Learning to Select Visual In-Context Demonstrations

Supplementary Material

A. Prompt Construction and In-Context Learning

We utilize a multimodal few-shot prompting strategy to query the Vision-Language Model (Gemma-3-4b-it). The prompt is constructed as an interleaved sequence of images and text, following the chat template structure required by the instruction-tuned model.

A.1. Prompt Structure

For a given query image x_q and a selected set of K demonstration examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_K, y_K)\}$, where x_i is an image and y_i is its ground truth label (e.g., age, count, or score), the conversation history is constructed as follows:

$$\mathcal{P} = [I(x_1), T(y_1), I(x_2), T(y_2), \ldots, I(x_K), T(y_K), I(x_q), Q_{task}] \tag{9}$$

where:

- $I(x)$ represents the image token inputs processed by the vision encoder (SigLIP).
- $T(y)$ is the text string representing the label of the demonstration image (e.g., "Age: 25").
- Q_{task} is the task-specific textual instruction that prompts the model to predict the label for the final image x_q.

We enforce strict output formatting by appending constraints to the system instruction and limiting generation to 20 tokens.

A.2. Task-Specific Instructions

The specific text prompts (Q_{task}) used for each dataset are detailed in Table 4.

A.3. Label Formatting

For the demonstration examples, the ground truth values are formatted as simple text strings to accompany the images:

- **Age Prediction:** "Age: $< y_i >$"
- **Scoring Tasks:** "Score: $< y_i >$"

This consistency allows the VLM to recognize the mapping pattern effectively.

B. Efficiency of Large-Scale Action Selection

A critical challenge in applying Reinforcement Learning to retrieval tasks is the magnitude of the action space. In our setting, the agent must select from a dataset of $N \approx 50,000$ candidate images. Standard discrete RL algorithms (e.g., DQN, PPO) face prohibitive convergence and computational hurdles at this scale. We adopt a method inspired by the Wolpertinger architecture [9], which maps states to continuous embedding coordinates rather than discrete indices. Below, we analyze the three primary advantages of this approach over discrete action spaces.

B.1. Overcoming the Exploration Cliff

In a standard discrete setting with N actions, the policy π must explore a multinomial distribution over N independent logits.

- **Discrete RL:** With $N = 50,000$, the probability of selecting a specific optimal demonstration via random exploration (e.g., ϵ-greedy) is $P(a^*) \approx 2 \times 10^{-5}$. This results in a vanishing gradient problem where the agent effectively never encounters a positive reward signal, leading to failure in convergence.
- **Proposed Method:** Our agent outputs a continuous vector $\hat{a} \in \mathbb{R}^D$. Exploration occurs in the semantic space. Even an imperfect vector output will retrieve neighbors in the embedding space that likely share task-relevant features with the optimal target, providing a denser reward signal and facilitating curriculum learning.

B.2. Bridging the Semantic Gap

Discrete RL treats actions as categorical indices without inherent relationships.

- **Discrete RL:** To a standard DQN, index i and index j are orthogonal, even if the underlying images are semantically identical (e.g., two similar images of a "Golden Retriever"). Learning that index i yields high reward provides zero information about index j. The agent must independently explore and learn values for all $50,000$ indices.
- **Proposed Method:** By operating in the embedding space, we exploit the inductive bias of the pre-trained encoder (SigLIP). If the agent learns to navigate towards a specific region in $\mathbb{R}^D$ (e.g., "dog-like images"), it simultaneously increases the selection probability for all semantically related candidates. This generalization capability drastically reduces the sample complexity required for training.

B.3. Computational Complexity

The computational cost of policy evaluation differs significantly between the methods due to the mechanism of action selection.

- **Discrete RL:** Standard policy gradient methods require a Softmax normalization over the entire action space to

Table 4. List of task-specific instructions used to prompt the VLM. The model is provided with K interleaved image-label pairs prior to these instructions.

Task / Dataset	Query Instruction (Q_{task})
Age Prediction *(UTKFace)*	"What is the age of the person in the last image? Only output the estimated age as a number."
Aesthetic Assessment *(AVA)*	"What is the aesthetic score of the last image on a scale from 0 to 10? Only output the score as a floating number."
Facial Beauty *(FBP5500)*	"What is the facial beauty score of the last image on a scale from 0 to 5? Only output the score as a floating number."
Image Quality *(KADID-10k, KonIQ-10k)*	"What is the image quality score of the last image on a scale from 0 to 5? Only output the score as a floating number."

compute probabilities:

$$P(a_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{10}$$

This incurs a computational complexity of $O(N)$ per step. As N grows (e.g., $N = 50,000$), calculating gradients for every output node becomes memory-intensive and prohibits efficient scaling.

- **Proposed Method (Two-Stage Selection):** Our approach decouples action generation from selection, reducing complexity from linear to logarithmic.
  1. **Proto-Action Generation:** The policy outputs a continuous vector $\hat{a} \in \mathbb{R}^D$.
  2. **Candidate Retrieval:** We utilize Approximate Nearest Neighbor search (FAISS) to retrieve a small set of candidates $\mathcal{C}_k$ (where $k \ll N$, e.g., $k = 200$) closest to $\hat{a}$. This search scales logarithmically $O(\log N)$.
  3. **Final Selection:** The policy evaluates Q-values only for the actions in $\mathcal{C}_k$.

  The total complexity is $O(\log N + k)$, which allows the method to scale to millions of images while keeping the evaluation cost constant.

C. Additional Implementation Details

C.1. Network Architecture & Hyperparameters

Our Dueling DQN agent utilizes a custom **Query-Centric Transformer Decoder** with specific architectural choices designed for stability and sample efficiency. The exact configuration is detailed below:

- **Transformer Configuration:**
  - **Layers (L):** 2
  - **Attention Heads (H):** 4
  - **Embedding Dimension (D):** 768 (matching the SigLIP vision encoder)
  - **Feedforward Dimension:** 3072 ($4 \times D$)
  - **Positional Encoding:** Learnable embeddings are added to the demonstration memory sequence to encode slot order.
  - **Normalization:** LayerNorm is applied *before* the attention/FFN blocks (`norm_first=True`) for improved training stability.
  - **Activation:** GELU
  - **Dropout:** 0.1
- **Dueling Heads:** Unlike standard architectures that use heavy MLPs, we found that single linear projections were sufficient given the rich representations from the Transformer context.
  - **Value Head:** A single linear layer mapping $D \to 1$.
  - **Advantage Head:** A single linear layer mapping $D \to D$, followed by L_2 normalization.

Table 5. Comparison between Standard Discrete RL and the proposed Continuous Embedding (Wolpertinger) approach for large-scale selection.

Feature	Discrete RL (e.g., PPO)	Proposed Method
Output Space	N logits (One per item)	Vector $\in \mathbb{R}^D$
Selection	Softmax over N	ANN Retrieval (k) $\to$ Argmax
Semantics	Orthogonal Actions	Semantic Neighbors
Complexity	Linear $O(N)$	Logarithmic $O(\log N + k)$
Scalability	Limited ($N < 10^4$)	Unlimited (via FAISS)

C.2. Optimization Process

We train the agent using the **Adam** optimizer with a learning rate of 5×10^{-6}. We employ gradient clipping with a max norm of 1.0 to ensure stability throughout the training process. The target network is updated using a soft update parameter $\tau = 0.005$.

We utilize a Replay Buffer with a capacity of 50,000 transitions and sample mini-batches of size 32. To encourage exploration, we use an ϵ-greedy schedule starting at $\epsilon = 0.9$ and decaying exponentially to $\epsilon = 0.05$ over the first 100,000 steps.

C.3. Data Splits and Evaluation Protocol

To ensure rigorous evaluation and prevent data leakage, we perform a strict separation between the data used for demonstration retrieval and the data used for evaluation queries.

C.3.1. Dataset Partitioning

For each task (e.g., Age Estimation, Aesthetic Scoring), we first load the raw dataset and filter for valid images. To maintain a manageable memory footprint for the FAISS index during experimentation, we cap the maximum dataset size at $N_{max} = 25,000$ samples. If a dataset exceeds this limit, we perform a random downsampling.

We partition this data into two disjoint sets using an 80/20 random split:

- **Demonstration Pool ($\mathcal{D}_{train}$):** Comprising 80% of the data, this set serves as the candidate pool. All K-shot demonstrations retrieved by our agent (or baselines) are strictly drawn from this pool. The FAISS index is built solely on the SigLIP embeddings of this set.
- **Query Set ($\mathcal{D}_{test}$):** Comprising the remaining 20%, this set is used exclusively to provide query images x_q. These images are never used as demonstrations.

C.3.2. Evaluation Sampling

While the Query Set ($\mathcal{D}_{test}$) may contain up to 5,000 images (20% of 25,000), performing full inference on large Vision-Language Models (e.g., Gemma-3-4B-IT, InternVL2) for every sample is computationally prohibitive due to the latency of auto-regressive generation.

To balance statistical significance with computational efficiency, we randomly sample a fixed subset of $N_{eval} = 1,000$ queries from $\mathcal{D}_{test}$ for the final quantitative evaluation. This sample size is sufficient to capture performance trends and calculate metrics (MAE, Accuracy) with low variance while keeping the evaluation time feasible.

C.4. FAISS Index Configuration and Retrieval Strategy

To efficiently handle the action space of $N \approx 50,000$ images, we employ the Facebook AI Similarity Search (FAISS) library. We construct an Inverted File with Product Quantization ('IndexIVFPQ') index to balance memory usage with retrieval speed. The configuration matches the embedding dimension of the SigLIP encoder ($D = 768$).

- **Metric:** Inner Product. We use `METRIC_INNER_PRODUCT`. Since all embeddings are L_2 normalized, this is equivalent to Cosine Similarity.
- **Coarse Quantizer ('nlist'):** 100 Voronoi cells. We use a flat inner product quantizer ('IndexFlatIP') for the coarse level.
- **Sub-Quantizers ('M'):** 8. The vectors are split into 8 sub-vectors.
- **Encoding Bits:** 8 bits per sub-vector.
- **Search Depth ('nprobe'):** 10. During inference and training, we visit the nearest 10 Voronoi cells.
- **Candidate Pool Size (k):** 200. During the training step, we retrieve the top $k = 200$ candidates for the generated proto-action. This pool size is critical for the Dueling DQN architecture, as it serves as the sample set to approximate the mean advantage value:

$$\bar{A}(s, \cdot) \approx \frac{1}{k} \sum_{a_j \in \text{top-}k} A(s, a_j) \tag{11}$$

C.5. Environment & Reward Shaping

To address the cold-start problem, the environment employs an **anchor initialization** strategy: the initial state s_0 always includes the query image and its nearest neighbor (retrieved via FAISS) as the first demonstration.

We utilize a **differential reward function** to encourage marginal improvement. The immediate reward r_t is calculated as the change in the MLLM's performance score:

$$r_t = \frac{1}{\lambda}(S_t - S_{t-1}) \tag{12}$$

where $\lambda = 10.0$ is a global scaling constant for the replay buffer.

The performance score S_t is task-dependent, designed to normalize the error magnitude across different output ranges (e.g., $0 - 100$ for age vs. $0 - 5$ for quality). Let $\delta = |y_{pred} - y_{gt}|$ be the absolute error. S_t is defined as:

$$S_t = \begin{cases} -\delta & \text{Age Prediction} \\ -10 \cdot \delta & \text{Aesthetic Scoring (0-10)} \\ -20 \cdot \delta & \text{Quality/Beauty Scoring (0-5)} \end{cases} \tag{13}$$

For crowd counting, we use a relative error formulation (damped by a floor of 10 heads) to handle the large variance in crowd sizes. For scoring tasks, we apply multipliers (10 or 20) to amplify the gradient signal for small floating-point errors. If the agent selects an invalid action, the episode terminates with a fixed penalty of -0.5.

D. Extended Demonstration Set Analysis

To ensure the universality of our learned policy, we extended the Demonstration Set Analysis presented in the main paper to cover all five benchmark datasets. This experiment was conducted in the Intra-Model setting (Train Gemma 3 4B-it / Eval Gemma 3 4B-it). The results, summarized visually in Fig. 6 and Fig. 7, consistently confirm the emergence of a sophisticated, multi-objective policy across all tasks.

E. Comprehensive Cross-Model Generalization

In the main paper, we demonstrated the transfer capability of a single policy. Here, we present a comprehensive, all-to-all transfer analysis to rigorously test the universality of our approach.

We utilize three state-of-the-art MLLMs for training source policies: **Gemma 3 4B-it**, **Qwen 2.5 7B**, and **InternVL2-8B**. We evaluate these policies against all four target MLLMs, including **Phi-3.5-vision**. We define a transfer matrix experiment where we train a distinct LSD agent on each source model and evaluate it on every target model.

E.1. The Transfer Matrix

The aggregate results on the UTKFace dataset ($K = 4$) are presented in Tab. 6.

E.2. Detailed Transfer Performance Plots

To visualize the robustness of these policies as the number of demonstrations (K) increases, we plot the performance curves for every combination of Source and Target models in Figures 8, 9, and 10.

E.3. Analysis of Transfer Patterns

This comprehensive analysis yields three critical insights into the transferability of learned retrieval policies:

- **Superiority over Random Baselines:** Across all transfer scenarios (both Intra and Cross), the LSD policy consistently and significantly outperforms random selection (green lines in Figures). This confirms that the agent learns a fundamental, valid retrieval heuristic—likely selecting diverse anchors—that is universally more effective than chance, regardless of the target model.
- **The Specialization Gap (LSD vs. kNN):**
  - **Intra-Model (Diagonal):** When the source and target models match (e.g., Gemma → Gemma), LSD consistently outperforms the strong kNN baseline. This indicates the agent learns to exploit model-specific sensitivities to specific examples or ordering.
  - **Cross-Model (Off-Diagonal):** When transferring to a new model (e.g., Gemma → Qwen), the performance gap narrows. LSD performs comparably to kNN, sometimes slightly better or worse depending on the specific pairing. This suggests that while the "diversity" heuristic is universal, the fine-grained "optimality" of a specific set is highly coupled to the inference dynamics of the specific MLLM used during training.
- **Model-Specific Sensitivity:** We observe that **Phi-3.5-vision** appears particularly resistant to policy transfer, with cross-trained policies often converging to kNN-level performance but rarely exceeding it. This highlights that different MLLM architectures (e.g., Phi vs. Qwen) may rely on fundamentally different internal mechanisms for in-context learning, limiting the direct transferability of a specialized policy.

F. Cross-Dataset Generalization Analysis

In the main paper, we presented the performance of agents trained specifically for their target domains (**LSD-Self**). To evaluate the universality of the learned retrieval policy, we employ a **LSD-Cross** protocol: we take the agent trained solely on **UTKFace** (Age Prediction) and evaluate it directly on the remaining datasets without fine-tuning.

F.1. Results and Discussion

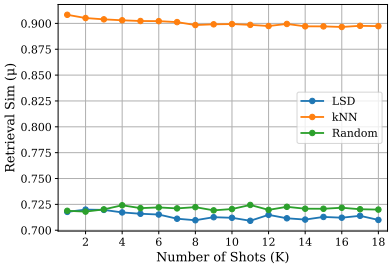
The results are visualized in Fig. 11. The transfer performance varies significantly by task nature, revealing distinct behaviors of the learned policy.

Robust Transfer on Objective Distortions (KADID-10k). As shown in Fig. 11d, the **LSD-Cross** agent (Red) demonstrates exceptional transfer capabilities on the KADID-10k dataset. Despite being trained on faces, the policy—which learns to select diverse anchors to span the regression range—is highly effective for Image Quality Assessment. It matches the performance of the domain-specific **LSD-Self** agent and significantly outperforms the kNN baseline. This confirms our hypothesis that for objective regression tasks, a "diversity-aware" selection strategy is universally beneficial and task-agnostic.

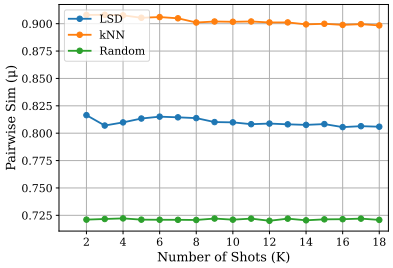
Negative Transfer on Facial Analysis (SCUT-FBP5500). In Fig. 11b, we observe a case of negative transfer. The Age-trained policy performs significantly worse than Random selection. We hypothesize this is due to conflicting objectives: the UTKFace agent is incentivized to retrieve a maximally diverse age range (e.g., toddlers and the elderly). However, for facial beauty scoring, extreme age diversity may introduce noise or out-of-distribution examples that confuse the MLLM's attractiveness estimation.

Generic Heuristics on Aesthetics (AVA). On the AVA dataset (Fig. 11a), the **LSD-Cross** (Red) and **LSD-Self** (Blue) lines are nearly indistinguishable. This implies that training specifically on AVA yielded the same generic retrieval strategy as training on Age. However, both fall short of the kNN baseline. This reinforces the finding that

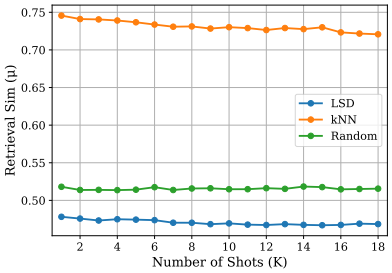
COLUMNS: (Left) Demo-Query Similarity; (Right) Pairwise Similarity



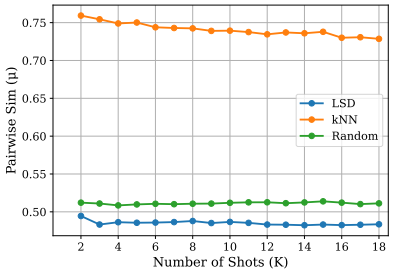
UTKFace: Demo-Query Feature Similarity



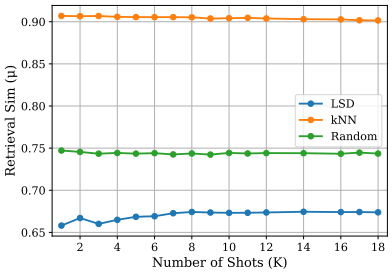
UTKFace: Pairwise Feature Similarity



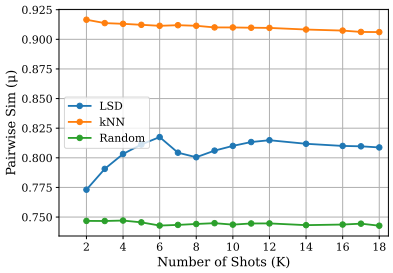
AVA: Demo-Query Feature Similarity



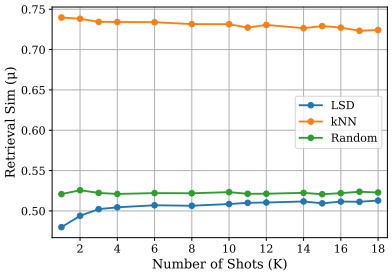
AVA: Pairwise Feature Similarity



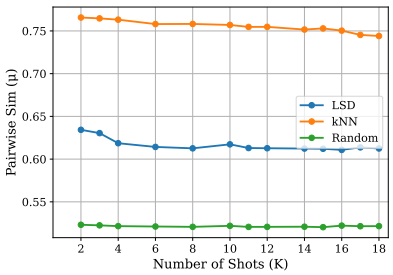
SCUT-FBP5500: Demo-Query Feature Similarity



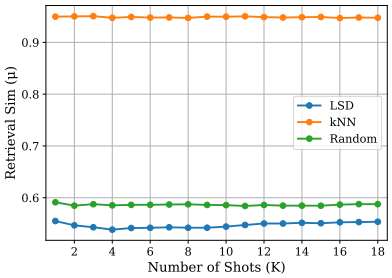
SCUT-FBP5500: Pairwise Feature Similarity



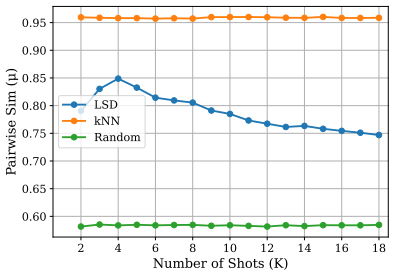
KonIQ-10k: Demo-Query Feature Similarity



KonIQ-10k: Pairwise Feature Similarity



KADID-10k: Demo-Query Feature Similarity



KADID-10k: Pairwise Feature Similarity

Figure 6. **Extended Feature-Space Analysis (Relevance and Similarity).** The plots in the right column demonstrate that on all five datasets, LSD (blue line) actively seeks low redundancy, maintaining the trend LSD $\ll$ kNN in pairwise similarity, which is the key behavioral difference.

(a) UTKFace: Label MAE vs. Query

(b) UTKFace: Pairwise Label MAE

(c) AVA: Label MAE vs. Query

(d) AVA: Pairwise Label MAE

(e) SCUT-FBP5500: Label MAE vs. Query

(f) SCUT-FBP5500: Pairwise Label MAE

(g) KonIQ-10k: Label MAE vs. Query

(h) KonIQ-10k: Pairwise Label MAE

(i) KADID-10k: Label MAE vs. Query

(j) KADID-10k: Pairwise Label MAE

Figure 7. **Extended Label-Space Analysis (Emergent Relevance and Consistency).** The results show a critical, task-dependent pattern in minimizing the label difference (ΔLabel) between the query and selected demos. For **Objective Tasks** (UTKFace, KonIQ-10k, KADID-10k), the **LSD policy (blue)** is the most effective implicit label retriever. Conversely, for **Subjective Tasks** (AVA, SCUT-FBP5500), the **kNN Baseline (orange)** consistently selects demos that minimize the label difference. This confirms that the optimal policy for minimizing label-space MAE aligns with the task's underlying human perception structure.

Table 6. **Cross-Model Transfer Matrix (MAE ↓) on UTKFace at $K = 4$.** Rows represent the *Source Policy* (training model). Columns represent the *Target MLLM* (evaluation model). **Diagonal entries (gray)** represent Intra-Model performance (Specialization), where LSD typically achieves its peak performance. **Off-diagonal entries** represent Inter-Model performance (Generalization). While LSD consistently outperforms Random selection (not shown), it performs comparably to the kNN baseline in cross-model settings, suggesting that model-specific nuances play a significant role in optimal retrieval.

Source Policy (Training)	Target MLLM (Evaluation) Gemma 3 4B	Qwen 2.5 7B	Phi-3.5-vision	InternVL2-8B
Baseline: kNN	*7.27*	*6.54*	*5.96*	*7.38*
LSD (Gemma)	6.27	**5.58**	6.05	10.62
LSD (Qwen)	6.81	**5.95**	6.17	9.69
LSD (InternVL)	6.01	5.77	**5.06**	8.87

for subjective, content-heavy tasks like aesthetics, semantic similarity (kNN) remains the dominant factor, and the "diversity" heuristic learned by LSD provides less benefit.

G. Extended Qualitative Analysis

In Fig. 12, we visualize the retrieval behavior of the proposed LSD agent versus the kNN baseline. The results highlight a critical limitation of standard retrieval in regression tasks: *semantic redundancy*.

Overcoming Semantic Redundancy (KADID-10k & AVA). The most distinct failure mode of kNN is visible in the KADID-10k example (Row 2). Because the dataset contains multiple distorted versions of the same reference images, kNN retrieves 11 versions of the *same beach scene*. This provides the VLM with no comparative information regarding quality standards. LSD, driven by the reward signal, learns to avoid this redundancy, selecting completely different scenes (sports, traffic, buildings) to illustrate the concept of "image quality" broadly. Similarly, in AVA (Row 4), kNN matches the red color of the query berries to flamingos, whereas LSD retrieves structurally diverse images (architecture, objects) that likely span the aesthetic scoring range.

Demographic and Age Diversity (SCUT-FBP5500 & UTKFace). For facial analysis tasks, kNN tends to over-index on demographic similarity.

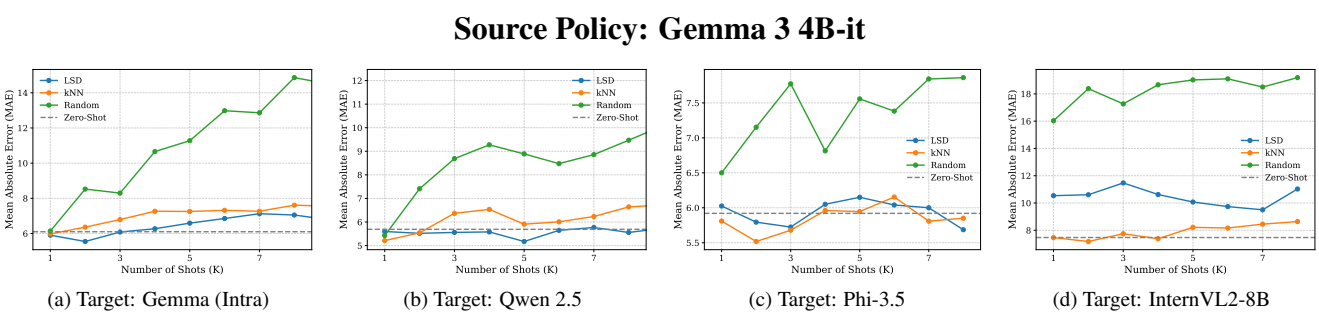
- On **UTKFace** (Row 1), the kNN baseline retrieves almost exclusively babies and toddlers for a child query. This prevents the VLM from accessing "anchor" examples of adults or the elderly, which are necessary to calibrate the upper bounds of age estimation. LSD retrieves a full age spectrum.
- On **SCUT-FBP5500** (Row 5), kNN restricts the context to the same gender and ethnicity (Asian males) as the query. LSD breaks this demographic lock, retrieving Caucasian and Asian faces of both genders, which encourages the VLM to abstract the concept of "facial beauty" away from specific demographic features.

Subjective Tasks (AVA, SCUT-FBP5500). For the subjective tasks, the behavioral difference remains distinct, even if the quantitative advantage is smaller.

- On **AVA** (Fig. 12b), kNN retrieves images with near-identical composition (e.g., 12 sunsets), effectively asking the model to "rate this sunset based on these other sunsets." LSD retrieves a diverse portfolio of photography styles (e.g., macro, portrait, landscape). While kNN performed better quantitatively on AVA, LSD's policy is demonstrably more informative about the *general concept* of aesthetics, rather than just specific object aesthetics.
- Similarly, on **SCUT-FBP5500** (Fig. 12c), kNN selects faces that look like "siblings" of the query. LSD selects a cohort that varies significantly in appearance and attractiveness rating, attempting to provide a broader comparative scale for the MLLM.

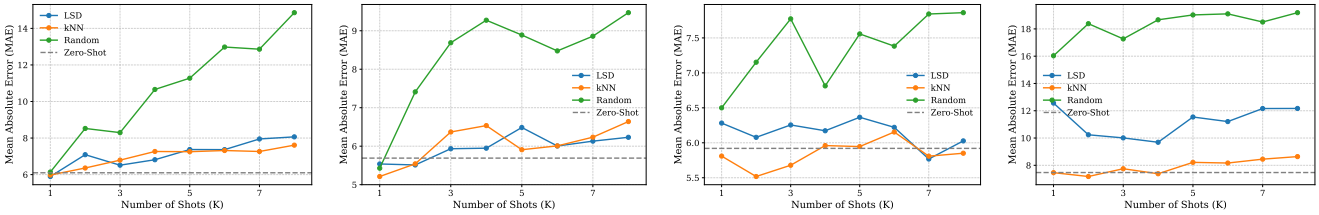
Table 7. **Cross-Dataset Generalization Analysis.** We report the MAE (↓) for the **kNN** baseline, the Domain-Specific Agent (**Self**), and the Cross-Trained Agent (**Cross**, trained on UTKFace) across $K \in \{1, 4, 8\}$. **LSD-Self** typically sets the upper bound. Comparing **LSD-Cross** to **kNN** reveals where the learned "diversity" heuristic transfers effectively (e.g., KADID-10k) versus where domain-specific visual matching is superior (e.g., AVA).

		K=1			K=4			K=8		
Dataset	**0-Shot**	**kNN**	**Self**	**Cross**	**kNN**	**Self**	**Cross**	**kNN**	**Self**	**Cross**
AVA	1.38	**1.58**	1.70	1.61	**1.22**	1.26	1.39	**0.98**	0.99	1.18
SCUT-FBP5500	1.07	**0.91**	1.13	1.01	**0.36**	0.53	1.03	**0.38**	0.52	0.98
KonIQ-10k	0.78	**0.73**	0.74	0.72	**0.45**	0.48	0.57	**0.37**	0.38	0.44
KADID-10k	1.13	1.16	**1.07**	1.12	1.06	**0.98**	0.94	1.05	**0.94**	1.02



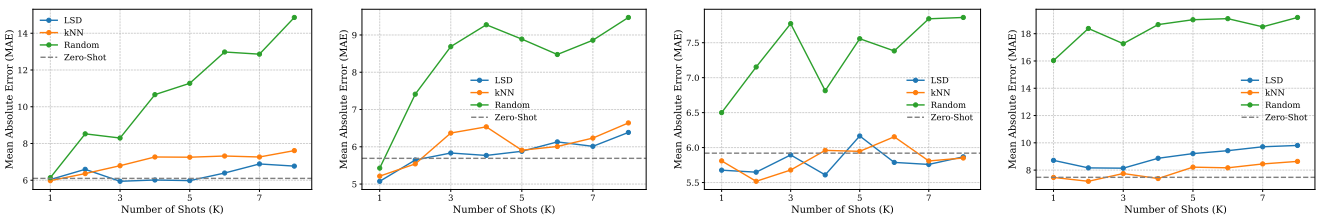
(a) Target: Gemma (Intra) (b) Target: Qwen 2.5 (c) Target: Phi-3.5 (d) Target: InternVL2-8B

Figure 8. **Transfer Scaling for Source Policy: Gemma 3 4B-it.** Performance of the Gemma-trained LSD policy evaluated across all four target models. The policy generalizes well, consistently beating kNN on Qwen and InternVL, and matching it on Phi.



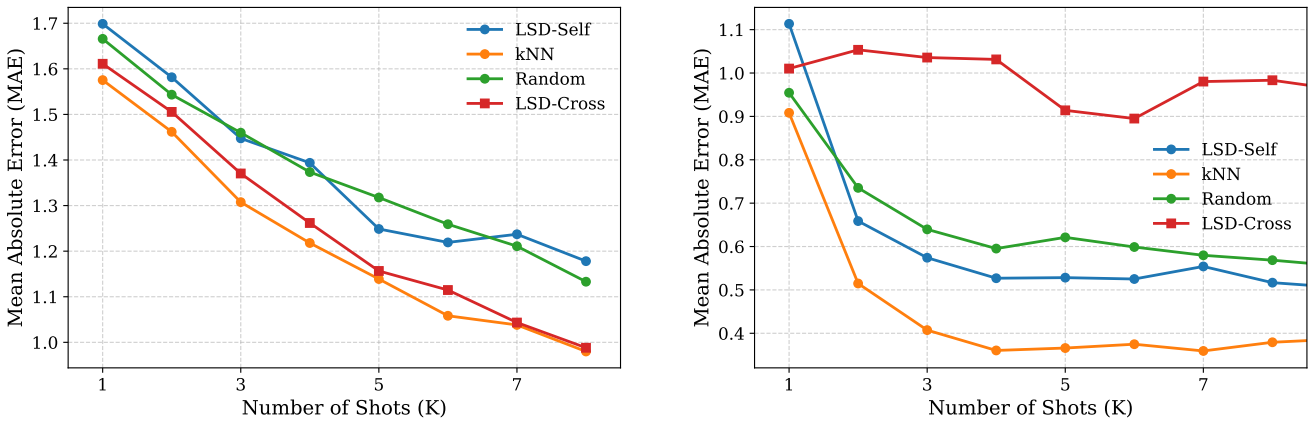
(a) Target: Gemma 3 (b) Target: Qwen (Intra) (c) Target: Phi-3.5 (d) Target: InternVL2-8B

Figure 9. **Transfer Scaling for Source Policy: Qwen 2.5 7B.** Performance of the Qwen-trained LSD policy evaluated across all targets.



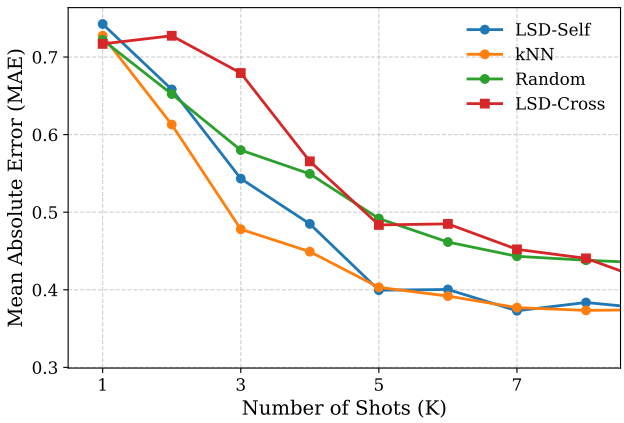
(a) Target: Gemma 3 (b) Target: Qwen 2.5 (c) Target: Phi-3.5 (d) Target: InternVL (Intra)

Figure 10. **Transfer Scaling for Source Policy: InternVL2-8B.** Performance of the InternVL-trained LSD policy evaluated across all targets.

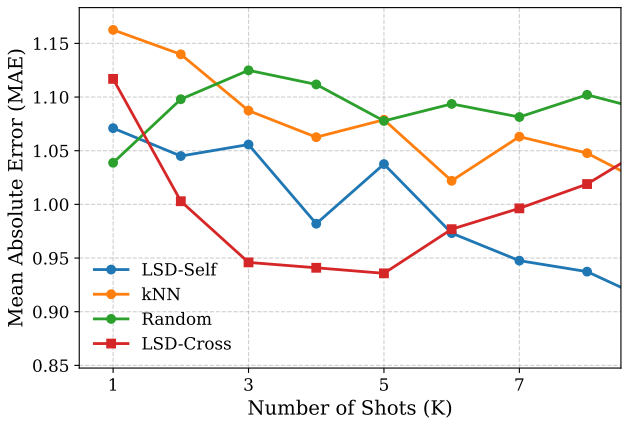


(a) AVA (Aesthetic Rating)

(b) SCUT-FBP5500 (Facial Beauty)



(c) KonIQ-10k (Wild Image Quality)



(d) KADID-10k (Distorted Image Quality)

Figure 11. **Cross-Dataset Generalization Results.** We compare **LSD-Self** (Blue, trained on target), **LSD-Cross** (Red, trained on Age), **kNN** (Orange), and **Random** (Green). **(d) Successful Transfer:** On KADID-10k, the Age policy (Red) transfers remarkably well, matching the Self-trained agent and beating kNN/Random. **(b) Negative Transfer:** On SCUT-FBP5500, the Age policy hurts performance, performing worse than random. **(a) Generic Policy:** On AVA, the Cross and Self policies perform identically, suggesting the learned retrieval strategy is task-agnostic but inferior to semantic matching (kNN) for aesthetics.

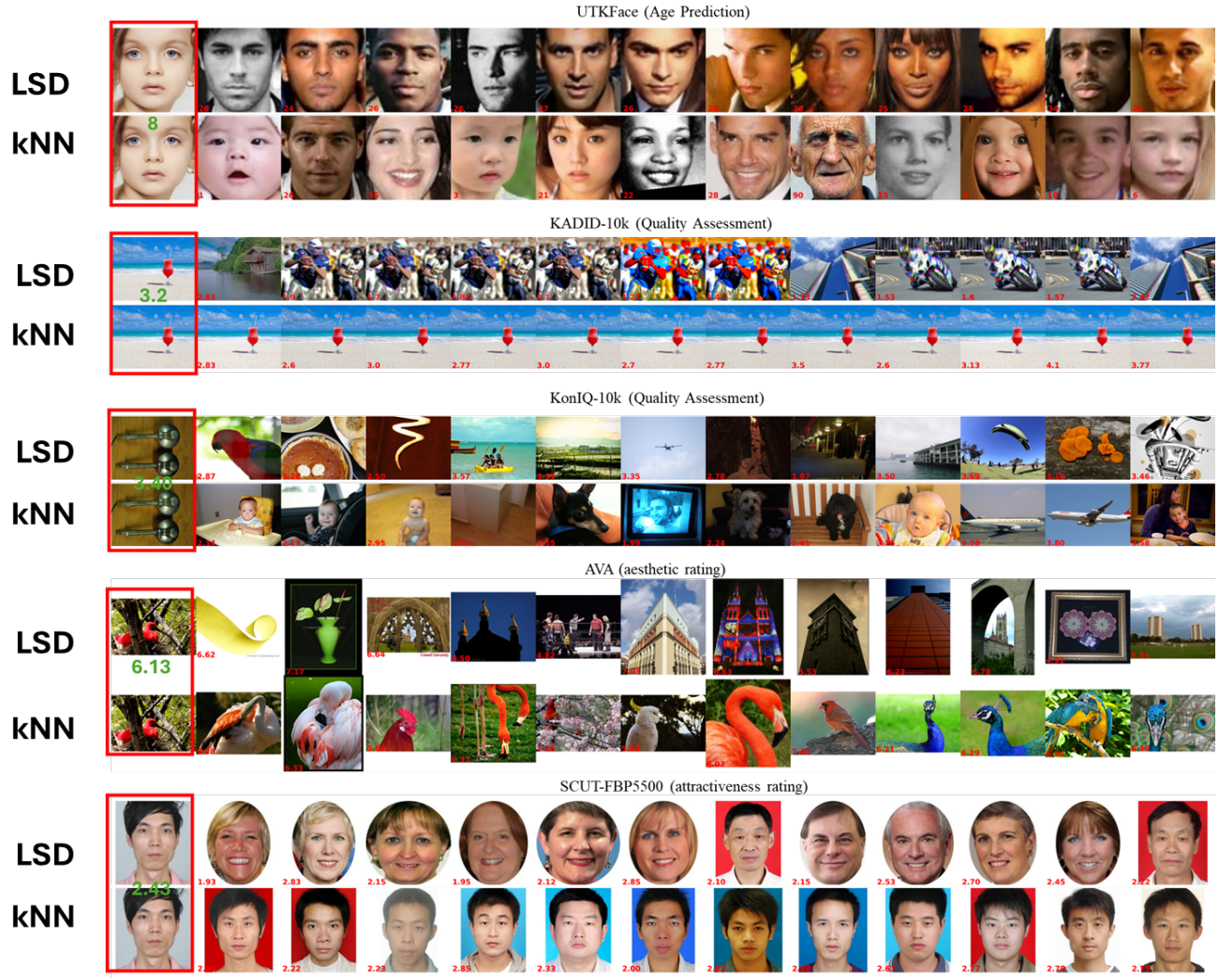


Figure 12. **Extended Qualitative Comparison of Selected Demonstrations ($K = 11$) across benchmark datasets. Row 1: UTKFace (Age).** For a query of a **young child**, the **kNN** baseline retrieves a homogeneous set of other children and babies. In contrast, **LSD (Ours)** retrieves a diverse timeline of faces, ranging from toddlers to adults and the elderly, providing the VLM with a complete regression scale. **Row 2: KADID-10k (Quality).** For a query of a beach scene, **kNN** fails by retrieving near-duplicate versions of the *same source image*, adding zero new information. **LSD** selects visually distinct scenes (sports, cityscapes) with varying distortion types. **Row 3: KonIQ-10k (Quality).** For a query of abstract metal spheres, **LSD** retrieves a broad semantic range (animals, food, landscapes), whereas kNN gets stuck in a narrow cluster of indoor/portrait shots. **Row 4: AVA (Aesthetics).** For a query of red berries, **kNN** relies on color and content matching, retrieving flamingos and other birds. **LSD** ignores the specific content, selecting architecture and objects to illustrate broad aesthetic principles. **Row 5: SCUT-FBP5500 (Beauty).** For a query of an Asian male, **kNN** exhibits high demographic bias, retrieving only other Asian males. **LSD** retrieves a diverse set of demographics (varying gender and race), reducing bias and helping the model score attractiveness independent of demographic features.