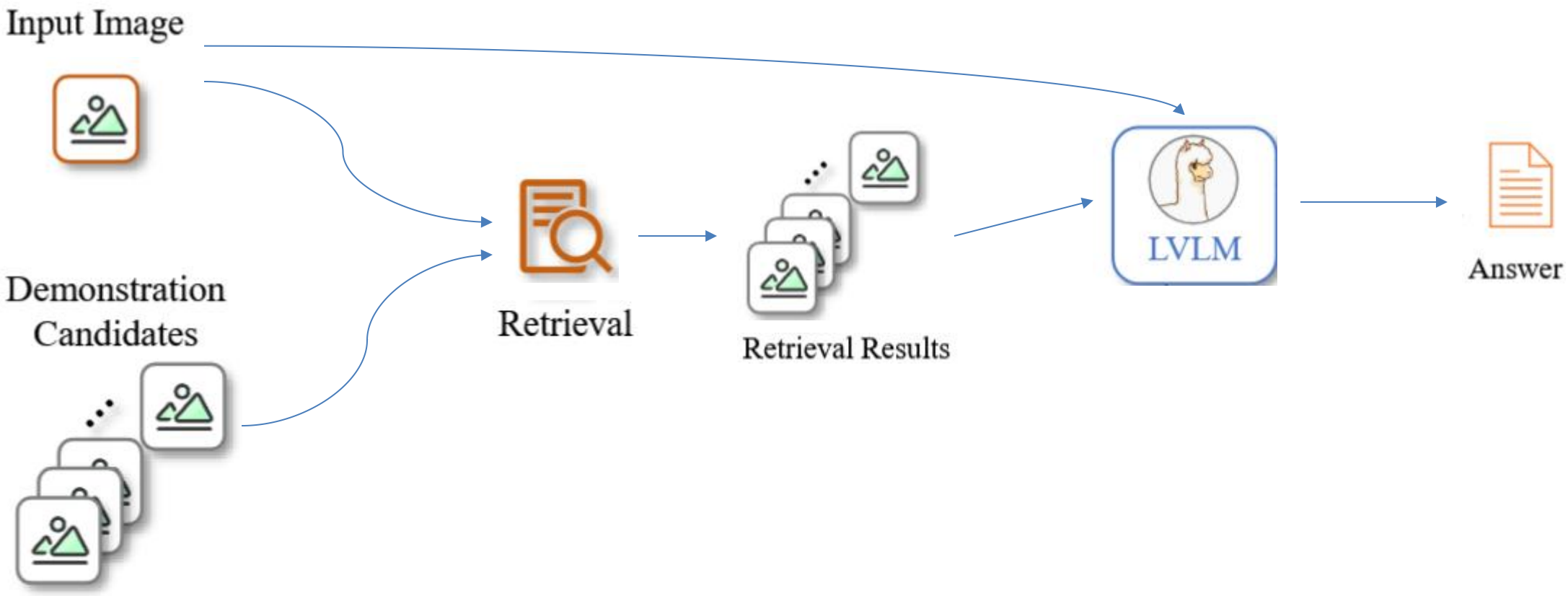
Learning to Select Visual In-Context Demonstrations

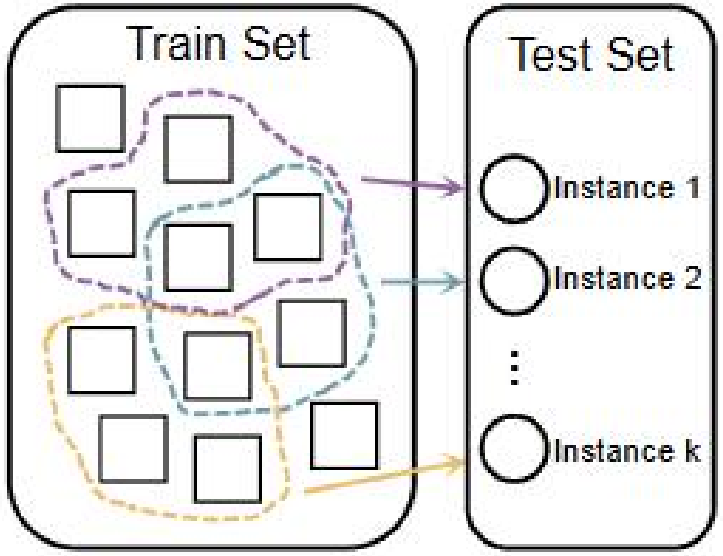
Eugene Lee[1], Yu-Chi Lin[2], Jiajie Diao[1]
[1] University of Cincinnati [2] University of California, Los Angeles
eugene.lee@uc.edu, yclin0177@g.ucla.edu, jiajie.diao@uc.edu

CVPR FINDINGS 2026

Visual In-Context Learning

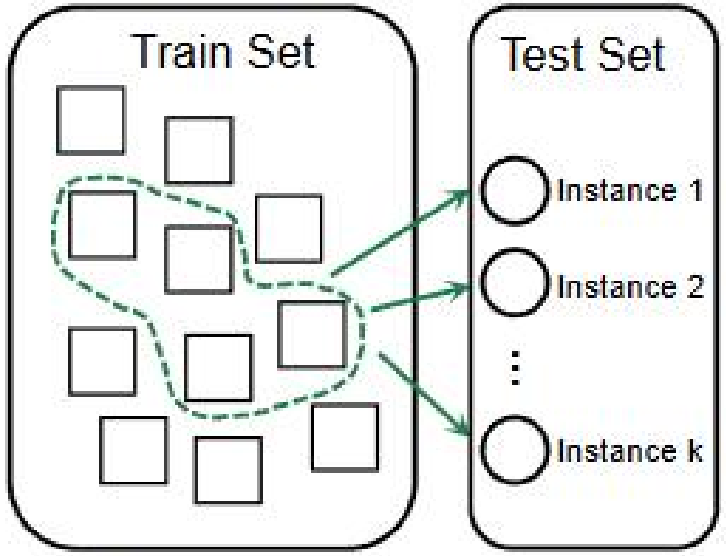


Demonstration Selection for In-Context Learning



Retrieval-based In-context Demonstration Selection

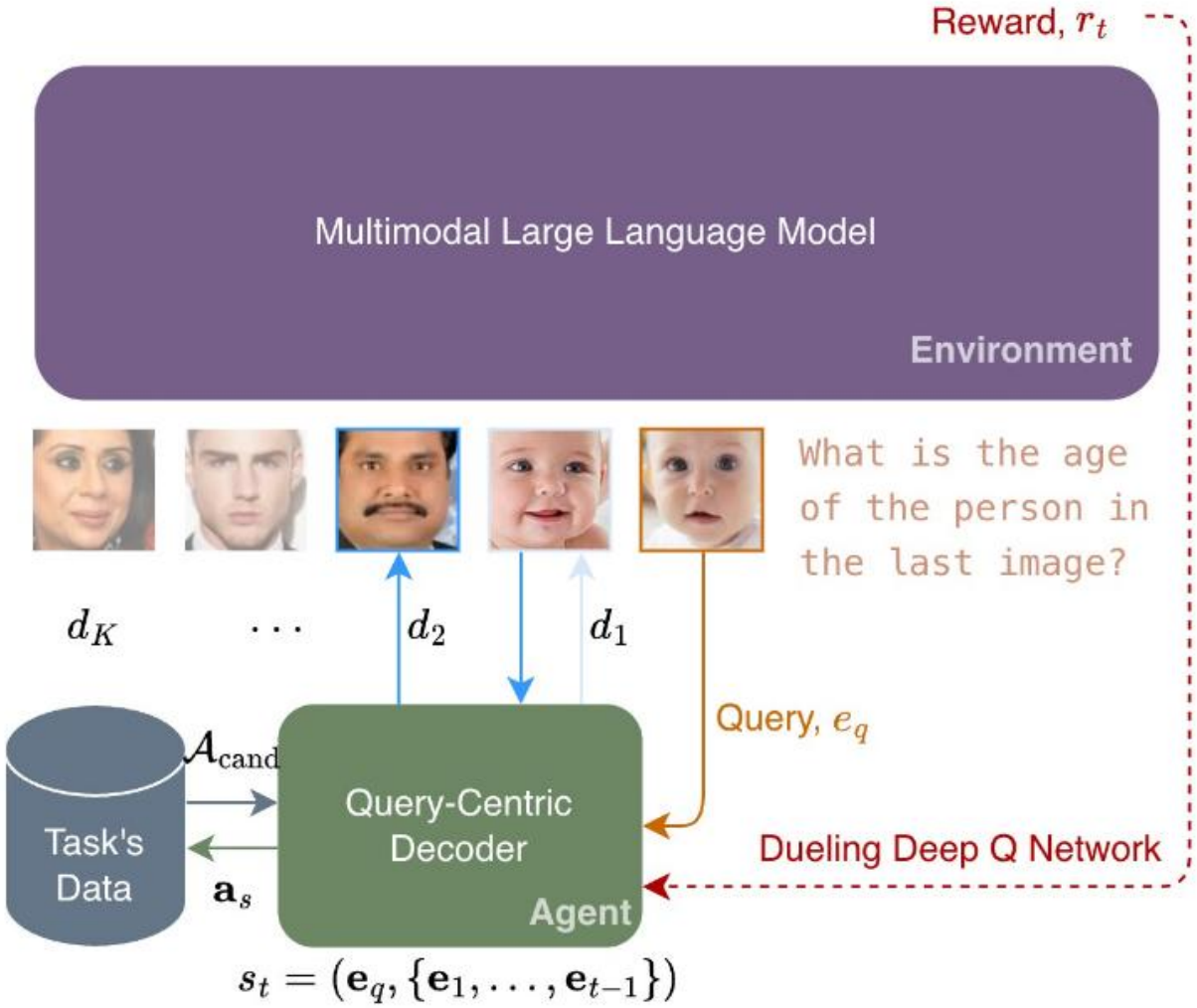
SupPR (2023 **NIPS**)
VICL (2024 **ACL**)



Representative In-context Demonstration Selection

DPP (2023 **EMNLP**)
CASE (2025 **ICML**)

Overview



adopt a Reinforcement Learning (RL) framework to learn a policy that iteratively constructs a high-quality demonstration set.

Problem Formulation as an MDP

model the K-shot demonstration selection process as a finite-horizon Markov Decision Process (MDP) , defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$.

State $(s_t \in \mathcal{S})$**:** defined by the query *q* and the ordered set of demonstrations D_{t-1} selected so far

Action $(a_t \in \mathcal{A})$**:** the selection of a new demonstration from the pool of all available examples

Transition $(\mathcal{P})$**:** Upon taking action a_t , the environment transitions from $s_t = (q, D_{t-1})$ to $s_{t+1} = (q, D_t)$

Reward $(\mathcal{R})$**:** $R(s_t) = -\text{MAE}(\mathcal{V}(q, D_{t-1}))$ $r(s_t, a_t) = R(s_{t+1}) - R(s_t)$

Discount (γ)**:** to balance immediate and future rewards

Dueling Q-Network for Large Action Spaces

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} A(s,a') \right)$$

$s \in \mathcal{S}$ → computes $V(s)$ and a *D*-dimensional "advantage query" vector $\mathbf{a}_s$

$a \in \mathcal{A}$ → N samples represented by a *D*-dimensional embedding $\mathbf{e}_i$

$$A(s, a_i) = \mathbf{a}_s^\top \mathbf{e}_i$$

Network Architecture

Memory： $\mathbf{M} = \mathbf{E}_D + \mathbf{P}$

Query-Centric State Encoder

$$\mathbf{x}_q^{(0)} = \mathbf{e}_q$$

$$\mathbf{x}'_q = \mathbf{x}_q^{(l-1)} + \text{Softmax}\left(\frac{(\mathbf{x}_q^{(l-1)} W_Q^{(l)})(\mathbf{M} W_K^{(l)})^\top}{\sqrt{d_k}}\right)(\mathbf{M} W_V^{(l)})$$

$$\mathbf{x}_q^{(l)} = \mathbf{x}'_q + \text{FFN}^{(l)}(\mathbf{x}'_q) \quad \forall l \in \{1, \ldots, L\}$$

$$\mathbf{c}_s = \mathbf{x}_q^{(L)}$$

Value Head $V(s) = \mathbf{w}_v^\top \mathbf{c}_s + b_v$

Advantage Head $\mathbf{a}_s = \dfrac{\mathbf{W}_a \mathbf{c}_s + \mathbf{b}_a}{\|\mathbf{W}_a \mathbf{c}_s + \mathbf{b}_a\|_2}$

Action Selection

ε-greedy policy

With probability ε, we explore by selecting a random valid action from the *N* candidates returned by FAISS. With probability 1 − ε, we exploit by executing the following steps:

1. Compute the state-value $V(s_t)$ and the advantage query $\mathbf{a}_{s_t}$ using the policy network Q_θ.
2. Use the FAISS index to retrieve the top *N* candidate actions: $\mathcal{A}_{\text{cand}} = \text{FAISS}(\mathbf{a}_{s_t}, \mathcal{N})$.
3. Calculate the advantage $A(s_t, a_j)$ for all $a_j \in \mathcal{A}_{\text{cand}}$.
4. Approximate the mean advantage using only these candidates: $\bar{A} \approx \frac{1}{N} \sum_{a_j \in \mathcal{A}_{\text{cand}}} A(s_t, a_j)$.
5. Select the best action at according to the dueling Q-value:

$$a_t = \text{argmax}_{a_j \in \mathcal{A}_{\text{cand}}} \left(V(s_t) + (A(s_t, a_j) - \bar{A}) \right)$$

$$A(s, a_i) = \mathbf{a}_s^\top \mathbf{e}_i$$

Optimization

We store transitions $(s_t, a_t, r_t, s_{t+1}, \text{done})$ in a replay buffer. For a mini-batch of *B* transitions, we compute the target y_t using the target network Q_{θ^-}:

$$y_t = r_t + \gamma(1 - \text{done}) \cdot \max_{a' \in \mathcal{A}'_{\text{cand}}} Q(s_{t+1}, a'; \theta^-)$$

Policy network Q_θ is then updated by minimizing the Smooth L1 (Huber) Loss between the predicted $Q(s_t, a_t; \theta)$ and the target y_t:

$$L(\theta) = \frac{1}{B} \sum_{(s,a,r,s') \in \mathcal{B}} \mathcal{L}_{\text{Huber}} \left(y_t - Q(s_t, a_t; \theta) \right)$$

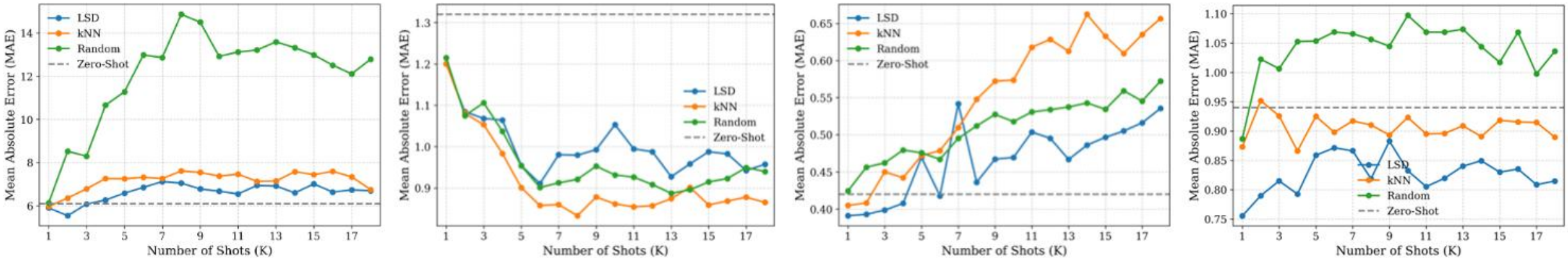
The target network weights θ^- are updated via a soft polyak average: $\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^-$

Main Performance vs. K

Table 1. **Main Performance (MAE ↓) Comparison vs. Number of Shots (K).** We report the Mean Absolute Error (MAE) for all methods on the five benchmark datasets, evaluated with Gemma 3 4B-it for $K \in \{1, 4, 8, 16\}$. Our proposed method, **LSD**, consistently outperforms all baselines, and the performance gap widens as K increases. The 0-shot and a fully **Supervised** (Sup.) baseline are also provided for reference. Best results are in **bold**.

			Random				kNN				LSD (Ours)			
Dataset	**Sup.**	**0-Shot**	K=1	K=4	K=8	K=16	K=1	K=4	K=8	K=16	K=1	K=4	K=8	K=16
UTKFace	4.42 [27]	6.10	6.14	10.66	14.86	12.51	5.98	7.27	7.61	7.60	**5.90**	**6.27**	**7.05**	**6.64**
AVA	-	1.32	1.21	1.03	0.92	0.92	**1.20**	**0.98**	**0.83**	**0.86**	**1.20**	1.06	0.98	0.98
SCUT-FBP5500	0.26 [45]	0.59	0.58	0.64	0.64	0.68	**0.53**	**0.39**	**0.40**	**0.44**	0.55	0.62	0.67	0.75
KonIQ-10k	0.39 [34]	0.42	0.42	0.48	**0.44**	0.56	0.40	0.44	0.55	0.61	**0.39**	**0.40**	0.51	**0.51**
KADID-10k	-	0.94	0.89	1.07	1.05	1.07	0.87	0.87	0.91	0.92	**0.76**	**0.79**	**0.82**	**0.84**

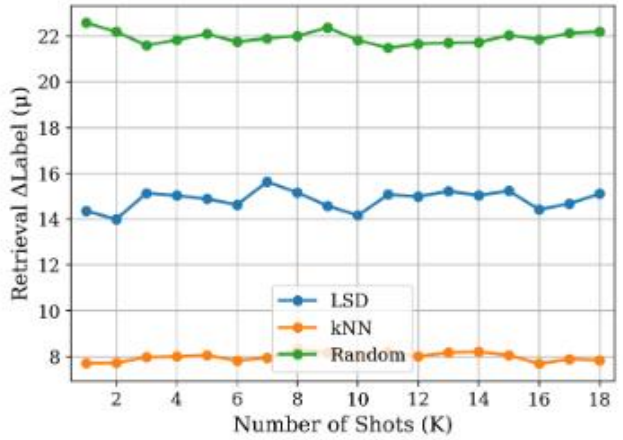
Main Performance vs. K



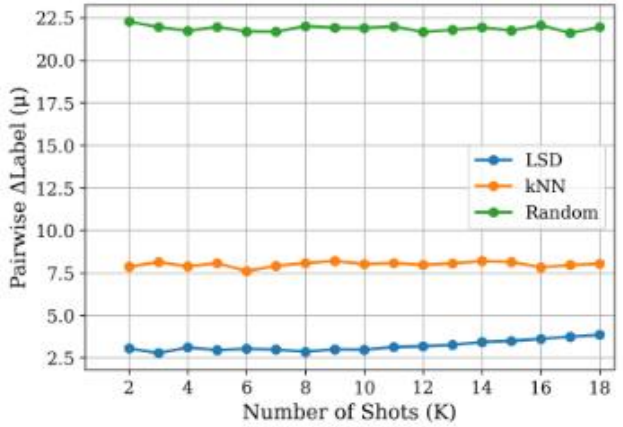
(a) UTKFace (Age Prediction) (b) AVA (Aesthetic Rating) (c) KonIQ-10k (Image Quality) (d) KADID-10k (Image Quality)

Figure 2. **Performance vs. Number of Shots (K) on four datasets.** We plot the MAE as K increases. The results are task-dependent: **(a), (c), (d) Objective Tasks (UTKFace, KonIQ, KADID):** Our LSD policy (blue) consistently outperforms the kNN baseline (orange). **(b) Subjective Task (AVA):** The kNN baseline, which is based on visual similarity, consistently outperforms LSD.

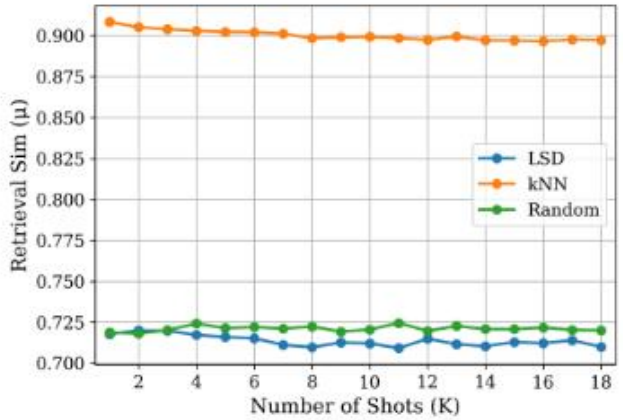
Demonstration Set Analysis



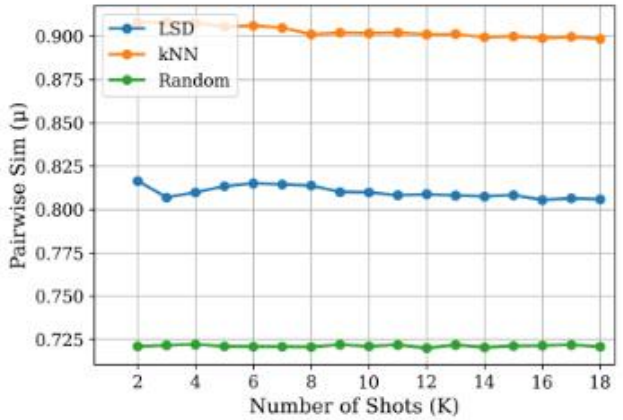
(a) MAE of Demo Labels vs. Query



(b) Pairwise Label MAE



(c) Demo-Query Feature Similarity



(d) Pairwise Feature Similarity

Qualitative Analysis

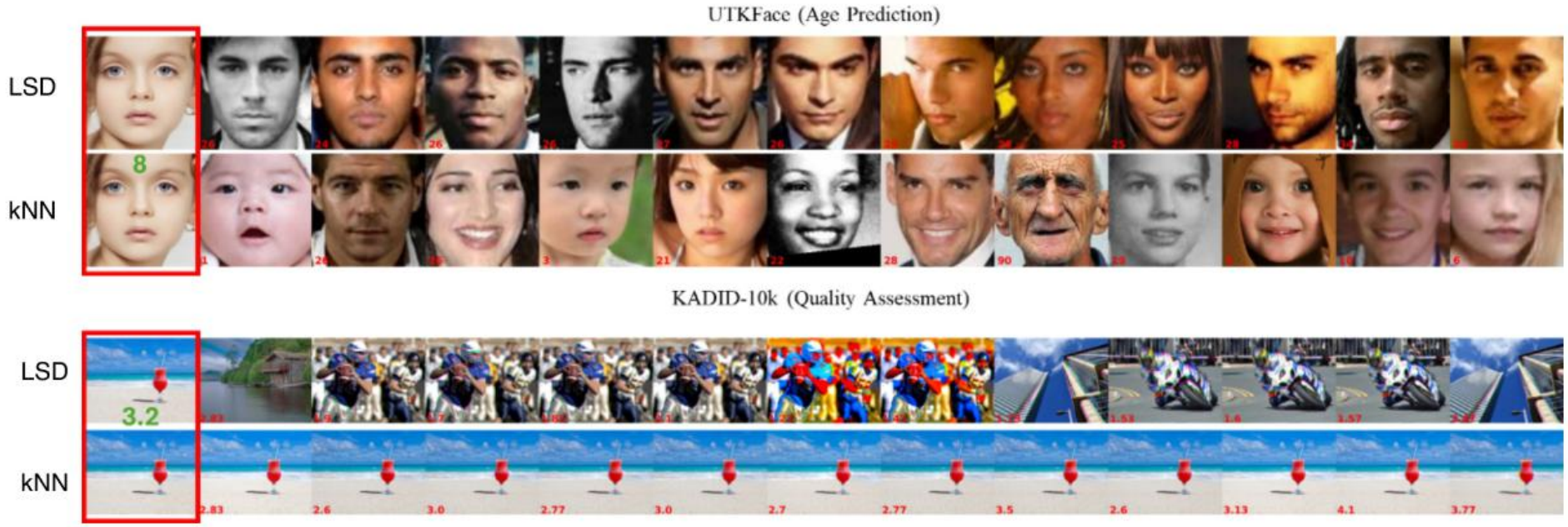


Figure 4. **Qualitative Comparison of Selected Demonstrations** ($K = 12$). (a) **UTKFace:** For an 8-year-old query, kNN selects only images with highly similar features (e.g., other young children). LSD selects a diverse spectrum of visual features (e.g., varied ages, genders, and lighting conditions) to build a richer context. (b) **KADID-10k:** For a motion-blurred query, kNN selects only other distorted versions of the *same source image*. LSD selects a varied set, including the pristine original and images with *different distortion types* from *different source images*, defining the quality boundaries.

Ablation Study

Table 3. **Ablation Study on Decoder Input Strategy (MAE ↓).** We compare our query-centric model against a standard decoder-only model (Concat Input) on UTKFace for $K \in \{4, 8, 16\}$, both using $L = 2$ layers. We also note the qualitative policy behavior.

	MAE ↓			
Decoder Input Strategy	K=4	K=8	K=16	**Policy Behavior**
Query-Centric	**6.27**	7.05	**6.64**	Query-specific demos
Concat Input	7.01	**6.42**	7.74	Non-query-specific

Thanks