2024.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Logan Lawrence, Oindrila Saha, Megan Wei, Chen Sun, Subhransu Maji, and Grant Van Horn. You may speak freely: Improving the fine-grained visual recognition capabilities of multimodal large language models with answer extraction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1428–1437, 2026.

Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *Advances in Neural Information Processing Systems*, 37:131224–131246, 2024.

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

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1457, 2015.

Fan Liu, Wenwen Cai, Jian Huo, Chuanyi Zhang, Delong Chen, and Jun Zhou. Making large vision language models to be good few-shot learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5415–5423, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pp. 126–142. Springer, 2024.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. In *International Conference on Learning Representations*, 2025.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

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

Reza Shirkavand, Shangqian Gao, Peiran Yu, and Heng Huang. Cost-aware contrastive routing for llms. *Advances in Neural Information Processing Systems*, 38:155627–155659, 2025.

Julio Silva-Rodríguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23681–23690, 2024.

Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. CARROT: A cost aware rate optimal router. *arXiv preprint arXiv:2502.03261*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11888–11898, 2023.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12884–12893, 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

Jiayu Wang, Yifei Ming, Zixuan Ke, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. Skillorchestra: Learning to route agents via skill transfer. *arXiv preprint arXiv:2602.19672*, 2026.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. Mixllm: Dynamic routing in mixed large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10912–10922, 2025.

Canshi Wei. Enhancing fine-grained image classifications via cascaded vision language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1857–1871, 2024.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Guangzhi Xiong, Sanchit Sinha, Zhenghao He, and Aidong Zhang. Retrieving counterfactuals improves visual in-context learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24352–24362, 2026.

Yutong Yang, Lifu Huang, Yijie Lin, Xi Peng, and Mouxing Yang. Endowing vision-language models with system 2 thinking for fine-grained visual recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11802–11810, 2026.

Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 420–435, 2018.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 3081–3089, 2022.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 141233–141265, 2025.

Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pp. 834–849. Springer, 2014.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022.

Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 5209–5217, 2017.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022a.

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, 2022b.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Jie Zhu, Yiyang Su, and Xiaoming Liu. Can textual reasoning improve the performance of mllms on fine-grained visual classification? In *CVPR Findings*, 2026.

Appendix overview. We refer to our fixed-weight class–local routing method as Visual Evidence Fusion Routing (VEFR). Appendix A specifies the evaluation protocol and recognition skills. Appendices B and C examine skill complementarity and routing evidence. Appendix D details cost estimation and deployment controls. Appendix E studies limited profiling, changes of executor, and profile reuse. Appendix F places selected routing decisions in the context of population-level failure modes.

A Experimental Details and Reproducibility

A.1 Datasets, splits, and evaluation units

Table 3: **Dataset populations used in the experiments.** Original training images are split within each class using seed 42; both subsets retain the same classes.

Dataset	Classes	Original Train	Router-Train	Validation	Test
CUB	200	5,994	4,795	1,199	5,794
Aircraft	100	6,667	5,333	1,334	3,333
Cars	196	8,144	6,515	1,629	8,041
Pets	37	3,680	2,944	736	3,669
Dogs	120	12,000	9,600	2,400	8,580
Flowers	102	2,040	1,632	408	6,149
Food	101	75,750	60,600	15,150	25,250
SUN	397	19,850	15,880	3,970	19,850

Within each class, the original training images are split into a profiling and router-training subset (80%) and Validation (20%) using seed 42. The two subsets contain the same classes but disjoint images. We record skill outcomes and fit trainable routers on the 80% subset; Validation selects their settings and deployment operating points. With these settings fixed, we recompute the competence statistics and retrain the routers on all original training images before Test evaluation. Accuracy is averaged equally over the eight datasets; token usage is averaged over their Test images (80,666 in total).

The evidence encoder and the executable retrieval skill play different roles. The headline router uses L2-normalized DINOv3 ViT-B/16 pooler features and cosine similarity. The executable CLIP kNN skill, top-k candidate lists, recognition-skill outputs, and Visual-ICL demonstrations remain fixed when the routing evidence encoder changes. Thus an encoder ablation changes the evidence used to select a skill, not the skill's underlying predictions.

A.2 Executable recognition skills and prompts

There are five executable skills: CLIP retrieval, Direct MLLM, Category Description, two-stage CoT, and Visual ICL. CLIP retrieval makes no MLLM call. Direct and Description receive the query image and the same ordered candidate list. CoT first generates an unrestricted textual prediction, then resolves that response against the candidates without a second image input. Visual ICL interleaves the query with retrieved labeled demonstrations and repeats the query at the end.

CLIP candidate construction. For each class c, we encode the prompt "a photo of a `<class name>`." using frozen CLIP ViT-B/16. After L2 normalization, we rank classes by image–text cosine similarity and retain the top-$\min(20, |\mathcal{C}|)$ candidates. This candidate list is constructed before MLLM execution and is shared by Direct, Description, and the text-only resolution stage of CoT. Visual ICL retrieves demonstrations separately and uses the dataset-wide class options. MLLM outputs are normalized and matched against valid candidates; invalid outputs are counted as incorrect, except that CoT failures fall back to the CLIP top-1 prediction following its two-stage design.

Below, `<type>` is the dataset-specific category type, and `<candidates>` denotes the ordered category names. The category types are bird species, aircraft model, car model, pet breed, dog breed, flower species, food category, and scene category, respectively. The ICL question is "What is the category of the bird in this image?" for CUB and the corresponding flower question for Flowers; other datasets use "What is the `<type>` in this image?".

System message. Direct, Description, and both CoT stages use:

> You are a helpful assistant who specializes in fine-grained classification.

Visual ICL uses the interleaved user message below without an additional system message.

Direct MLLM.

> Which `<type>` best matches the image? Choose from the following categories:
> `<candidates>`
> Output exactly one category name copied from the list. Do not explain your answer.

Category Description. The description for each candidate is inserted as a separate block:

> Which `<type>` best matches the image? Choose from the following categories and corresponding descriptions:
> Category: `<candidate 1>`
> `<description 1>`
> Category: `<candidate 2>`
> `<description 2>`
> *[Repeat the category/description block for every candidate.]*
> Output exactly one category name copied from the list. Do not explain your answer.

Category descriptions are generated once, before evaluation, by prompting GPT-4.1 mini with each class name for concise diagnostic visual traits. The text-only prompts specify relevant attributes for each dataset, such as plumage and body parts for birds or body details for cars.

CoT: free prediction followed by candidate resolution. The first stage receives the image and the text prompt:

> What is the `<type>` in this image?

The second stage is text-only:

> What is the most likely `<type>` indicated in this response?
> Response:
> `<first-stage response>`
> Answer from:
> `<candidates>`
> Output exactly one line in this format:
> Answer: `<one label copied exactly from Answer from>`
> Do not output any other words, explanation, or punctuation-only lines.

Visual ICL. The full message order is:

> Your task is to perform Image Classification. You need to choose one of the following options: `<output options>`
> *[Query image]*
> Question: `<classification question>`
> Here are 16 in-context examples to help you answer the question:
> *[Demonstration image 1]*
> Question: `<classification question>`
> Answer: `<demonstration label 1>`
> *[Repeat the image/question/answer block for all demonstrations.]*
> Here is the original question again.
> *[Query image repeated]*
> Question: `<classification question>`
> Please provide your response by directly outputting the answer.

Demonstrations are retrieved from the permitted labeled gallery, with self-demonstrations excluded during profiling.

A.3 Adaptation of Routing Baselines

All routing baselines are adapted to the same five-skill recognition setting. For each profiling image, we record a five-dimensional correctness vector indicating whether each skill produces the correct label. Caption-based routers Shirkavand et al. (2025); Li et al. (2024); Chen et al. (2024c;b); Zhang et al. (2025) use the same visual caption generated by Qwen3-VL-8B; the Prompted VLM router instead receives the query image.

Learned routers. Because existing skill routing methods reply on textual descriptions, Instance-kNN, MLP Router, and RouterDC operate on frozen CLIP-B/16 text embeddings of the image caption. Instance-kNN estimates skill competence from neighboring profiling captions using inverse-distance weighting. MLP Router predicts the five skill success probabilities with binary cross-entropy supervision. RouterDC is adapted by replacing its original routing targets with the five-dimensional skill correctness vectors and optimizing the corresponding contrastive objectives.

Cascade routing. The FrugalGPT-style cascade follows a sequential decision process: a scorer estimates whether the current skill is likely to succeed, and the cascade executes additional skills when the confidence is insufficient. The scorer is trained from caption–skill outcome pairs, while the ordering and stopping thresholds are selected on Validation. All intermediate executions are counted in the reported token cost.

Prompted VLM router. The prompted VLM baseline predicts the preferred skill from the query image and available skill descriptions. Table 1 retains its recorded image-based choices. Its reported token usage includes estimated caption-generation cost, but no caption-input routing evaluation is implied.

CLIP baselines. CLIP zero-shot and prompt ensembling use frozen text prototypes. CLIP kNN, Tip-Adapter-Full, and CLAP-Full use the complete labeled gallery following their original formulations. These methods do not invoke MLLMs.

Trainable routers select hyperparameters on Validation and are retrained on the full original training set after selection. Their results average five independent router-training seeds on the same image split; these are distinct from the five support-gallery draws in the limited-profiling study.

A.4 Hyperparameter Selection for VEFR

For the main full-profile experiments, we select the neighborhood size, similarity temperature, and fusion weight on the Validation set using the grid $K \in \{5, 10, 20, 50\}$, $T \in \{0.03, 0.05, 0.07, 0.10\}$, and $\gamma \in \{0, 0.1, \ldots, 1\}$. The full-profile results reported in the main paper use the unsmoothed estimator with $\tau = 0$.

When multiple configurations achieve the same Validation accuracy, we select the one with lower estimated token cost, followed by the fixed skill ordering CLIP, Direct, Description, CoT, and ICL.

Table 4: **Validation-selected VEFR configurations.** The selected hyperparameters are frozen before full-train refitting and Test evaluation.

Dataset	Neighborhood K	Temperature T	Fusion γ
CUB	50	0.07	0.6
Aircraft	5	0.07	0.2
Cars	20	0.05	0.7
Pets	5	0.10	0.1
Dogs	20	0.03	0.2
Flowers	20	0.03	1.0
Food	20	0.05	0.2
SUN	50	0.10	0.1

The fusion coefficient γ is selected independently for each dataset and kept fixed across its Test queries; only the evidence-dependent competence estimates vary by query.

A.5 Resources and Reproducibility Metadata

We report MLLM resource usage using input/output tokens, requests, and image inputs. Token counts exclude frozen local encoder computation. Following the skill definitions, CoT uses two MLLM requests with one image input, while Visual ICL uses one request with 18 image inputs: 16 demonstrations and the query image presented twice.

Experiments evaluate Qwen3-VL-8B, Qwen3-VL-32B, Gemma3-12B, and Gemma3-4B executors. Local routing overhead is measured separately on an A100-SXM4-80GB and reported in Table 11; it excludes remote MLLM inference latency.

B Skill Complementarity and Pool Controls

B.1 Complementary errors

Let Y_s indicate whether skill s correctly classifies a Test image. For two skills a and b, the directional rescue rate

$$R_{a\to b} = \Pr(Y_b = 1 \mid Y_a = 0) \tag{13}$$

measures the fraction of a's errors corrected by b. The pairwise-oracle gain

$$U_{a,b} = \Pr(Y_a = 1 \vee Y_b = 1) - \max\{\Pr(Y_a = 1), \Pr(Y_b = 1)\} \tag{14}$$

instead measures the additional coverage available to a selector that knows both outcomes. These quantities are computed within each dataset and then averaged equally over the eight datasets; oracle selection is retrospective.

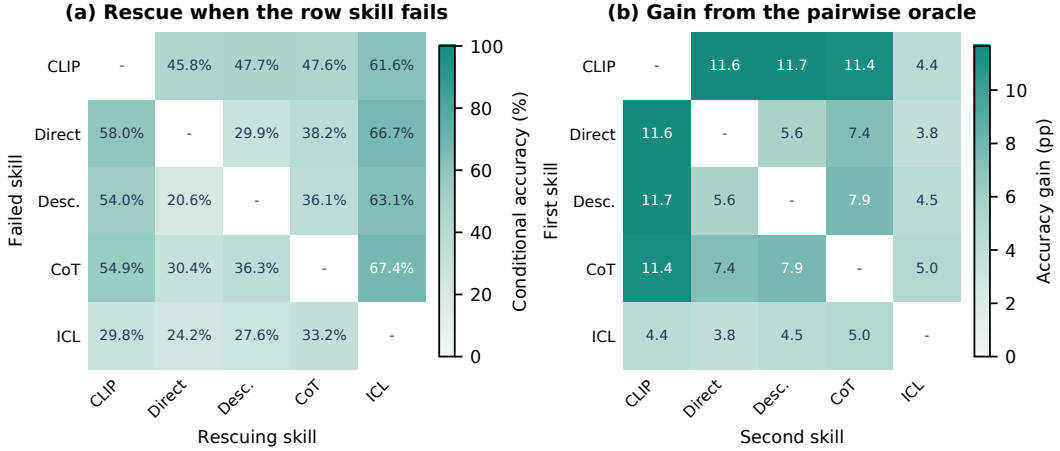


Figure 4: **Complementary errors in the five-skill pool.** (a) The column skill's accuracy conditional on failure of the row skill. (b) Pairwise-oracle improvement over the stronger individual skill, in percentage points. Both panels average dataset-level quantities equally.

Visual ICL is the strongest fixed skill, yet another skill corrects 24.2–33.2% of its errors, depending on the alternative chosen. Thus the skill pool has coverage beyond any fixed skill. The oracle gains in Figure 4(b) quantify this opportunity, not the accuracy of a deployable router: the successful skill is unknown before execution.

B.2 Expanding the executable pool

Adding Tip-Adapter raises the macro oracle union by 0.79 percentage points, but improves routing accuracy by only 0.12 points with CLIP-kNN evidence. With calibrated Tip evidence, the expanded pool costs fewer tokens but does not improve accuracy. Additional skill coverage is therefore useful only when competence estimation can identify the queries that benefit from it. The headline pool retains the original five skills; Tip-Adapter remains a separate recognition baseline.

Table 5: **Effect of adding Tip-Adapter to the skill pool.** The evidence column identifies either class probabilities from CLIP-kNN neighbors or Tip-Adapter logits temperature-calibrated on Validation ("Calibrated Tip"). "Tip selected" is the fraction of Test queries assigned to that skill; oracle union counts a query as correct when any skill in the pool succeeds. Accuracy and oracle union are macro-averaged; tokens/image and Tip selection are image-weighted.

Skill pool	Evidence	Acc. ↑	Tokens ↓	Tip selected	Oracle union ↑
5 skills	CLIP-kNN	83.83	5,682	–	93.42
5 + Tip	CLIP-kNN	**83.95**	5,571	7.1%	**94.21**
5 skills	Calibrated Tip	83.87	5,020	–	93.42
5 + Tip	Calibrated Tip	83.45	**4,734**	11.8%	**94.21**

B.3 Visual-ICL context size

The main results use 16 in-context demonstrations. To examine sensitivity to this choice, Figure 5 varies the demonstration count $M \in \{1, 2, 4, 8, 16\}$ on CUB and Aircraft. This M is distinct from the profiling-neighborhood size K used by the router. For each M, the Visual-ICL outcomes and competence profiles are recomputed, and routing settings are chosen on the corresponding Validation split; non-ICL skill outcomes remain fixed.

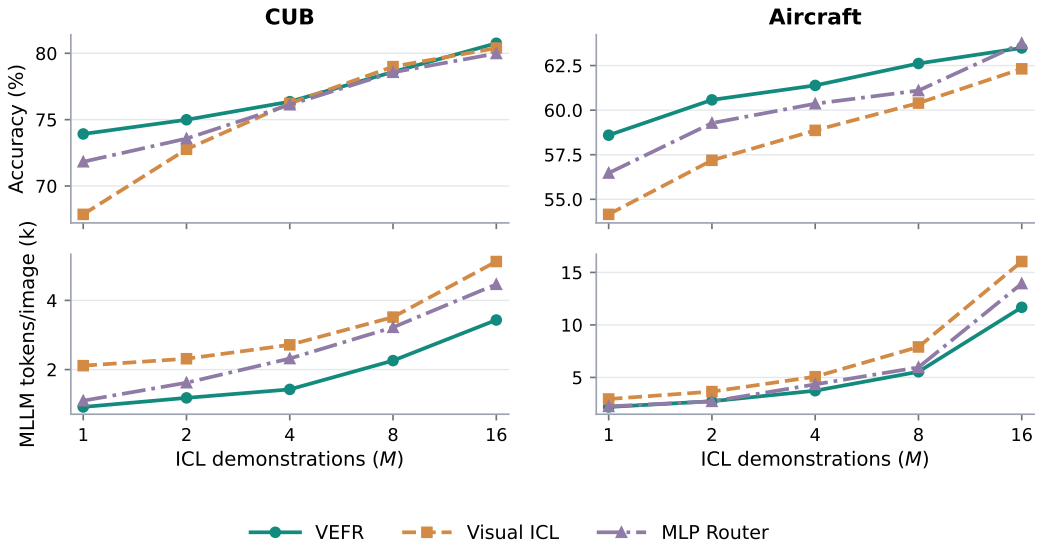


Figure 5: **Routing across Visual-ICL context sizes.** Test accuracy and recorded MLLM tokens/image on CUB and Aircraft for $M = 1, 2, 4, 8, 16$ demonstrations. Each setting is profiled and selected on its own Validation split. The MLP Router here uses DINOv3 features and binary-cross-entropy skill supervision.

At $M = 4$, VEFR gains 0.14 and 2.52 points over Visual ICL on CUB and Aircraft while using 47.5% and 26.3% fewer tokens, respectively. The trend persists across the evaluated context sizes: a larger ICL context does not remove the value of choosing a skill per query.

C Visual Evidence and Fusion Robustness

C.1 How do the evidence branches alter decisions?

Class and Local select different skills on 49.13% of Test queries, yet VEFR changes only 9.22% of Class-only choices (Table 6). On these changed queries, accuracy rises from 73.96% to 77.68%. The local branch thus provides selective corrections to a class-conditioned reference rather than replacing it on most images.

Table 6: **Effect of evidence fusion on routing decisions.** Statistics use frozen Test routes across eight datasets; the last two rows condition on queries whose Class-only choice is changed by VEFR.

Statistic	Value (%)
Class and Local choose different skills	49.13
VEFR changes the Class-only choice	9.22
VEFR accuracy among changed choices	77.68
Class-only accuracy among changed choices	73.96

C.2 Robustness across evidence encoders

We replace the DINOv3 routing encoder with CLIP-B/16, CLIP-L/14, DINOv2-B/14, or SigLIP-B/16. The five recognition skills, CLIP candidate lists, and Validation/Test split remain unchanged; only the visual evidence used to estimate competence changes. Tables 7 and 8 compare Class-only and VEFR at matched retrieval settings under full and limited profiling.

Table 7: **Class–local fusion across visual encoders.** (a) Macro accuracy and (b) per-dataset gains over Class-only at matched retrieval settings. Positive differences favor VEFR; W/T/L counts dataset-level wins, ties, and losses.

(a) Macro accuracy at matched retrieval settings

Evidence encoder	Class only (%)	VEFR (%)	Gain (pp)
CLIP ViT-B/16	83.66	83.96	+0.30
CLIP ViT-L/14	83.67	84.15	**+0.48**
DINOv2 ViT-B/14	83.91	84.12	+0.20
DINOv3 ViT-B/16	84.15	84.50	+0.34
SigLIP ViT-B/16	**84.22**	**84.59**	+0.37

(b) Per-dataset accuracy gains

Evidence encoder	CUB	Airc.	Cars	Pets	Dogs	Flowers	Food	SUN	W/T/L
CLIP ViT-B/16	+0.21	+0.72	+0.06	−0.71	+0.17	+1.81	0.00	+0.13	6/1/1
CLIP ViT-L/14	+0.28	**+1.59**	−0.30	0.00	+0.08	**+1.85**	+0.11	+0.21	6/1/1
DINOv2 ViT-B/14	+0.40	+0.54	+0.19	+0.14	−0.17	+0.36	**+0.12**	+0.07	7/0/1
DINOv3 ViT-B/16	**+0.55**	+0.33	0.00	**+0.25**	**+0.68**	+0.63	+0.11	+0.20	7/1/0
SigLIP ViT-B/16	+0.33	0.00	**+0.32**	−0.25	+0.42	+1.69	+0.05	**+0.38**	6/1/1

Table 8: **Factorization gains across visual encoders and profiling budgets.** Entries are eight-dataset macro accuracy differences (pp), Class+Local minus unregularized Class-only. Each shot is one profiled image per class; few-shot values average five paired, class-balanced support galleries. Full uses all original training images.

Evidence encoder	1-shot	2-shot	4-shot	Full
CLIP-B/16	$+\mathbf{3.36}$	$+1.97$	$+\mathbf{1.37}$	$+0.30$
CLIP-L/14	$+3.12$	$+\mathbf{2.11}$	$+1.35$	$+\mathbf{0.48}$
DINOv2-B/14	$+2.68$	$+1.92$	$+1.29$	$+0.20$
DINOv3-B/16	$+2.82$	$+1.89$	$+1.18$	$+0.34$
SigLIP-B/16	$+3.00$	$+1.67$	$+1.25$	$+0.37$

Adding local evidence improves the eight-dataset mean for every encoder, including with one to four profiling examples per class. The effect is not uniform across datasets, as Table 7(b) shows. Table 9 reports the corresponding full-profile accuracy and realized token usage against fixed reference methods; SigLIP attains the highest macro accuracy among VEFR encoders, while DINOv3 uses the fewest MLLM tokens.

Table 9: **Encoder robustness against fixed recognition baselines.** (a) Per-dataset accuracy and (b) macro accuracy, token usage, and savings relative to ICL. Encoder rows use VEFR; ICL and FrugalGPT retain their original inputs and outputs.

(a) Test accuracy (%) on all eight datasets

Pipeline / encoder	CUB	Airc.	Cars	Pets	Dogs	Flowers	Food	SUN
Visual ICL	80.39	62.32	84.93	92.97	78.61	**99.20**	92.35	79.22
FrugalGPT (caption)	80.02	62.20	86.49	92.59	78.84	98.52	92.39	79.30
CLIP ViT-B/16	80.70	62.80	87.30	93.08	78.32	98.02	92.40	79.07
CLIP ViT-L/14	80.86	63.70	86.42	93.84	78.39	97.98	**92.61**	79.36
DINOv2 ViT-B/14	80.84	63.19	86.83	**93.87**	79.34	97.45	92.35	79.08
DINOv3 ViT-B/16	**81.15**	**64.63**	87.74	93.54	**79.41**	97.56	92.48	79.46
SigLIP ViT-B/16	80.77	64.03	**87.82**	93.49	79.27	99.14	92.44	**79.77**

(b) Eight-dataset accuracy and deployment cost

Pipeline / encoder	Macro acc. (%)	Tokens/img	Saving vs. ICL (%)
Visual ICL	83.75	7,373	Ref.
FrugalGPT (caption)	83.80	8,230	—
CLIP ViT-B/16	83.96	5,516	25.2
CLIP ViT-L/14	84.15	5,072	31.2
DINOv2 ViT-B/14	84.12	4,966	32.6
DINOv3 ViT-B/16	84.50	**4,747**	**35.6**
SigLIP ViT-B/16	**84.59**	5,222	29.2

Encoder rows use VEFR; fixed references retain their original inputs. Accuracy is macro-averaged and tokens are image-weighted; local encoder computation is excluded. Table 8 reports the paired contribution of local evidence rather than selecting datasets by Test performance.

C.3 Sensitivity to the profiling neighborhood

Table 10 varies the neighborhood size $K \in \{1, 3, 5, 10, 20, 50\}$ used by both the class posterior and local correctness estimate. The skill outcomes are fixed; at each K, other method-specific settings are selected on Validation. The main results instead select K per dataset from $\{5, 10, 20, 50\}$.

Table 10: **Full-profile routing across matched neighborhood sizes.** Accuracy is averaged over eight datasets; MLLM token usage is reported per Test image. Each fixed K uses Validation-selected remaining settings. CLIP kNN makes no MLLM calls and is omitted from the token rows.

K	1	3	5	10	20	50
Accuracy (%) ↑						
CLIP kNN	67.91	70.55	71.97	73.08	73.60	73.88
Instance-kNN	73.26	75.38	77.09	80.05	81.85	83.13
VEFR	**83.87**	**84.14**	**84.26**	**84.21**	**84.35**	**84.68**
MLLM tokens/image ↓						
Instance-kNN	**926**	**1,680**	**2,389**	**3,684**	4,991	6,130
VEFR	3,865	4,102	4,234	4,391	**4,580**	**5,293**

VEFR stays between 83.87% and 84.68% across this sweep, including the single-neighbor case where the local estimate comes from one profiling image. This is a pipeline sensitivity check rather than an isolated intervention on local evidence: changing K also changes the class posterior and the skills ultimately selected. Instance-kNN varies more strongly with K, consistent with the class branch stabilizing the fused estimate when local evidence is sparse.

D Deployment and Cost Estimation

D.1 Pre-execution cost and operating points

Accuracy-first routing does not use a cost estimate. The Utility and Batch-budget policies in Table 1 use the mean skill cost in Eq. 9, computed from profiling images for Validation selection and from the full original training set for Test. Recorded costs include MLLM input and output tokens; CLIP retrieval has zero MLLM token cost. The deployment results report token usage measured after skill execution.

The shared-budget comparison and skill allocations in Figure 3(a,b) instead use the same query-dependent cost estimate for every router. For each MLLM skill, a Ridge regressor f_s predicts token usage from pre-execution candidate-label statistics and image geometry ϕ_i. With profiling images $\mathcal{P}$, we add a one-sided quantile of five-fold held-out residuals:

$$\widehat{C}^{\mathrm{up}}_{i,s} = \max\Big\{0,\ f_s(\phi_i) + \Big[Q_q\Big(\{C_{j,s} - f_s^{(-F(j))}(\phi_j)\}_{j\in\mathcal{P}}\Big)\Big]_+\Big\}. \tag{15}$$

Here $F(j)$ denotes example j's fold, $f_s^{(-F(j))}$ is fitted without that fold, Q_q is the empirical q-quantile, and $[u]_+ = \max\{u, 0\}$. Figure 3(a,b) fixes the Ridge penalty to 1 and $q = 0.95$. The upper estimate is intended to reduce underprediction; it does not guarantee that realized cost meets a requested budget.

For Table 1, Validation selects $\lambda = 0.05$ for Utility from $\{0, 0.0025, 0.005, 0.01, 0.015, 0.02, 0.03, 0.05, 0.08, 0.1, 0.15, 0.2\}$ and $b = 0.7$ for Batch budget from $\{0, 0.05, \ldots, 1\}$. The latter interpolates between the minimum predicted cost and the predicted cost of accuracy-first routing:

$$\begin{aligned} B &= B_{\min} + b(B_{\max} - B_{\min}), \qquad B_{\min} = \sum_{i\in\mathcal{B}} \min_s \widehat{C}_s, \\ B_{\max} &= \sum_{i\in\mathcal{B}} \min_{s\in\arg\max_t \widehat{A}^{\mathrm{hyb}}_{i,t}} \widehat{C}_s. \end{aligned} \tag{16}$$

Because CLIP has zero MLLM token cost, $B_{\min} = 0$. Among Validation points within one accuracy point of accuracy-first routing, we choose the least expensive; the batch policy must also meet its realized Validation budget. The optimization constraint itself applies to estimated cost.

D.2 Batch allocation and realized cost

We solve Eq. 12 with the HiGHS integer optimizer and break ties at the certified maximum predicted reward by minimizing estimated cost. The resulting assignments have either relative MIP gap below 10^{-8} or a dual bound within 0.001 percentage points of mean predicted reward. These are optimization certificates, not guarantees about Test accuracy.

At budget fraction ρ, the shared controller in Figure 3(a,b) uses $B^{\mathrm{shared}}(\rho) = \rho \sum_{i\in\mathcal{B}} \max_s \widehat{C}^{\mathrm{up}}_{i,s}$. It supplies every compared router with the same skill costs and integer allocation rule, so only the competence estimates differ.

For larger batches, a Lagrangian approximation chooses $s_i(\mu) = \arg\max_s\{\widehat{A}^{\mathrm{hyb}}_{i,s} - \mu\widehat{C}_{i,s}\}$ for $\mu \geq 0$, with dual bound

$$D(\mu) = \mu B + \sum_{i\in\mathcal{B}} \max_s\{\widehat{A}^{\mathrm{hyb}}_{i,s} - \mu\widehat{C}_{i,s}\}. \tag{17}$$

Here $\widehat{C}_{i,s} = \widehat{C}_s$ for the mean-cost policy. Bisection over μ is followed by budget repair and feasible upgrades. The dual bound certifies a maximum gap of 0.0026 percentage points in mean predicted reward on the evaluated instances, with no estimated-budget violation. All reported paper results nevertheless use the integer solution.

As a separate check of realized cost, we replay 24 sampled batches per dataset and batch size across eight Test sets. Batch sizes are 64, 256, and 1,024 images, each evaluated at 0.3, 0.7, and 1.0 times its predicted accuracy-first cost. This replay uses a separately fitted Ridge model (penalty 10,

$q = 0.99$). No evaluated batch exceeds its requested budget; the largest realized-to-budget ratio is 0.956. This observation concerns the evaluated batches rather than a general budget guarantee.

Table 11 reports local computation excluding remote MLLM execution.

Table 11: **Local routing overhead.** Per-query median and 95th-percentile (P95) latency on an A100-SXM4-80GB after warm-up; remote MLLM execution is excluded.

Component	Median (ms)	P95 (ms)
Query encoding (DINOv3)	13.82	18.29
Neighbor search	4.91	5.05
Competence and cost estimation	0.42	0.48
Total per-query routing	17.78	23.77

E Profiling Robustness and Executor Generalization

E.1 Sensitivity to profiling-set size

We vary the profiling set from 1 to 16 images per class to examine how much skill-outcome supervision VEFR requires. The headline full-profile method uses $\tau = 0$; here we also evaluate optional global smoothing to stabilize sparse class statistics. Only the selected class-balanced support images provide skill-success records and form the routing-evidence gallery. The recognition skills and Validation split are held fixed. The optional global prior is the mean over these same records, $g_s = |\mathcal{D}_{\text{sup}}|^{-1} \sum_{j \in \mathcal{D}_{\text{sup}}} y_{j,s}$; it does not use additional skill outcomes. Base and smoothed VEFR use $K = 10$ and $T = 0.07$; Validation selects γ for both and $\tau \in \{0, 0.5, 1, 2, 4, 8, 16, 32\}$ only for the smoothed variant. Results average five paired, independently sampled support galleries.

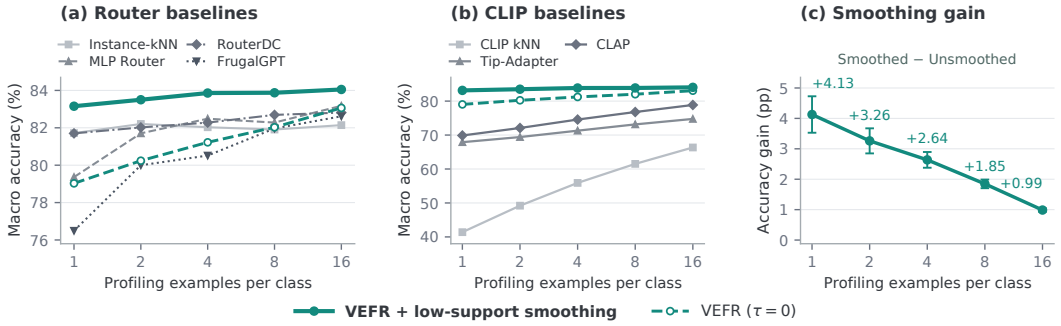


Figure 6: **Sensitivity to profiling-set size.** Eight-dataset macro accuracy against (a) router and (b) CLIP baselines; the panels use different accuracy scales. (c) Accuracy gain from smoothing over the unsmoothed VEFR as the number of profiled images per class grows. Error bars show one standard deviation over five paired support galleries.

With low-support smoothing, VEFR is relatively insensitive to profiling-set size over the evaluated range: macro accuracy increases from 83.16% with one image per class to 84.05% with sixteen, a difference of only 0.89 percentage points (Figure 6 and Table 12). It exceeds the four evaluated router baselines at every profiling size. In contrast, unsmoothed VEFR is more sensitive to sparse records, increasing from 79.03% to 83.06%. The smoothing gain consequently contracts from 4.13 to 0.99 points as support grows, consistent with regularization becoming less important when class statistics are better supported. This stability comes with higher token usage than unsmoothed VEFR. Global-only remains competitive at one and two shots, indicating that a reliable overall skill preference explains much of the recovery in the sparsest regime. These results support smoothing as a low-support regularizer; the full-profile headline method retains $\tau = 0$.

As a full-profile control, we also replace local evidence with the training-wide skill mean g_s and select its mixing weight $\eta \in \{0, 0.1, \ldots, 1\}$ on Validation, with the headline K and T fixed. This

Table 12: **Routing across profiling-set sizes.** Each shot is one profiled image per class. Cells report eight-dataset macro accuracy (%) / realized MLLM tokens per Test image, averaged over five paired class-balanced support galleries. Bold/underline rank each metric within a shot; † denotes baselines adapted to the shared skill pool.

Method	1-shot	2-shot	4-shot	8-shot	16-shot
Instance-kNN†	81.72 / 6,011	82.21 / 6,144	82.03 / 5,943	81.91 / 5,845	82.14 / 5,752
MLP Router†	79.36 / 4,318	81.70 / 5,881	82.49 / 6,890	82.28 / 6,575	83.18 / 7,288
RouterDC†	81.70 / 6,569	82.02 / 6,487	82.28 / 6,723	82.69 / 6,779	82.81 / 6,639
FrugalGPT†	76.49 / 6,070	80.00 / 6,256	80.52 / 6,723	81.96 / 7,834	82.62 / 7,900
Global-only	**83.35** / 6,866	**83.52** / 7,338	83.75 / 7,373	83.75 / 7,373	83.75 / 7,373
Smoothed Class-only	83.23 / 6,776	83.49 / 7,021	83.84 / 7,066	**83.91** / 6,943	**84.07** / 6,513
VEFR ($\tau = 0$)	79.03 / **3,606**	80.24 / **3,770**	81.22 / **3,993**	82.03 / **3,983**	83.06 / **3,842**
VEFR + smoothing	83.16 / 6,698	83.50 / 7,009	**83.86** / 7,015	83.87 / 6,798	84.05 / 6,296

Class+global variant reaches 84.54% at 5,916 tokens/image; with the VEFR's matched weight $\eta = \gamma$, it reaches 84.44% at 6,195 tokens/image. The headline VEFR reaches 84.50% at 4,747 tokens/image. The global control can match accuracy, but local evidence achieves comparable accuracy with lower inference cost.

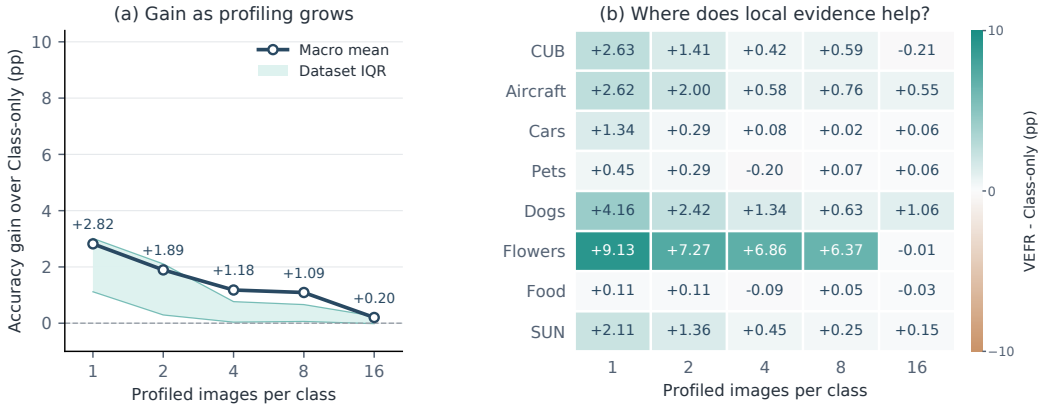


Figure 7: **Local-evidence gains without smoothing.** Paired Test-accuracy differences between unsmoothed VEFR and Class-only, first averaged over five class-balanced support galleries within each dataset. (a) Eight-dataset mean and interquartile range across dataset gains ("Dataset IQR"); (b) individual dataset gains. Values are percentage points.

Figure 7 shows that the contribution of local evidence is heterogeneous: its gain is largest in sparse settings and generally contracts as class statistics accumulate. Encoder-specific paired gains appear in Table 8; they do not change the recognition-skill outcomes.

E.2 Rebuilding profiles for a new executor

We change the skill-executing MLLM while keeping DINOv3 evidence, candidate generation, prompts, splits, and the Visual-ICL demonstration rule fixed. Each executor receives its own skill outcomes and rebuilt competence records. The four-skill condition removes Visual ICL before fitting; its Prompted VLM router uses a four-choice prompt. This tests applicability to a new executor, separately from reusing an old profile.

Figure 8 shows that VEFR reaches the highest accuracy among the evaluated routers for all four MLLMs with five skills, using 27–45% fewer tokens than the most accurate competing router within each executor. On Qwen3-VL-32B, the accuracy margin is only 0.04 points, so the benefit is primarily efficiency. Excluding Visual ICL lowers absolute accuracy but preserves the routing advantage among evaluated routers. The per-dataset outcomes underlying the figure are given in Tables 14–16;

accuracy and token cost should be compared within, rather than across, executors. In these tables, Air. and Flw. abbreviate Aircraft and Flowers.

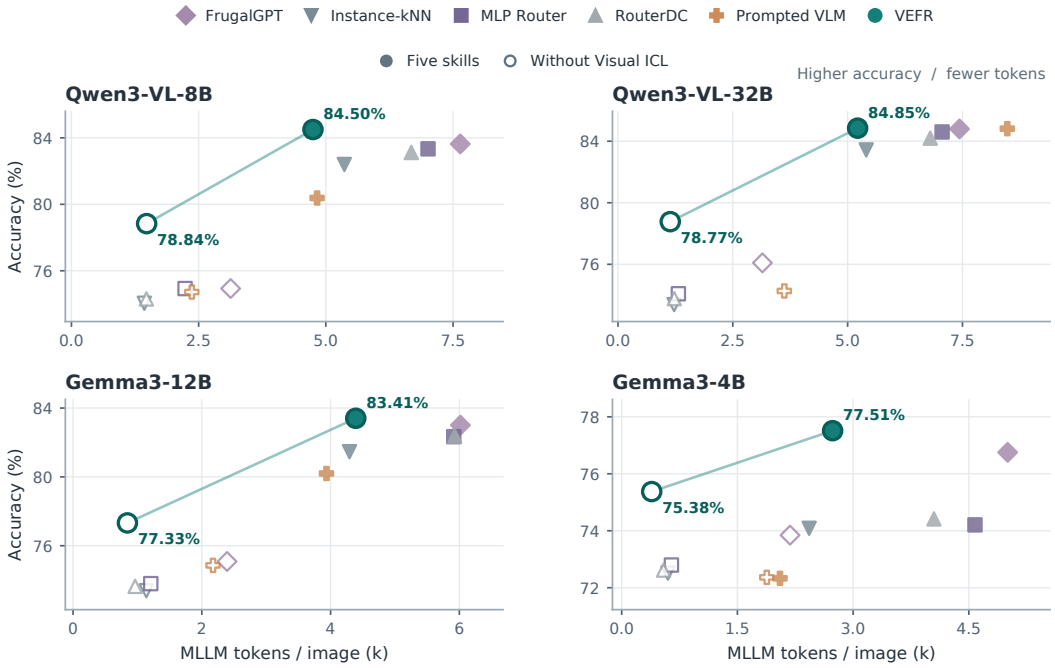


Figure 8: **Routing across MLLM executors and skill pools.** Each panel compares eight-dataset macro accuracy with realized MLLM tokens per image for one executor. Filled markers denote the five-skill pool and open markers exclude Visual ICL; each line joins the two VEFR settings.

E.3 Reusing competence across executors

Qwen3-VL-8B supplies a source class profile and local correctness records. Zero-shot transfer freezes those records and the source-selected K, T, and γ, changing only the skill executor. With target profiling, five paired class-balanced galleries contribute 1, 2, 4, 8, or 16 target skill outcomes per class. Target-only estimation shrinks the target class rates toward the target global rate from the same support, whereas the transfer variant uses the frozen source class profile as its prior. Both use $\tau = 4$ and the same target local gallery. The visual class gallery remains complete, and no target Validation outcomes are used for selection.

Table 13: **Cross-executor profile reuse under matched target support.** Eight-dataset macro accuracy (%). Columns 1–16 give profiled target images per class; 0 freezes the Qwen3-VL-8B profile without target outcomes, and Full uses all target training outcomes with source-frozen routing settings. Bold and underline rank matched-support methods within each target.

Target	Profile	0	1	2	4	8	16	Full
Qwen3-VL-32B	Frozen source	83.23	–	–	–	–	–	–
	Target-only	–	**83.64**	**83.75**	**84.01**	**84.30**	**84.61**	85.10
	Source class prior + target	–	83.42	83.61	83.95	84.28	84.55	–
Gemma3-12B	Frozen source	80.66	–	–	–	–	–	–
	Target-only	–	**81.19**	**81.65**	**82.17**	**82.50**	**82.85**	83.24
	Source class prior + target	–	81.07	81.58	82.01	82.36	82.77	–
Gemma3-4B	Frozen source	71.16	–	–	–	–	–	–
	Target-only	–	**74.16**	**74.73**	**75.48**	**76.21**	**76.78**	77.38
	Source class prior + target	–	73.06	74.15	75.27	76.07	76.75	–

Table 14: **Executor-specific results with Qwen3-VL-32B.** Test accuracy (%) and realized MLLM tokens/image; Avg. Acc. is the eight-dataset macro mean.

Method	Avg. Acc.	Tokens	CUB	Air.	Cars	Pets	Dogs	Flw.	Food	SUN
Fixed recognition skills										
Direct	68.47	**453**	61.77	39.21	71.15	82.83	55.28	78.44	86.70	72.40
Description	71.18	3,477	68.31	41.28	71.06	87.11	61.46	78.78	87.63	73.82
CoT	68.03	929	64.08	42.06	71.16	84.27	58.92	80.84	79.66	63.21
Visual ICL	**84.93**	7,001	84.60	65.05	86.81	92.40	77.26	**99.22**	92.59	**81.50**
Router baselines										
FrugalGPT	84.80	7,435	84.57	**65.53**	86.51	92.04	77.45	98.59	92.56	81.14
Instance-kNN	83.43	5,405	83.86	63.58	84.98	90.19	75.73	98.78	90.79	79.55
MLP Router	84.60	7,056	**84.62**	63.64	86.54	92.37	77.23	98.86	92.46	81.09
RouterDC	84.20	6,798	84.52	64.84	85.41	90.62	77.28	**99.22**	**92.60**	79.07
Prompted VLM	84.81	8,476	84.43	65.11	86.99	92.34	76.84	98.93	92.57	81.31
VEFR (Ours)										
VEFR (Accuracy-first)	84.85	5,217	83.26	65.29	**87.74**	**92.59**	**77.87**	98.60	92.57	80.86
VEFR (Utility)	84.39	3,930	82.50	65.05	87.35	92.34	77.58	97.53	92.24	80.56
VEFR (Batch budget)	84.45	4,174	82.69	64.72	87.25	92.01	77.55	98.28	92.46	80.61

Table 15: **Executor-specific results with Gemma3-12B.** Test accuracy (%) and realized MLLM tokens/image; Avg. Acc. is the eight-dataset macro mean.

Method	Avg. Acc.	Tokens	CUB	Air.	Cars	Pets	Dogs	Flw.	Food	SUN
Fixed recognition skills										
Direct	65.63	**409**	39.51	37.08	77.02	87.76	53.00	70.24	88.78	71.69
Description	64.47	3,425	36.74	38.07	77.04	84.25	47.59	71.17	88.76	72.18
CoT	62.46	766	41.39	38.67	73.83	85.53	52.94	69.88	80.39	57.04
Visual ICL	82.80	6,119	71.63	64.45	88.14	**93.98**	74.17	**97.95**	92.86	**79.24**
Router baselines										
FrugalGPT	83.01	6,017	72.32	**65.32**	88.16	**93.98**	74.31	**97.95**	92.89	79.13
Instance-kNN	81.46	4,294	71.97	62.98	85.87	92.56	72.34	97.20	91.32	77.47
MLP Router	82.34	5,913	72.21	64.03	87.80	93.65	73.96	95.12	92.88	79.06
RouterDC	82.37	5,928	72.13	64.00	88.06	92.97	73.99	96.26	93.05	78.52
Prompted VLM	80.20	3,937	72.20	63.58	84.87	90.02	72.56	91.69	90.65	76.02
VEFR (Ours)										
VEFR (Accuracy-first)	**83.41**	4,391	74.30	64.36	**88.60**	93.89	**76.11**	97.64	**93.12**	79.22
VEFR (Utility)	82.84	2,625	**74.58**	64.15	87.87	93.08	75.20	96.63	92.58	78.63
VEFR (Batch budget)	82.93	3,066	74.51	62.89	88.09	93.68	75.24	97.41	92.93	78.69

Table 16: **Executor-specific results with Gemma3-4B.** Test accuracy (%) and realized MLLM tokens/image; Avg. Acc. is the eight-dataset macro mean.

Method	Avg. Acc.	Tokens	CUB	Air.	Cars	Pets	Dogs	Flw.	Food	SUN
Fixed recognition skills										
Direct	52.32	**409**	19.99	29.40	66.88	69.12	38.59	59.80	74.63	60.19
Description	54.72	3,425	25.92	29.49	65.02	70.54	41.14	60.95	78.74	65.96
CoT	60.38	702	39.87	31.41	66.92	85.77	52.94	70.92	78.04	57.16
Visual ICL	71.14	6,119	55.06	46.86	80.71	84.57	57.05	90.05	87.90	66.95
Router baselines										
FrugalGPT	76.75	5,004	71.49	48.66	**83.56**	88.50	66.36	92.65	90.39	72.41
Instance-kNN	74.09	2,430	71.18	46.38	77.47	86.32	62.46	90.10	89.21	69.58
MLP Router	74.21	4,580	71.14	46.80	80.05	84.79	63.94	90.24	89.26	67.42
RouterDC	74.42	4,048	71.09	46.20	80.10	86.70	63.03	90.81	89.16	68.23
Prompted VLM	72.33	2,052	71.07	43.29	74.58	84.00	62.35	83.66	88.97	70.72
VEFR (Ours)										
VEFR (Accuracy-first)	77.51	2,734	71.80	50.35	82.34	**90.49**	**70.27**	93.02	90.50	71.35
VEFR (Utility)	**77.64**	1,284	72.01	**50.95**	81.81	90.30	70.10	**93.12**	90.22	**72.61**
VEFR (Batch budget)	77.48	1,496	**72.09**	50.68	81.63	90.38	69.95	91.93	**90.61**	72.59

The zero-shot and few-shot rows answer different questions: the latter replaces the source local cache with target outcomes. At matched target support, the transferred class prior does not consistently improve over building a target-only profile. By 16 shots, target-only accuracy is within 0.60 points of the matched Full reference while using 18.7% of its target profiling calls. Full here uses all target outcomes with source-frozen routing settings; it is not the separately selected target-specific result in Figure 8. The results show that class competence is a useful source of hypotheses, but cannot be assumed invariant across executors.

F Qualitative Analysis and Failure Modes

F.1 Population-level failure limits

We analyze the frozen full-profile routes on the eight Test sets, with equal weight per dataset. Figure 9(a) partitions queries by whether the Class-only and Local-only choices are correct. When only Class-only succeeds, VEFR preserves a correct choice for 80.67% of queries in that group; when only Local-only succeeds, it does so for 39.99%. The fixed fusion thus retains class-based successes more reliably than local-only successes in these disagreement groups. Neither component's failure implies that all skills fail: their selected skills can be wrong even when another skill is correct.

Figure 9(b) separates these routing misses from limitations of the available skill pool. The skill oracle reaches 93.42% macro accuracy. VEFR's 15.50-point error decomposes into 8.92 points where a correct skill was available but not selected, and 6.58 points where no skill succeeded. The oracle is a retrospective diagnostic rather than a deployable router.

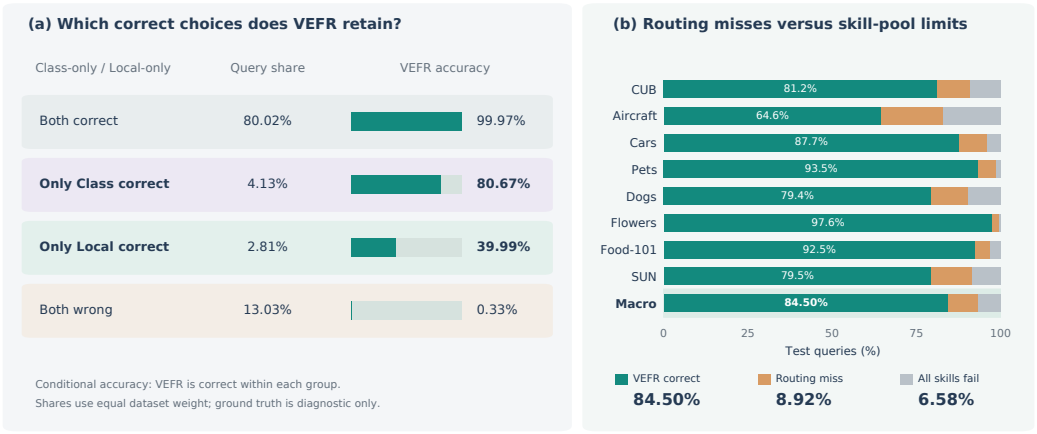


Figure 9: **Complementarity and limits of the skill pool.** (a) Query share and conditional VEFR accuracy for four Class-only/Local-only outcome groups. (b) Correct VEFR choices, routing misses with an available correct skill, and failure of all five skills. The highlighted Macro row and summary values use equal dataset weight; conditional accuracy is computed within each group under the same weighting.

The class posterior informs skill selection rather than directly determining the final label. A selected skill can therefore recognize the query correctly even when the highest-posterior class is incorrect. Figure 10(a,b) illustrates such recoveries.

F.2 Examples on selected benchmarks

Figures 11–13 extend the CUB examples in Figure 10 to Cars, Dogs, and Food-101. All four datasets use the same six routing situations. Panels (a,b) distinguish an incorrect top-ranked class from the final recognition outcome; both show nonzero true-class posterior mass, with the posterior gap measured in percentage points. Panels (c,d) illustrate complementary corrections by local and class evidence. Panel (e) exposes a fusion failure even though both components select correct skills, whereas panel (f) shows the limit of rerouting when no available skill succeeds. Each query is dis-

played with its three nearest profiling images, but routing uses the full neighborhood. Query labels and cached outcomes are used only for retrospective analysis. These selected examples illustrate mechanisms rather than their prevalence.

The CUB examples in Figure 10(a,b) have posterior gaps of 44.4 and 21.9 percentage points, respectively, yet the selected ICL and CoT skills recognize the queries correctly.

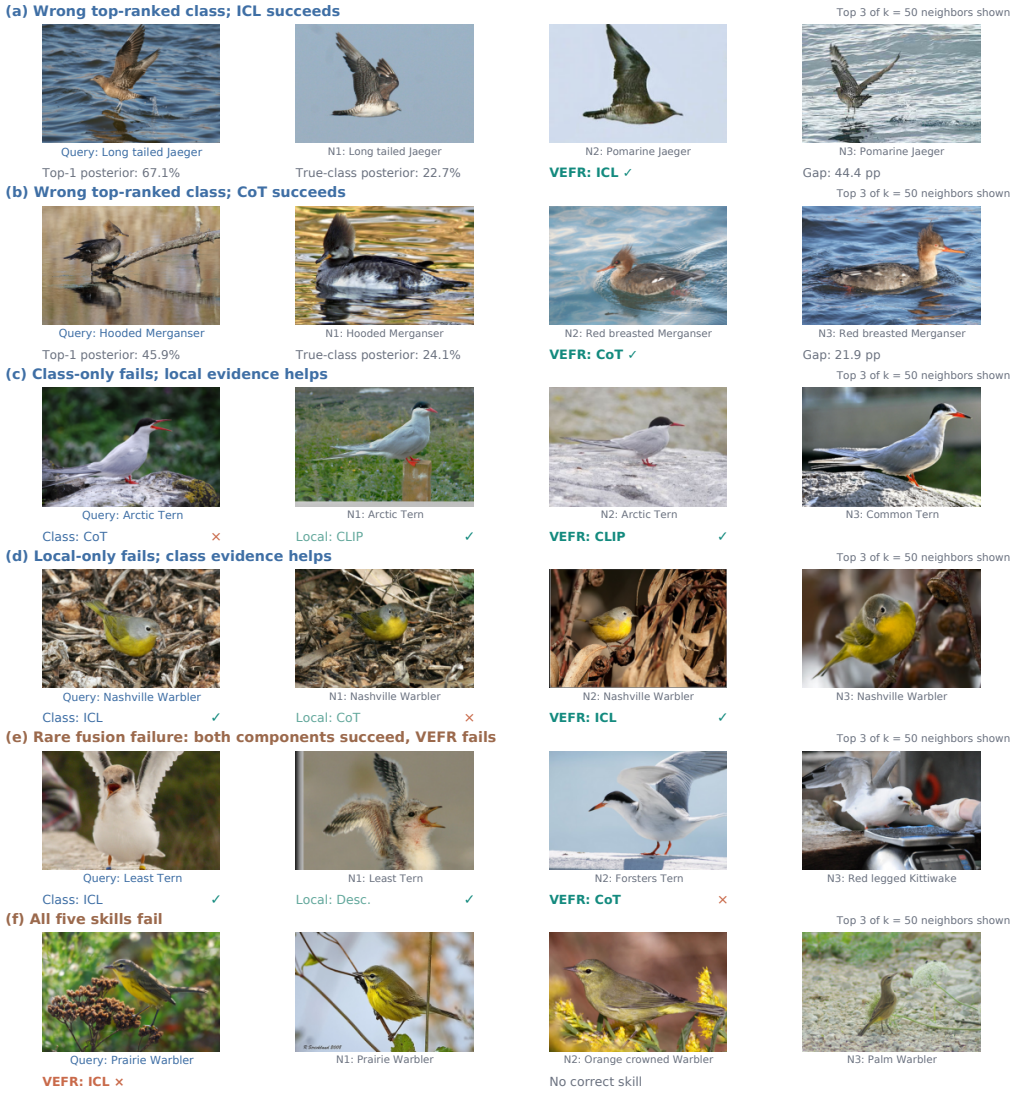


Figure 10: **Complementary evidence and failure cases on CUB.** Each query and its three nearest profiling images are labeled by class; routing uses the full neighborhood. Query labels are shown only for retrospective diagnosis. The frozen unsmoothed VEFR can recover from misleading class or local evidence, but may still miss an available correct skill or fail when no skill predicts correctly. ✓/× denote correct/incorrect routing outcomes.

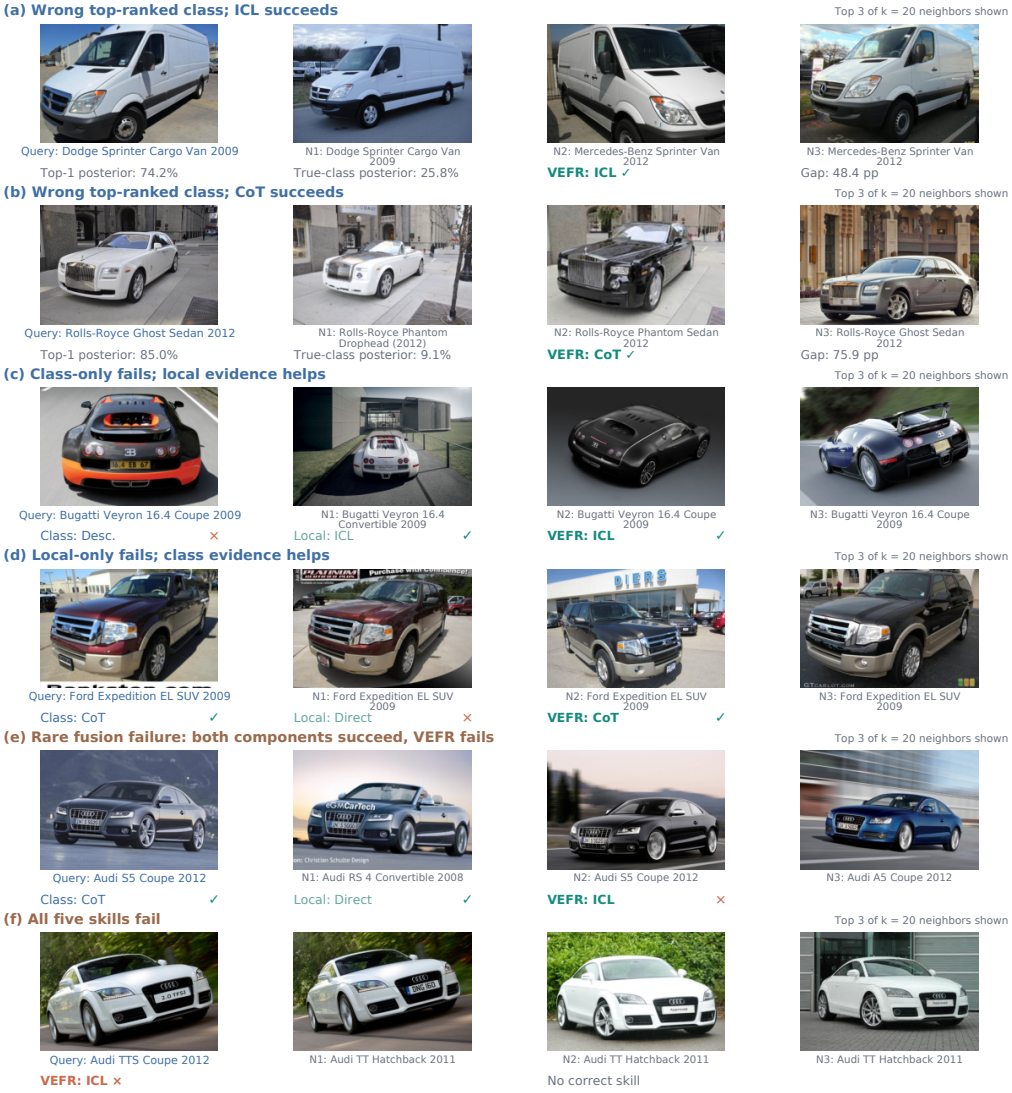


Figure 11: **Complementary evidence and failure cases on Cars.** The selected skill can recognize a query despite a strongly incorrect class preference; the remaining cases distinguish component corrections from fusion and skill-pool failures.

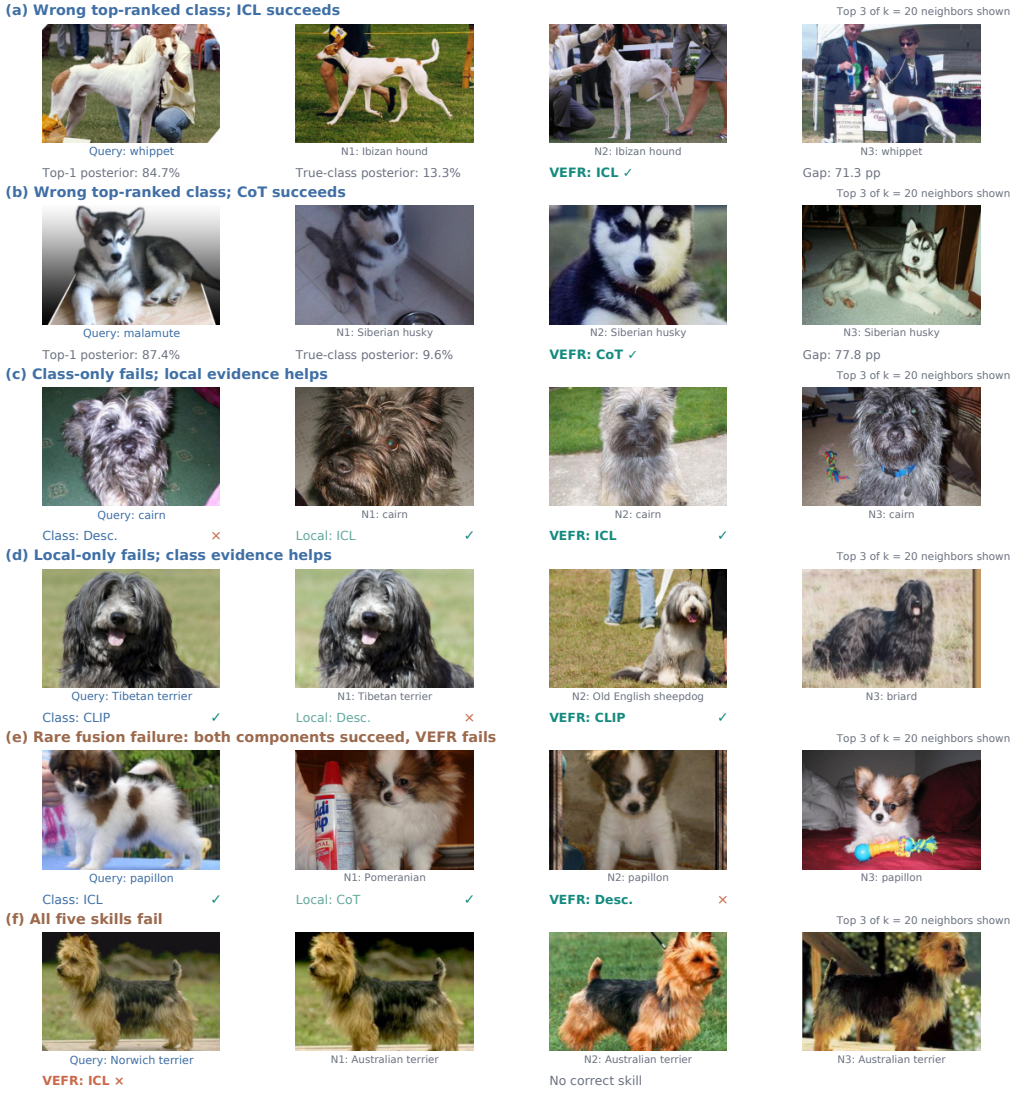


Figure 12: **Complementary evidence and failure cases on Dogs.** Similar appearance across breeds can mislead either evidence source. Their complementary corrections coexist with failures of fusion and of the available skill pool.

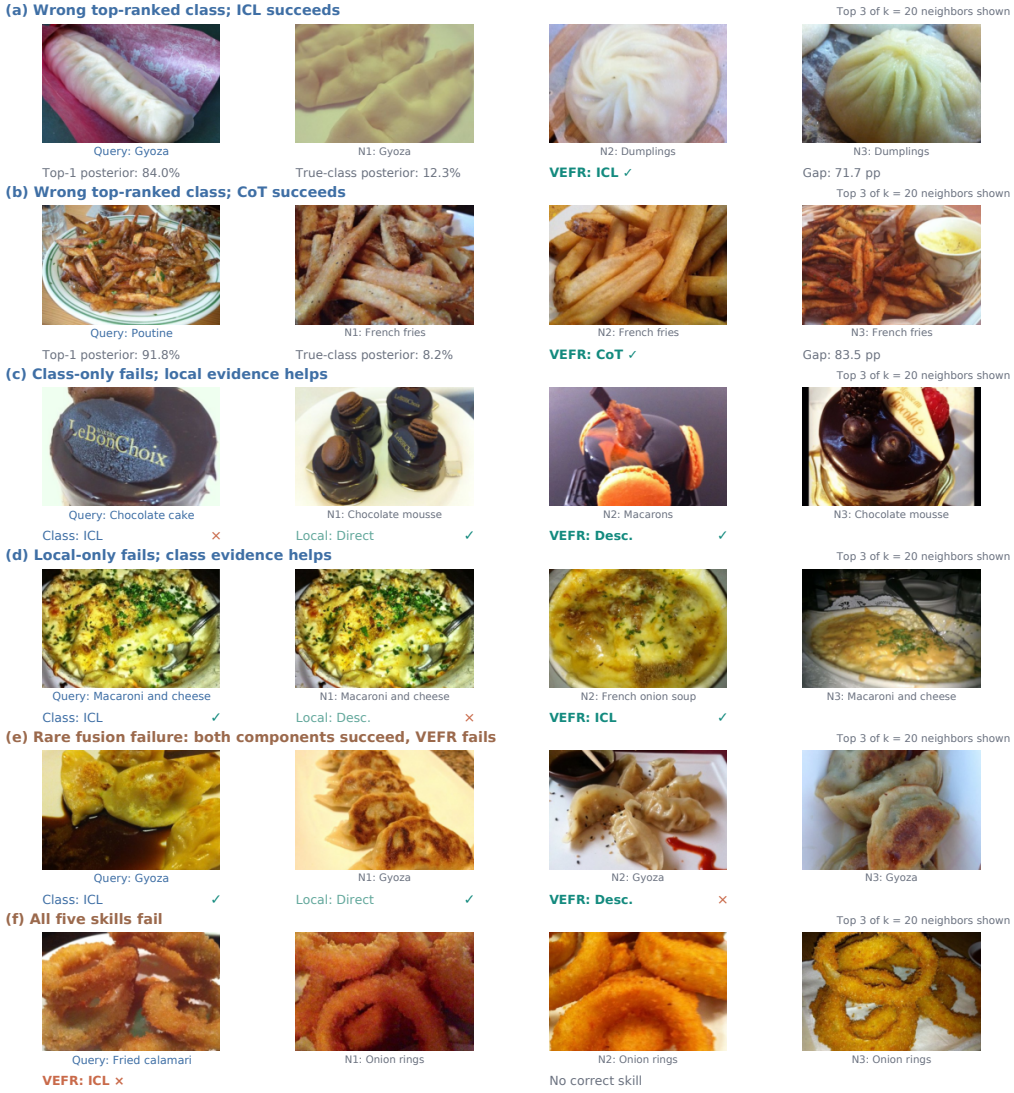


Figure 13: **Complementary evidence and failure cases on Food-101.** Recognition can succeed despite a wrong top-ranked food class. The component and failure cases distinguish useful local evidence from incorrect skill selection and failure of every skill.