LATA: Laplacian-Assisted Transductive Adaptation for Conformal Uncertainty in Medical VLMs

Behzad Bozorgtabar[1*] Dwarikanath Mahapatra[2] Sudipta Roy[3]
Muzammal Naseer[2] Imran Razzak[4] Zongyuan Ge[5]
[1]Aarhus University, Denmark [2]Khalifa University, UAE [3]Jio University, India
[4]MBZUAI, UAE [5]Monash University, Australia
[*]behzad@ece.au.dk

A. Algorithm of LATA

The end-to-end **LATA** procedure is summarized in Algorithm 1.

Algorithm 1 LATA

Inputs : Frozen VLM (ϕ, W); labeled calibration $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$; unlabeled test $\mathcal{D}_{\text{test}} = \{x_j\}_{j=1}^{m}$; frozen ViLU; error level α; nonconformity rule $S_{\text{base}} \in \{\text{LAC}, \text{APS}, \text{RAPS}\}$; hyperparameters $(k, \sigma, \gamma, T_{\text{iter}}, \lambda, \eta)$; optional prior (β, m).
Output: Prediction sets $\{\mathcal{C}(x_j)\}_{j=1}^{m}$.

$\mathcal{U} \leftarrow \mathcal{D}_{\text{cal}} \cup \mathcal{D}_{\text{test}}$; // joint pool
foreach $x \in \mathcal{U}$ **do**
  $q(x) \leftarrow p(W, \phi(x))$; // Eq. (1); zero-shot probs, $q(x) \in \Delta^{C-1}$
; // **Block 1: LATA (deterministic transductive refinement)**
Build a *symmetric* kNN graph W^{g} on $\{\phi(x)/\|\phi(x)\|_2 : x \in \mathcal{U}\}$ using union; weights via Gaussian kernel with bandwidth σ ; // Sec. 3.4
if $\beta > 0$ **then**
  foreach $x \in \mathcal{U}$ **do**
    $q(x) \leftarrow \text{Renorm}\big(q(x) \odot m^{\beta}\big)$; // $[q_{ik} \leftarrow q_{ik} m_k^{\beta} / \sum_{\ell} q_{i\ell} m_{\ell}^{\beta}]$
Initialize $\tilde{z}^{(0)}(x) \leftarrow q(x)$ for all $x \in \mathcal{U}$ **for** $t = 1$ **to** T_{iter} **do**
  foreach $x_i \in \mathcal{U}$ **do**
    for $c = 1$ **to** C **do**
      $m_{ic} \leftarrow \sum_j W_{ij}^{\text{g}} \tilde{z}_{jc}^{(t-1)}$ $\tilde{z}_{ic}^{(t)} \leftarrow q_{ic} \exp\big(\gamma\, m_{ic}\big)$
    $\tilde{z}_i^{(t)} \leftarrow \tilde{z}_i^{(t)} / \sum_{c=1}^{C} \tilde{z}_{ic}^{(t)}$; // row-normalize
Set $\tilde{z}(x) \leftarrow \tilde{z}^{(T_{\text{iter}})}(x)$ for all $x \in \mathcal{U}$; // Eq. (6)
; // **Block 2: Failure-aware conformal prediction**
foreach $x \in \mathcal{U}$ **do**
  $(u(x), \alpha(x)) \leftarrow \text{ViLU}(x)$; // frozen head; identical on cal & test
for $i = 1$ **to** n **do**
  $s_i \leftarrow S_{\text{base}}(\tilde{z}(x_i), y_i)\,\big(1 + \lambda\, u(x_i)\big) \;-\; \eta\, \alpha_{y_i}(x_i)$; // Eq. (8)
$\hat{s} \leftarrow$ empirical $(1-\alpha)$ quantile of $\{s_i\}_{i=1}^{n}$; // Eq. (2)
foreach $x_j \in \mathcal{D}_{\text{test}}$ **do**
  $\mathcal{C}(x_j) \leftarrow \big\{\, y \in \{1, \ldots, C\} : \; S_{\text{base}}(\tilde{z}(x_j), y)\big(1 + \lambda u(x_j)\big) - \eta\, \alpha_y(x_j) \le \hat{s} \,\big\}$
return $\{\mathcal{C}(x_j)\}_{j=1}^{m}$

B. Related Work

Transfer learning in VLMs. Foundation VLMs deliver strong zero-shot recognition, but performance can deteriorate when downstream concepts are under-represented during pretraining, motivating few-shot transfer. Popular strategies include *prompt learning* (optimizing task-specific textual tokens) [14, 17] and *black-box adapters*/linear probes that operate on frozen embeddings; the latter often yields competitive accuracy at lower compute by blending visual and text logits (e.g., constrained or text-informed probes such as CLAP [8] or LP++ [3]). While effective, these methods are typically *inductive*: they adapt from the labeled calibration split but do not exploit the unlabeled test distribution, and—when the same labels are reused for conformal calibration—can compromise exchangeability. Recent *transductive* black-box approaches (e.g., TransCLIP [16]) incorporate unlabeled test data via simple generative priors, improving accuracy but leaving coverage guarantees unaddressed. Our work departs from accuracy-only transfer by targeting *reliable* few-shot adaptation: **LATA** is label-free, training-free, and transductive, and we couple it with a failure-aware conformal score—preserving split-conformal validity while improving set efficiency and class-wise balance at fixed coverage.

Conformal prediction with VLMs. Conformal prediction (CP) [2, 12] provides finite-sample coverage guarantees; in vision, it is commonly used in its split form SCP [5, 13] with black-box models and standard nonconformity scores (LAC [7], APS [6], RAPS [1]). However, these methods rely on models trained under the assumption that the training and test data are independently and identically distributed (i.i.d.)—an assumption that does not hold for pre-trained VLMs, which are the focus of our work. Extending CP beyond image-only classifiers to VLMs is recent: Conf-OT [9] rebalances zero-shot logits via opti-

Table S1. **Complexity & compute profile**. Asymptotics in $N{=}n{+}m$, C, K, d.

Method	Labels at transfer?	Training?	Per-query refits	Dominant complexity	Memory	Time/img (s)†	SCP validity
LATA (ours)	No	No	None	$O(N \log N \cdot d) + O(T_{\text{iter}}\, k\, N\, C) + O(nC)$	$O(kN{+}NC)$	**0.05–0.06**	Yes
SCA-T	No	Yes (unsup.)	None	$O(T'NC) + O(nC)$	logits+buffers	1.04–1.15	Yes
Conf-OT	No	No	None	$O(S\,K\,(N{+}m)) + O(nC)$	OT matrices	0.60–0.70	Yes
FCA	**Yes** (cal.)	**Yes**	$O(C)$	$O(C{\times}\text{fit\_cost}) + O(nC)$	probe heads+logits	— (solver-dep.)	Yes

†Measured on our setup (same backbone, prompts, hardware) and averaged over LAC/APS at $\alpha{=}0.10$.

mal transport before SCP, improving size but offering limited control of class-wise balance; FCA [11] performs full-conformal adaptation with labeled calibration data. To preserve exchangeability, it evaluates test queries via per-label refits—effectively $\mathcal{O}(C)$ gradient-based updates per query/window. This delivers strong accuracy and compact sets, but at the expense of heavy compute, higher latency, and a departure from the black-box setting; and SCA-T [10] regularizes predictions transductively on the unlabeled pool, improving coverage/efficiency without labels but without leveraging multimodal failure/plausibility cues. Our method differs by being *label-free, training-free, and black-box*: it smooths zero-shot probabilities over an image–image kNN graph built on the joint calibration and test pool and augments SCP with a vision–language uncertainty head for failure-aware scoring. The transform is deterministic and applied identically to both splits (preserving SCP validity), while delivering smaller sets and lower CCV at the desired coverage.

C. Computational Complexity

In this section, we compare the computational profile of **LATA** against competitive baselines. Let $n{=}|\mathcal{D}_{\text{cal}}|{=}C{\times}K$, let m denote the number of test samples processed together, and let $N{=}n{+}m$ be the joint pool size used for transductive refinement. Let d be the embedding dimension, k the kNN degree, and T_{iter} the number of mean-field (CCCP) [15] passes.

LATA is *training-free* and *label-free* at transfer time: it applies a deterministic refinement to the joint pool, then performs standard conformal prediction over all C classes. This preserves exchangeability and SCP validity while enabling transductive adaptation.

Complexity of LATA:

- **Graph build:** approximate kNN on ℓ_2-normalized embeddings: $O(N \log N \cdot d)$ time; $O(kN)$ memory.
- **LATA updates (mean-field / CCCP):** sparse neighbor aggregation over all C labels: $O(T_{\text{iter}}\, k\, N\, C)$ time; $O(kN{+}NC)$ memory; no backprop.
- **Failure-aware head:** lightweight per-sample forward over C labels: $O(NC)$ time; $O(NC)$ memory.
- **Conformalization (exact):** compute calibration scores over C labels ($O(nC)$; APS/RAPS add $O(C \log C)$ for sorting), then test set filtering $O(mC)$.

Protocol (joint-pool, transductive). Given calibration $\mathcal{D}_{\text{cal}}$ and a batch of m test queries, form the joint pool of size $N{=}n{+}m$, build a symmetric kNN graph on ℓ_2-normalized embeddings, run T_{iter} mean-field passes, compute the $(1{-}\alpha)$ quantile on the calibration subset, and conformalize the m queries with this threshold. In our runs we use a fixed budget ($k{=}15$, $T_{\text{iter}}{=}8$); time/memory numbers in Table S1 reflect this setting.

Contrasts to baselines (same backbones/prompts). SCA-T [10] performs unsupervised entropy minimization on the joint pool (calibration+test), regularized by a label-marginal prior. Its solver operates at the logits level and incurs a per joint-pool complexity of $O(T'NC)$, where T' is the number of optimization steps, followed by standard conformal scoring at $O(nC)$.

Conf-OT [9] rebalances zero-shot logits using Sinkhorn optimal transport [18] over a $K{\times}N$ similarity matrix (with K calibration shots per class and $N{=}n{+}m$ total items in the joint pool). The dominant cost per joint-pool pass is $O(SKN)$ for S Sinkhorn iterations, plus $O(nC)$ for conformalization. With small S and K, the wall-clock overhead remains low.

FCA [11] performs full conformal adaptation using labeled calibration data. For each class y, it fits a linear probe (via SS-Text [11]) in a transductive conformal adaptation loop. Test-time inference entails $O(C)$ separate refits per joint pool, plus $O(nC)$ for scoring. Although gradients are avoided by the SS-Text optimizer [11], runtime and memory scale linearly with C, making FCA less efficient than single-solver black-box methods like **LATA**.

As shown in Table S1, **LATA**'s per–joint-pool cost is dominated by (i) approximate kNN graph construction on ℓ_2-normalized embeddings and (ii) T_{iter} sparse mean-field passes over that graph. Conformalization then adds an exact $O(nC)$ to compute the calibration quantile and $O(mC)$ to apply it to the m queries (with $N = n{+}m$ total items in the joint pool). **LATA** requires no gradients, no $O(C)$ per-class refits, and uses a single deterministic, symmetry-preserving transform on calibration and test, maintaining SCP validity. Compared to SCA-T and Conf-OT, **LATA** avoids dense logits-level optimization or optimal-transport solves, relying instead on lightweight neighbor aggregation with a frozen failure-aware head—while keeping black-box

Table S2. **Sensitivity to failure-aware weights** (λ, η) (APS, $\alpha = 0.10$, LATA-LF). All configurations remain near the nominal coverage (≈ 0.90) and retain clear gains in set size and CCV compared to SCP (Size $= 4.05$, CCV $= 9.59$) and SCA-T (Size $= 3.35$, CCV $= 7.18$).

λ	η	Cov.	Size ↓ / CCV ↓
0.25	0.10	0.897	2.90 / 6.55
0.25	0.25	0.898	2.88 / 6.40
0.25	0.50	0.895	2.82 / 6.50
0.50	0.10	0.900	2.96 / 6.45
0.50	**0.25**	**0.900**	**2.95 / 6.32**
0.50	0.50	0.898	2.88 / 6.35
1.00	0.10	0.907	3.08 / 6.25
1.00	0.25	0.907	3.07 / 6.15
1.00	0.50	0.905	3.00 / 6.20

usage and conformal guarantees. Versus FCA, **LATA** removes the cost of repeated per-class refits yet still conformalizes over the full label space without sacrificing coverage.

D. Additional Ablation Studies

D.1. Sensitivity to failure-aware weights

We study the robustness of the failure-aware score (Eq. (8) of the main manuscript) by varying the difficulty weight λ and plausibility weight η around our default setting $(\lambda, \eta) = (0.5, 0.25)$, while keeping all other hyperparameters fixed (APS, $\alpha = 0.10$, **LATA-LF**, averaged over all tasks). Table S2 reports marginal coverage, average set size, and CCV.

Across $(\lambda, \eta) \in [0.25, 1.0] \times [0.10, 0.50]$, coverage remains tightly concentrated around 0.90 (max deviation ≤ 0.01). Larger λ mildly raises coverage and Size while improving CCV (e.g., λ: 0.25→1.0 at η=0.10: Cov. 0.897→0.907, Size 2.90→3.08, CCV 6.55→6.25). Increasing η tends to shrink sets with negligible impact on coverage, with CCV flat to slightly worse at very high η; the mid-range $\eta \approx 0.25$ is strongest. Our default $(\lambda, \eta) = (0.5, 0.25)$ lies near the center of this Pareto region, offering a robust efficiency–validity balance without task-specific tuning.

D.2. Sensitivity to Laplacian weight

The hyperparameter γ controls the strength of the graph–smoothness term in Eq. (5) of the main manuscript: small γ keeps the refined distributions close to the zero-shot predictor, while larger values enforce stronger agreement among neighbors on the kNN graph. We vary $\gamma \in \{0.20, 0.35, 0.50\}$ for **LATA-LF** (APS, α=0.10), keeping all other hyperparameters fixed.

As presented in Table S3, across $\gamma \in [0.20, 0.50]$, coverage remains tightly clustered around the nominal 0.90, and all settings yield substantially smaller, more class-balanced sets than SCP and SCA-T. Variation in efficiency and CCV is mild: weaker smoothing (γ=0.20) slightly increases CCV, while stronger smoothing (γ=0.50) produces only marginal changes in Size and CCV. The default γ=0.35 provides a robust middle ground; **LATA-LF** does not require task-specific tuning of this parameter to maintain its coverage and fairness gains.

Table S3. **Sensitivity to Laplacian weight** γ (APS, α=0.10, LATA-LF). All settings stay near nominal coverage and retain clear gains in Size/CCV vs. SCP (Size $= 4.05$, CCV $= 9.59$) and SCA-T (Size $= 3.35$, CCV $= 7.18$).

γ	Cov.	Size ↓	CCV ↓
0.20	0.899	3.00	6.55
0.35	**0.900**	**2.95**	**6.32**
0.50	0.901	2.98	6.30

Table S4. **Sensitivity to graph degree** k (APS, α=0.10, **LATA-LF**). All settings remain near nominal coverage and maintain clear gains over SCP (Size $= 4.05$, CCV $= 9.59$) and SCA-T (Size $= 3.35$, CCV $= 7.18$).

k	Cov.	Size ↓	CCV ↓
10	0.899	3.02	6.60
15	**0.900**	**2.95**	**6.32**
20	0.900	2.97	6.35

D.3. Sensitivity to graph degree

The parameter k controls the sparsity of the image–image kNN graph: smaller k yields weaker (more local) smoothing, while larger k increases connectivity and the strength of transductive propagation. We vary $k \in \{10, 15, 20\}$ for **LATA-LF** (APS, α=0.10), keeping all other hyperparameters fixed.

As shown in Table S4, across $k \in [10, 20]$, **LATA-LF** consistently stays near the nominal coverage (≈ 0.90) while substantially improving Size and CCV over SCP and SCA-T. Sparser graphs (k=10) slightly weaken the smoothing effect, yielding marginally larger sets and higher CCV; denser graphs (k=20) produce only tiny changes relative to the default. Overall, k=15 is a robust operating point, and **LATA-LF** does not require task-specific tuning of k to retain its coverage and fairness gains.

D.4. Compute–accuracy trade-off

Fig. S1 analyzes **LATA** as we vary the number of mean-field passes $T_{\text{iter}} \in \{4, 8, 12\}$ at $\alpha = 0.10$. Our default choice $T_{\text{iter}} = 8$ (approximately 0.06 s/img and 0.80 GB) achieves nominal coverage with strong efficiency (APS: size 3.10,

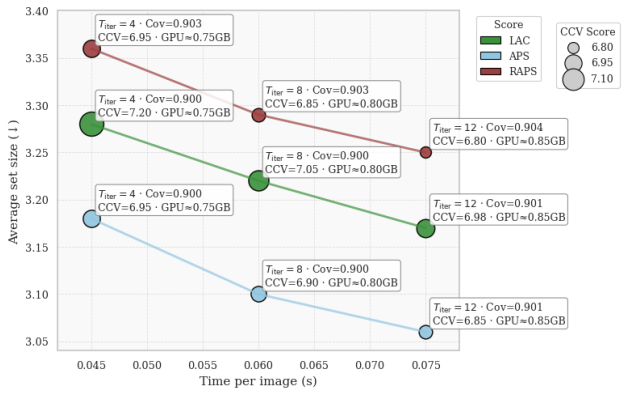

Figure S1. **Compute–accuracy trade-off for LATA** at α=0.10. Time per image (x-axis) vs. average set size (y-axis) for $T_{\text{iter}} \in \{4, 8, 12\}$. Colors denote LAC/APS/RAPS; marker size encodes CCV. Annotations show Cov., CCV, and GPU memory. Default T_{iter}=8 balances speed and reliability; T_{iter}=4 is faster with mild trade-offs; T_{iter}=12 yields limited gains.

CCV 6.90; LAC: 3.22/7.05; RAPS: 3.29/6.85).

Reducing to $T_{\text{iter}} = 4$ (< 0.045 s/img) speeds up inference by about 25% with a mild increase in Size/CCV (e.g., APS: Δ+0.08 Size, Δ+0.05 CCV). Increasing to $T_{\text{iter}} = 12$ yields only marginal gains (APS: 3.06/6.85). Across scores, APS is the most size-efficient; RAPS attains the lowest CCV and slightly higher coverage (≈ 0.903) at the cost of larger sets, while LAC lies between them. Timings were measured on a single RTX 4090 with identical backbones and prompts.

D.5. Temperature Sensitivity

Conformal inference is closely related to other uncertainty quantification frameworks—particularly calibration. Prior work often incorporates post-hoc calibration techniques such as temperature scaling to sharpen or smooth predictive probabilities before score computation (e.g., [1]). Temperature scaling rescales logits in Eq. (1) of the main manuscript by a factor τ, which can change score tails and thus efficiency (average set size) even when coverage remains valid. We therefore assess robustness to τ by applying the same scaling to both calibration and test (preserving exchangeability) and measuring conformal sets at α=0.10.

As shown in Fig. S2, for LAC the size curve is mildly U-shaped with small variation across $\tau \in [0.6, 1.4]$. For APS and RAPS, larger τ softens distributions and increases set size, as expected. Crucially, **LATA-LF** stays strictly below the Base (SCP) curve across all τ while maintaining nominal coverage, indicating that the graph-based refinement is robust to temperature perturbations. We default to τ=1.0 (main draft) and do not tune τ on target domains; performance is stable for $\tau \in [0.8, 1.2]$, and compute overhead is unaffected by τ.

Table S5. **Component ablation (APS)** at $\alpha \in \{0.10, 0.05\}$.

Variant	ACA↑	$\alpha = 0.10$			$\alpha = 0.05$		
		Cov.	Size↓	CCV↓	Cov.	Size↓	CCV↓
SCP	50.2	0.900	4.05	9.59	0.952	4.88	5.54
u-only	50.6	0.900	3.98	9.15	0.952	4.82	5.30
α-only	50.8	0.900	3.55	8.80	0.952	4.62	5.15
LATA (no ViLU)	55.6	0.900	3.05	6.60	0.954	3.88	3.70
LATA (ours)	**57.1**	**0.900**	**2.95**	**6.32**	**0.954**	**3.78**	**3.55**

Table S6. **Sensitivity to the label-informed prior** (β). We apply the prior as $q \leftarrow \text{Renorm}(q \odot m^{\beta})$ (equivalently, logits $z \leftarrow z + \beta \log m$), where m are Dirichlet-smoothed calibration marginals (fixed once) and the same transform is applied to calibration and test. Numbers are averaged across tasks.

		LAC, α=0.10			**LAC**, α=0.05		
β	**ACA**↑	Cov.	Size↓	CCV↓	Cov.	Size↓	CCV↓
0.0	57.0	0.900	**3.07**	6.40	0.952	**3.76**	**3.35**
0.1	57.2	0.905	3.11	6.32	0.958	3.82	3.40
0.2	**57.4**	**0.910**	3.15	**6.25**	**0.962**	3.86	3.40
0.3	57.4	0.914	3.20	6.22	0.966	3.92	3.45

Table S7. **Prior sensitivity for APS**. Same prior application and setup as Table S6.

		APS, α=0.10			**APS**, α=0.05		
β	**ACA**↑	Cov.	Size↓	CCV↓	Cov.	Size↓	CCV↓
0.0	57.1	0.900	**2.95**	6.32	0.954	**3.78**	3.55
0.1	57.3	0.905	2.99	6.28	0.959	3.83	3.50
0.2	**57.5**	**0.910**	3.03	**6.25**	**0.963**	3.88	**3.45**
0.3	57.5	0.914	3.08	6.23	0.966	3.93	3.44

D.6. Component analysis

The APS ablation in Table S5 mirrors the LAC trend: **LATA** without ViLU [4] already secures most of the coverage–efficiency gains (Size 3.05, CCV 6.60 at α=0.10), and the full **LATA (ours)** variant delivers the strongest results while keeping coverage at nominal. The u and α terms provide complementary benefits—u reduces CCV with minimal size change, whereas α shrinks sets modestly—yet their combination with the graph refinement yields the best overall trade-off.

D.7. Sensitivity to label-informed prior

The label-informed prior β serves as a gentle coverage–efficiency knob. As shown in Table S6 and Table S7, increasing β tightens marginal coverage with a small increase in set size. CCV improves for LAC at α=0.10 and for APS at both error levels, while for LAC at α=0.05

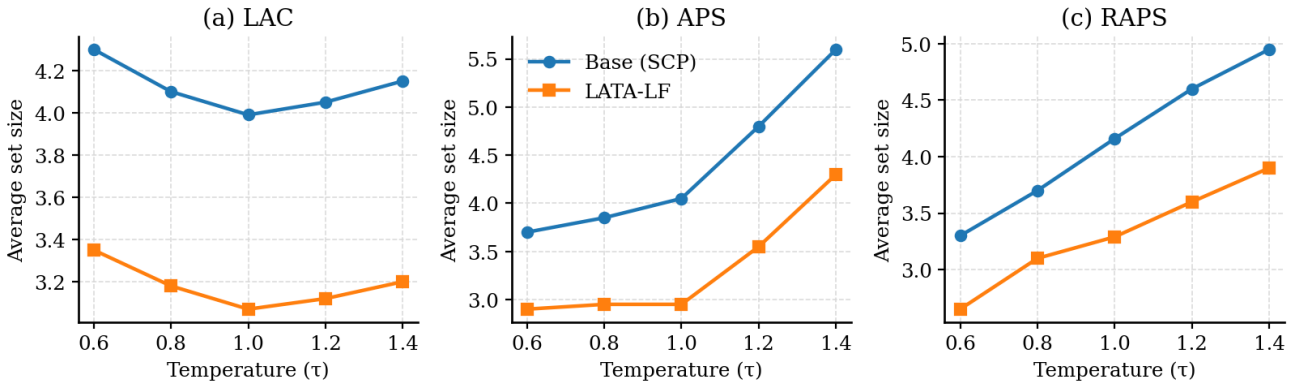

Figure S2. **Temperature sensitivity** at α=0.10. Average set size versus temperature for LAC, APS, and RAPS under Base (SCP) and **LATA-LF**. **LATA-LF** preserves nominal coverage and remains strictly more efficient than Base across $\tau \in [0.6, 1.4]$.

the fairness gains are neutral or slightly negative due to the tighter target. We therefore report **LATA-LF** (β=0) as the default and **LATA-LI** (β=0.2) as a practical, validity-preserving variant; the prior is applied once and identically to calibration and test, preserving exchangeability.

D.8. ViLU pretraining source

In Table S8, we ablate the source domain used to pretrain the frozen ViLU head that provides the difficulty $u(x)$ and plausibility $\alpha(x)$ signals in the failure-aware score (Eq. (8), main manuscript). *Matched ViLU* denotes a head pretrained on data with the *same modality and label granularity* as the target task (e.g., histopathology for SICAPv2), while *Mismatched ViLU* is pretrained on a *different* modality/dataset family (e.g., chest X-ray). *No ViLU* disables these signals by setting $(\lambda, \eta) = (0, 0)$. Under a fixed unsupervised transductive (UT) setup (N=256, k=15, T_{iter}=8), and LAC at α=0.10, **coverage remains nominal** in all cases. *Matched ViLU* achieves the **smallest sets** and **lowest CCV**; using a mismatched source causes only a small efficiency/fairness drift ($\sim$+0.05 size, $\sim$+0.12 CCV), while removing ViLU has a clearer impact ($\sim$+0.15 size, $\sim$+0.50 CCV) at identical compute. Unless otherwise stated, we use the default weights $(\lambda, \eta) = (0.5, 0.25)$ and β=0 (**LATA-LF**).

Table S8. **Effect of ViLU pretraining source** (LAC, α=0.10). UT window: N=256, k=15, T_{iter}=8; averaged over random seeds.

Setting	Cov.	Size↓	CCV↓	T (s/img) / GPU (GB)
LATA (ours) (matched ViLU)	0.900	**3.07**	**6.40**	0.05 / 0.80
LATA (ours) (mismatched ViLU)	0.900	3.12	6.52	0.05 / 0.80
LATA (ours) (no ViLU: λ=η=0)	0.900	3.22	6.90	0.05 / 0.80

D.9. Resource-aware gating

We study a simple speed–memory knob that skips the mean-field/CCCP refinement for *easy* inputs, identified by a low ViLU risk $u(x)$. Specifically, for any x with $u(x) < \tau_u$, we bypass the update in Eq. (6) of the main manuscript and pass the base probabilities through unchanged (i.e., $\tilde{z}(x) = q(x)$ after any fixed prior bias), while applying the standard refinement to the remaining samples. The gating rule is applied *identically* to calibration and test, preserving exchangeability and split-conformal validity.

Table S9 reports results under our fixed UT setup (N=256, k=15, T_{iter}=8), using LAC at α=0.10. Increasing τ_u reduces compute (time $\downarrow$10–20%, GPU $\downarrow$12.5–18.8%) while maintaining nominal coverage and inducing only modest changes in Size/CCV.

Table S9. **Resource-aware u-gating under LATA (ours)** (LAC, α=0.10). Averaged over random seeds; UT: N=256, k=15, T_{iter}=8. We skip CCCP updates for inputs with $u(x) < \tau_u$. Raising τ_u lowers time/memory (e.g., τ_u=0.30: T $-$20%, GPU $-$18.8%) with **no coverage loss**.

τ_u	Cov.	Size↓	CCV↓	T (s/img) / GPU (GB)
0.00 (no gating)	0.900	**3.07**	**6.40**	0.050 / 0.80
0.20	0.900	3.12	6.48	0.045 / 0.70
0.30	0.900	3.16	6.55	0.040 / 0.65

E. Additional Qualitative Results

Qualitative comparison with FCA on Gleason grading. Fig. S3 contrasts FCA [11] and **LATA (ours)** on SICAPv2 Gleason grading (APS, α=0.10), showing per-grade set-size distributions (top) and class co-occurrence within prediction sets (bottom). Both methods behave sensibly—extreme grades NC/G5 are mostly singletons—yet **LATA (ours)** consistently shifts probability

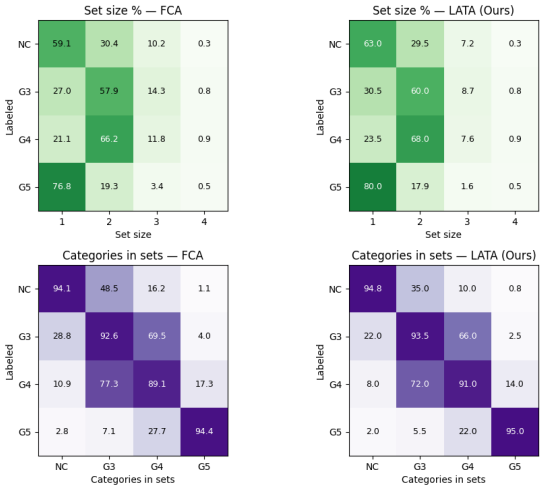

Figure S3. **Qualitative comparison of prediction sets on Gleason grading** (SICAPv2, APS, α=0.10). **LATA (ours)** produces smaller sets than FCA—especially for ambiguous G3/G4—while preserving strong diagonals and clinically plausible adjacency (G3↔G4) and further suppressing unlikely NC↔G5 co-occurrences.

mass from large to small sets, especially for the ambiguous mid-grades G3/G4 (e.g., size-3 frequency drops from roughly 14–12% with FCA to 9–8% with **LATA (ours)**). In the co-occurrence maps, **LATA (ours)** maintains strong diagonals (NC/G3/G5 $\geq$ 95%, G4 $\approx$ 91%) and preserves clinically plausible adjacency (frequent G3↔G4 co-appearance), while further suppressing *far* pairs such as NC↔G5 and G3/G4↔G5 compared to FCA. Overall, **LATA (ours)** produces more selective, adjacency-focused uncertainty—smaller sets and reduced spurious co-occurrences—while retaining the grading structure captured by FCA.

Coverage analysis across datasets. Fig. S4 and Fig. S5 show that **LATA** stays closest to the nominal target $(1-\alpha)$ while yielding compact sets across all nine datasets (LAC, α=0.10). In Fig. S4, **LATA**'s distributions are tightly concentrated at or slightly to the right of the dashed line (validity with low dispersion), whereas SCP is roughly centered but wider (higher CCV), and Adapt+SCP is systematically left-shifted (under-coverage), most visibly on *SICAPv2*, *MMAC*, and *COVID*. SCA-T narrows dispersion relative to SCP but remains broader and slightly left-biased compared to **LATA**. Fig. S5 corroborates this: across datasets, **LATA** occupies the desirable top-left frontier—matching or exceeding target coverage with smaller sets than SCP/SCA-T and without the exchangeability violation of Adapt+SCP. Notably, on long-tailed *NIH* it reduces set size substantially while keeping coverage near 0.91–0.93; on *CheXpert*, *MES-SIDOR*, and *FIVES* it attains high coverage with sets around 1–2; and on the harder *COVID* shift it pulls coverage back toward the target with markedly smaller sets. Overall, these figures illustrate the same trend: **LATA** improves efficiency and class-wise balance while remaining reliably on target.

F. Scoring Rules, Evaluation Metrics, and Dataset Details

Setup and notation. Let $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$ be the labeled calibration split and $\mathcal{D}_{\text{test}} = \{x_j\}_{j=1}^{m}$ the unlabeled test pool; assume exchangeability. Let $z(x) \in \Delta^{C-1}$ denote the class–probability vector used for conformal scoring (either zero-shot $q(x)$ or **LATA**-refined $\tilde{z}(x)$). A nonconformity score $S(x, y) \in \mathbb{R}$ quantifies how incompatible label y is for input x (*larger is worse*). The global threshold $\hat{s}$ is the split-conformal $(1-\alpha)$ quantile defined in Eq. (2) of the main manuscript. Given the threshold $\hat{s}$ computed from the calibration set, the prediction set for a test input x is defined as $\mathcal{C}(x) = , y \in 1, \ldots, C : S(x, y) \leq \hat{s},$. For each input x, let π_x denote the permutation that sorts the score vector $z(x)$ in descending order, and let $\text{rank}_x(y) \in 1, \ldots, C$ represent the rank position of label y under this ordering.

Least–Ambiguous Classifier (LAC). LAC [7] constructs conformal prediction sets by ranking class probabilities and selecting the most confident labels up to a fixed threshold. The underlying nonconformity score is defined as:

$$\mathcal{S}_{\text{LAC}}(x, y) = 1 - z_y(x), \tag{1}$$

where $z_y(x)$ denotes the softmax probability assigned to class y for input x. When model probabilities are well-calibrated, high-probability labels yield small scores, producing compact sets at fixed coverage. However, it lacks adaptivity to class imbalance or distributional uncertainty.

Adaptive Prediction Sets (APS). To improve adaptiveness over fixed-threshold methods like LAC, APS [6] constructs prediction sets by accumulating probability mass over the most likely classes. The nonconformity score for a label y is defined as:

$$\mathcal{S}_{\text{APS}}(x, y) = \sum_{j:\, \text{rank}_x(j) < \text{rank}_x(y)} z_j(x) + U \cdot z_y(x), \tag{2}$$

where $z_j(x)$ is the softmax probability of class j, and $U \sim \text{Unif}[0, 1]$ introduces randomization to ensure exact finite-sample coverage in the presence of ties. For deterministic behavior, U can be fixed to 1. The score reflects the cumulative mass of all labels more confident than y, plus a fraction of y's own score. Thus, hard or ambiguous examples—where probability is spread across many labels—receive higher scores and larger prediction sets, improving adaptation to uncertainty and class imbalance.

Regularized Adaptive Prediction Sets (RAPS). While APS improves class-wise coverage over LAC by adapting to uncertainty, it can lead to overly large prediction

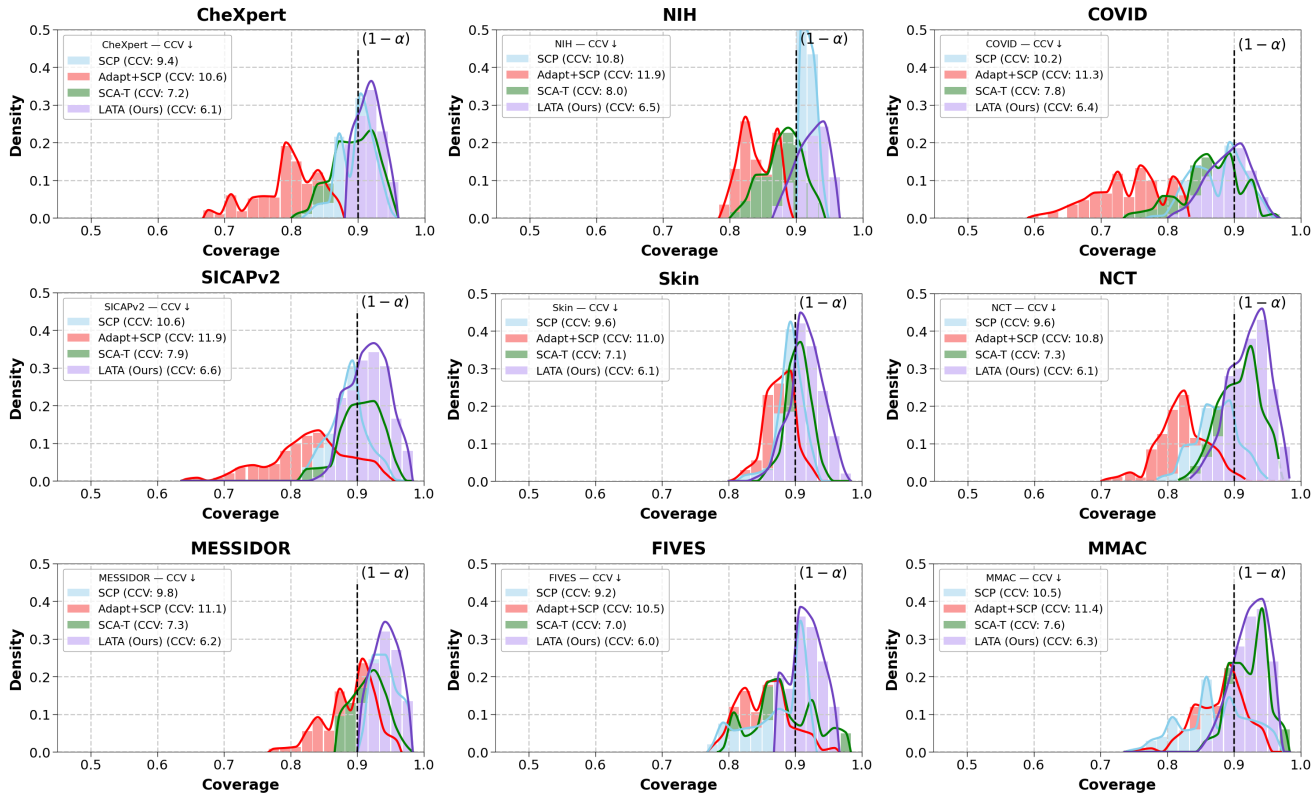

Figure S4. **Coverage analysis** under LAC at $\alpha = 0.10$. The dashed line indicates the nominal coverage level. **`LATA (ours)`** consistently achieves coverage closest to the target across all datasets.

sets—particularly in ambiguous examples. RAPS [1] addresses this by introducing a soft penalty on including lower-ranked classes beyond a fixed cutoff. The nonconformity score is defined as:

$$\begin{aligned}\mathcal{S}_{\text{RAPS}}(x, y) = &\sum_{j:\,\text{rank}_x(j)<\text{rank}_x(y)} z_j(x) \\ &+ \gamma_{\text{raps}} \cdot \max\{0,\, \text{rank}_x(y) - k_{\text{reg}}\} \\ &+ U \cdot z_y(x),\end{aligned} \tag{3}$$

where γ_{raps} controls the regularization strength, k_{reg} specifies the rank threshold after which penalties are applied, and $U \sim \text{Unif}[0, 1]$ enables randomized tie-breaking as in APS. The term $\max\{0,\, \text{rank}_x(y) - k_{\text{reg}}\}$ penalizes including labels with low confidence (i.e., high rank), effectively taming the tail of the score distribution. This regularization helps balance the trade-off between prediction set size and adaptivity across subgroups.

Coverage and set size. To evaluate the reliability and efficiency of conformal prediction, we compute two standard metrics on a labeled test set $\{(x_i, y_i)\}_{i=1}^{n_{\text{test}}}$.

The *empirical coverage* measures the proportion of test instances for which the true label is contained in the prediction set:

$$\text{Cov} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbb{1}\{y_i \in \mathcal{C}(x_i)\}, \tag{4}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. A conformal method satisfies marginal validity if $\text{Cov} \approx 1 - \alpha$.

The *average set size*, also referred to as inefficiency, quantifies the expected number of labels returned per prediction:

$$\text{Size} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} |\mathcal{C}(x_i)|\,. \tag{5}$$

Smaller prediction sets imply higher efficiency, provided that coverage remains close to the target level. Together, these metrics characterize the trade-off between reliability and compactness of the output sets.

ACA (Balanced Accuracy). Balanced accuracy, also known as classwise or macro accuracy, evaluates how well the model performs across all classes irrespective of imbalance. Let $n_c = |\{i : y_i = c\}|$ be the number of test samples in class c. Then, ACA is defined as:

$$\text{ACA} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{n_c} \sum_{i:\, y_i = c} \mathbb{1}\left\{\arg\max_j z_j(x_i) = c\right\}, \tag{6}$$

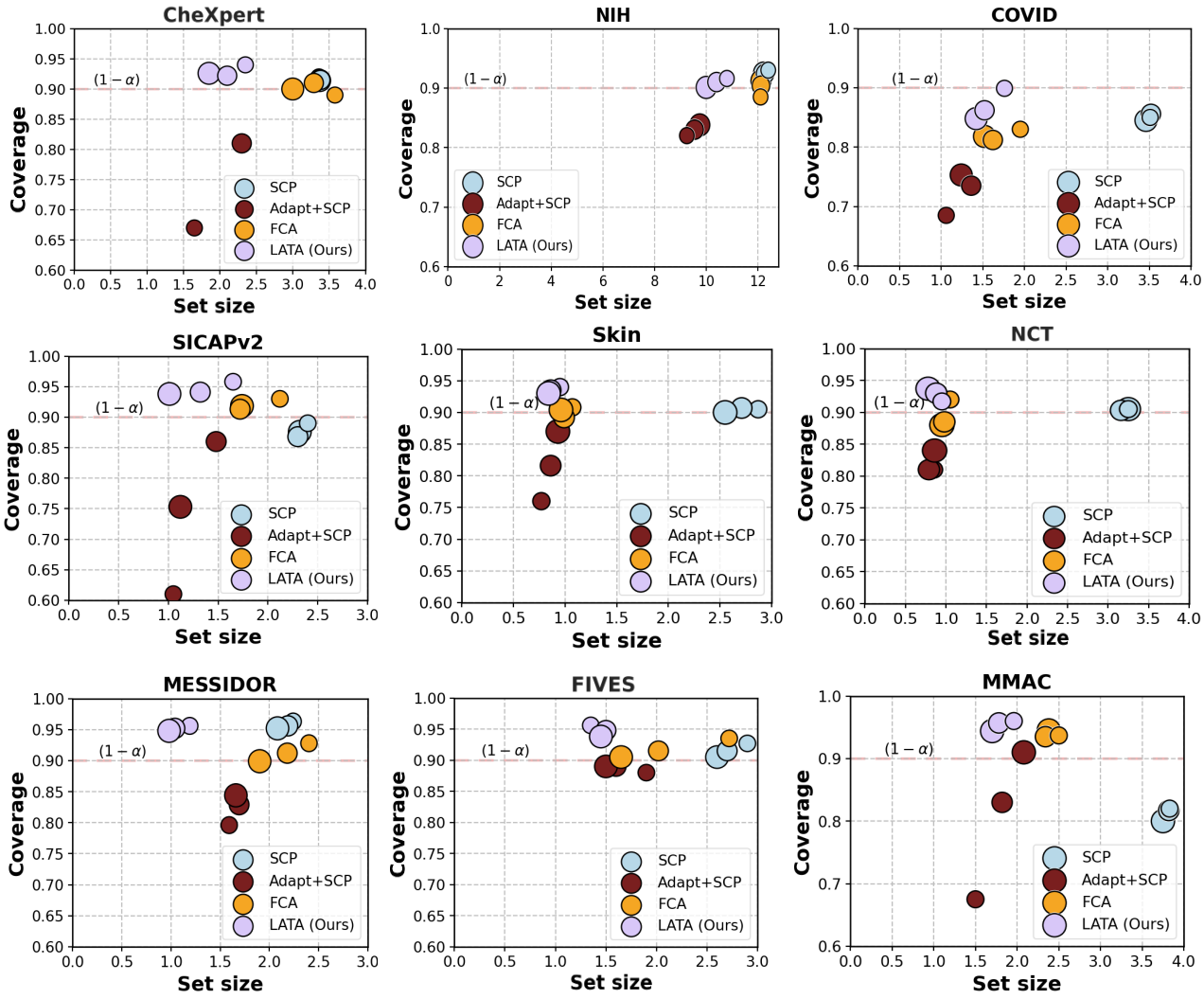

Figure S5. **Conformal prediction results across datasets** using $\alpha = 0.10$ and LAC [7]. Each point represents a performance, with bubble size indicating the number of adaptation shots $K \in \{4, 8, 16\}$.

where $z_j(x_i)$ is the predicted confidence for class j on sample x_i. This metric treats all classes equally, making it well-suited for imbalanced settings.

CCV (Class-Conditioned Coverage Gap). To assess how well conformal prediction methods balance coverage across different classes, we compute the classwise empirical coverage:

$$\widehat{\mathrm{Cov}}_c = \frac{1}{n_c} \sum_{i:\, y_i = c} \mathbb{1}\{y_i \in \mathcal{C}(x_i)\}, \tag{7}$$

and define the class-conditioned coverage gap (CCV) as:

$$\widehat{\mathrm{CCV}} = \frac{1}{C} \sum_{c=1}^{C} \left| \widehat{\mathrm{Cov}}_c - (1 - \alpha) \right|. \tag{8}$$

This metric captures the average deviation from the target coverage $(1 - \alpha)$ across all classes. A lower $\widehat{\mathrm{CCV}}$ indicates more uniform and fair coverage, with $\widehat{\mathrm{CCV}} = 0$ signifying perfect per-class alignment with the coverage goal.

In all experiments, the predicted class-probability vector is denoted by $z(x)$. Under **LATA**, we use its transductively refined version $\tilde{z}(x)$; otherwise we use the raw model output $q(x)$. The same deterministic transformation (e.g., **LATA** smoothing) is applied to both calibration and test samples, preserving exchangeability.

Dataset details. For all datasets and experiments, we adopt the dataset preparation protocol and train/calibration/test splits provided in the FCA codebase [11].

References

[1] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. arXiv preprint arXiv:2009.14193, 2020. 1, 4, 7

[2] Alex Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. arXiv preprint arXiv:1301.7375, 2013. 1

[3] Yunshi Huang, Fereshteh Shakeri, Jose Dolz, Malik Boudiaf, Houda Bahig, and Ismail Ben Ayed. Lp++: A surprisingly strong linear probe for few-shot clip. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 23773–23782, 2024. 1

[4] Marc Lafon, Yannis Karmim, Julio Silva-Rodríguez, Paul Couairon, Clément Rambour, Raphaël Fournier S'niehotta, Ismail Ben Ayed, Jose Dolz, and Nicolas Thome. ViLU: Learning Vision-Language Uncertainties for Failure Prediction. International Conference on Computer Vision (ICCV 2025), 2025. 4

[5] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In European conference on machine learning, pages 345–356. Springer, 2002. 1

[6] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. Advances in neural information processing systems, 33:3581–3591, 2020. 1, 6

[7] Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. Journal of the American Statistical Association, 114(525): 223–234, 2019. 1, 6, 8

[8] Julio Silva-Rodriguez, 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, pages 23681–23690, 2024. 1

[9] Julio Silva-Rodríguez, Ismail Ben Ayed, and Jose Dolz. Conformal prediction for zero-shot models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 19931–19941, 2025. 1, 2

[10] Julio Silva-Rodríguez, Ismail Ben Ayed, and Jose Dolz. Trustworthy few-shot transfer of medical vlms through split conformal prediction. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 658–668. Springer, 2025. 2

[11] Julio Silva-Rodríguez, Leo Fillioux, Paul-Henry Cournède, Maria Vakalopoulou, Stergios Christodoulidis, Ismail Ben Ayed, and Jose Dolz. Full conformal adaptation of medical vision-language models. In International Conference on Information Processing in Medical Imaging, pages 278–293. Springer, 2025. 2, 5, 8

[12] Vladimir Vovk. Conditional validity of inductive conformal predictors. In Asian conference on machine learning, pages 475–490. PMLR, 2012. 1

[13] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic learning in a random world. Springer, 2005. 1

[14] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6757–6767, 2023. 1

[15] Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). Advances in neural information processing systems, 14, 2001. 2

[16] Maxime Zanella, Benoît Gérin, and Ismail Ayed. Boosting vision-language models with transduction. Advances in Neural Information Processing Systems, 37:62223–62256, 2024. 1

[17] 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, 2022. 1

[18] Hao Zhu and Piotr Koniusz. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9078–9088, 2022. 2