

Selection-Centric AI: Learning to Select the Right Context for LLMs

From Data Selection to Context Selection to Deployable Systems

June 11, 2026

Speaker Information

Presenter	Xubin Wang wangxubin@ieee.org wangxubin.site
Research	Selection-Centric AI: data selection, LLM context optimization, deployable systems
Output	14 papers (incl. 3 preprints), 580+ citations, ICML / ACM CSUR / IEEE TKDE
Key Works	RDES (ICML 2025) · Meta-Sel (EMNLP under review) · On-Device AI Survey (ACM CSUR)

§1 About Me

Education · Research trajectory · Positioning

§2 RDES: RL for Demonstration Selection

Problem formulation · Related work · MDP design · Reward · Results · Ablation

§3 Meta-Sel & Selection-Centric AI

Motivation · Core idea · Features · Framework · Benchmark · Unified view

§4 Summary & Roadmap

Key takeaways · Research roadmap · Vision · Q&A

§1 About Me

Education & Training

- ▶ **BSc** Computer Science, Tianjin Normal Univ. (2015–2019)
- ▶ **MSc** Computer Science, Jilin Univ. (2019–2022)
- ▶ **PhD** Computer Science and Technology (2022–2026)
Degree conferred by **Hong Kong Baptist University**;
jointly trained with **Beijing Normal University**
Supervisor: Prof. Weijia Jia (IEEE Fellow)
- ▶ **Visiting Researcher**, Dept. of Computing, **The Hong Kong Polytechnic University** (2025)
Host: Prof. Qing Li (IEEE Fellow, Head of Dept.)

Research at a Glance

Papers	14 (incl. 3 arXiv)
Citations	580+
100+ cited	3 papers
Top venues	ICML · ACM CSUR IEEE TKDE
Open source	460+ Stars

One-line positioning

Selection-Centric AI:

select **better** ⇒ infer **faster** ⇒
deploy **reliably**.

§1 Research Trajectory: From Optimization to Deployment

Phase A · 2019–2022	Phase B · 2022–2026	Phase C · 2025–
Optimization Foundations Evolutionary algorithms for high-dimensional feature selection and imbalanced classification. Key works: SaWDE (KBS), MEL (TKDE)	In-Context Learning RL-based demonstration selection and meta-learning ranking for LLM inference quality. Key works: RDES (ICML 2025) , Meta-Sel	Systems & Deployment On-device AI, edge-cloud collaboration, and efficient inference frameworks. Key works: ACM CSUR (ESI Hot Paper)

Connecting thread: **selection** under **resource constraints** —from raw data to runtime prompts.

From the Big Picture to a Concrete Problem

§1 showed us the **three-layer selection-centric vision**:

data-level → context-level → deployment-level.

Now we zoom into the core layer: *context-level* selection.

The central question:

Given a test query and a pool of labeled examples,
which k demonstrations should go into the prompt
to maximize LLM accuracy?

Let's formalize and solve it.

§2 Formal Problem: ICL Demonstration Selection

Problem Setup

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a labeled demonstration pool, x_q a test query, $k \ll N$ the budget. A **selection function** $S : \mathcal{X} \times \mathcal{D} \rightarrow \mathcal{D}^k$ outputs k ordered demonstrations. The LLM predicts:

$$\hat{y}_q = \text{LLM}(x_q \mid S(x_q, \mathcal{D}))$$

Optimization Objective

$$S^* = \arg \max_{S: |S(\cdot)|=k} \mathbb{E}_{(x_q, y_q)} [\mathbb{I}(\hat{y}_q = y_q)]$$

Two Core Challenges

1. **Query-adaptivity**: S^* depends on the specific x_q —no fixed set works universally.
2. **Relevance–Diversity**: similarity retrieval oversamples majority class \Rightarrow poor minority coverage.

Why This Matters

- ▶ Scarce or expensive labeled data
- ▶ No budget or time for fine-tuning
- ▶ Wrong demos can **halve** accuracy (Liu et al., 2022)

§2 Motivation: Similarity-Only Selection Fails



Empirical Failure Mode

Pure similarity retrieval repeatedly selects the **same majority-class** or near-duplicate examples.

Consequence

Poor minority coverage; LLM never sees counter-examples; calibration degrades on rare classes.

Why This Matters for RDES

- ▶ ICL accuracy is **not monotonic** in demo count —bad demos hurt more than more demos help.
- ▶ A good selector must **actively** avoid redundant or harmful demonstrations.
- ▶ This failure mode is **systematic**, not anecdotal —occurs across datasets and LLM families.

§2 Related Work: Where RDES Fits

Existing Methods (10 baselines in RDES)

Type	Methods & Key Idea
Prompt Eng.	ZS (no demos), KP (knowledge), L2M (decompose), CoT (chain reasoning), SF (self-refine)
Demo Sel.	FS (random), FSC (FS+CoT), AES (RL), RDS (2-stage DPP), ADA (uncertainty+diversity)

Consistent with experimental setup (§2.8).

Limitations of Prior Work

- ▶ **Prompt Eng.:** ignore demonstrations entirely —no selection mechanism.
- ▶ **FS / FSC:** random selection —no query-adaptivity, no diversity optimization.
- ▶ **AES / RDS / ADA:** use static heuristics or two-stage DPP —score demos *independently*, not as a *joint* set.

→ *Need a principled way to jointly select & adapt demonstrations.*

§2 RDES Core Idea: Selection as Sequential Decision

We reformulate demonstration selection as a **sequential decision process**: select one demonstration at a time, conditioning on the evolving context.

Relevance

Diversity

Adaptivity

Goal: For every query x_q , dynamically build k demonstrations that jointly maximize accuracy *and* label coverage.

Key design principle: keep the LLM **frozen** —learn only a lightweight **RL policy** for selection.

§2 MDP Formulation

MDP Components

Component Definition

State s_t	Query TF-IDF, selected-set embedding, LLM output distr., diversity
Action a_t	Index of next demo from candidate pool \mathcal{K}
Transition	Append demo, re-query LLM, update diversity
Reward r_t	$\mathbb{I}(y_q = \hat{y}_{t+1}) + \lambda(t) \cdot \Delta D_t$
Horizon	$T = k$ (select exactly k demos)

State Design

1. ϕ_x : TF-IDF features of query x_q
2. ϕ_E : mean embedding of selected set E_t
3. ϕ_y : LLM output distribution over classes
4. D_t : normalized label diversity $\frac{|\mathcal{L}(E_t)|}{k}$

Training

RL-based policy learning with discretized states:

- ▶ ϵ -greedy exploration with annealing from 1.0
- ▶ LLM **frozen**; only lightweight policy trained (tabular Q / PPO)
- ▶ RL policy via LLM API calls; no model weights stored



Loop: LLM feedback \rightarrow correct earlier choices.

§2 Reward Design: Accuracy + Annealed Diversity

Composite Reward

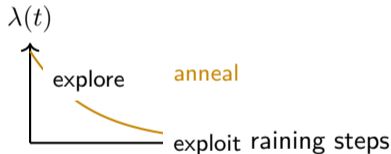
$$r_t = \underbrace{\mathbb{I}(y_q = \hat{y}_{t+1})}_{\text{Accuracy gain}} + \lambda(t) \underbrace{(D_{t+1} - D_t)}_{\Delta \text{Diversity}}$$

$D_t = \frac{|\mathcal{L}(E_t)|}{k}$: normalized label diversity (unique labels in selected set).

Annealing Schedule & Rationale

$\lambda(t) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})e^{-\eta t}$ (decays over training steps)

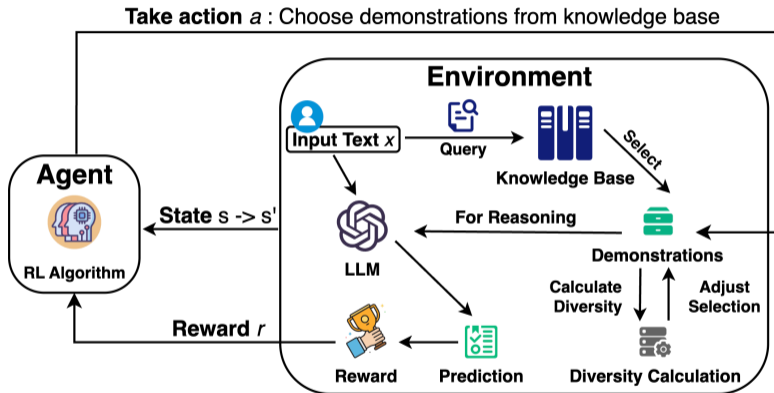
- ▶ **Early** (λ high): explore diverse label coverage, escape majority-class traps
- ▶ **Late** ($\lambda \rightarrow 0$): refine toward accuracy maximization
- ▶ Principled exploration-exploitation curriculum; analogous to entropy regularization in SAC



Complexity

Training $\mathcal{O}(NkT_{\text{ep}})$ LLM calls (RL exploration) Inference: **1** LLM call/query
Storage: lightweight RL policy only.

§2 RDES Framework Overview



Pipeline: Query \rightarrow RL policy selects demos \rightarrow frozen LLM inference \rightarrow composite reward updates policy. Test-time: 1 LLM call (policy selects demos without LLM).

§2 Experimental Setup

We evaluate across diverse settings: 4 intent datasets, 12 LLMs (open + closed), 10 baselines.

Datasets & Metrics

Dataset	#Classes	KB Size	Metric
Banking77	77	9,003	Accuracy
CLINC150	150	18,000	Accuracy
HWU64	64	8,828	Accuracy
Liu54	54	20,382	Accuracy

Mean over 3 seeds; challenge split via precision margin.

Implementation (Q-learning)

$k=5$, $\lambda_{\max}=0.5$, $\eta=0.002$, $\gamma=0.9$. Discretized state; ϵ -greedy with exponential decay.

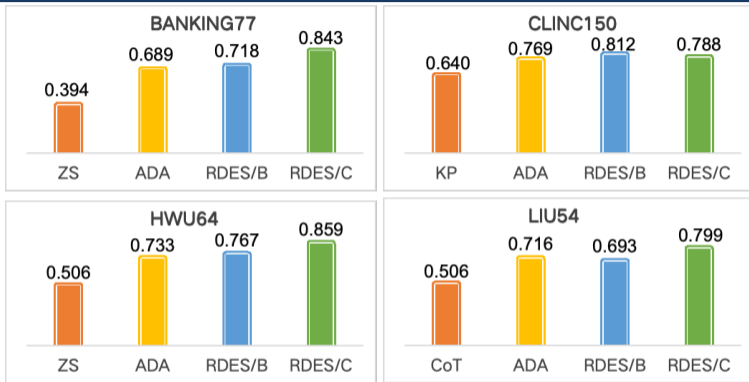
Models & Baselines

- ▶ **Closed** (4): GPT-3.5-turbo, Doubao-lite/pro-4k, Hunyuan-lite
- ▶ **Open** (8): Gemma-2-2B/9B, LLaMA-3.2-1B/3B, LLaMA-3-8B, Qwen-2.5-7B/14B, Qwen-1.5-72B
- ▶ **10 Baselines**: ZS, KP, L2M, CoT, SF, FS, FSC, AES, RDS, ADA

Training Regime

RL policy trained via LLM API calls; **LLM frozen** —only lightweight policy stored. Inference: trained policy selects demos, then single LLM call for prediction.

§2 Main Results: Consistent Gains Across 12 LLMs



Closed- and open-source model averages across 4 intent datasets. **RDES/C** (RDES + chain-of-thought) consistently outperforms all 10 baselines.

Headline Numbers

+12.8% avg. gain on closed-source models. 12 LLMs; overhead **<0.1%**. RDES/C wins **all 4** datasets.

§2 Ablation Study & RDES Summary

Ablation: Diversity Reward (Closed-Source Avg.)



Diversity reward drives largest gains: +23.8% (Banking77), +15.7% (CLINC150), +22.3% (HWU64), +22.3% (Liu54). Annealing $\lambda(t)$ is critical —fixed λ hurts both exploration and final accuracy.

RDES: Contributions & Outlook

- C1** First formulation of ICL selection as a **query-adaptive sequential MDP**.
- C2** Composite reward with **annealed diversity** —principled exploration-exploitation curriculum.
- C3** **+12.8%** avg. gain across 12 LLMs; <0.1% runtime overhead.

Limitation: RDES relies on **online RL training** —requires many LLM calls for exploration, making retraining slow and expensive.

Next (§3): Can we eliminate the training overhead while preserving selection quality?

→ **Meta-SEL.**

§3 Motivation: From Online RL to Offline Meta-Learning

RDES: Online RL



- × **Expensive training** LLM-in-the-loop RL exploration
 - × **Slow retraining** Online RL \Rightarrow hours, not seconds
 - × **Non-deterministic & opaque** ϵ -greedy, black-box Q-values
- \Rightarrow **Deployment bottleneck: retraining cost**

Meta-Sel: Offline Meta-Learning



- ✓ **Zero LLM calls during selection**
 - ✓ **Train once, deploy anywhere** Only 1 LLM call for prediction
 - ✓ **Deterministic & auditable** Interpretable weights
- \Rightarrow **Instant deployment, matches/exceeds RDES in 16/20 settings**

Core Question: Recover RDES-quality selection via cheap, pre-computable features — zero LLM calls for selection; only 1 forward pass for prediction.

§3 Meta-Sel: Supervised Reformulation of Selection

Key Insight

For classification ICL, a maximally useful demonstration **shares the same label as the query**:

$$\text{Score}(x_c, x_q) = P(y_c = y_q \mid x_c, x_q)$$

Ranking by label-agreement probability maximizes expected accuracy (“one-match-suffices” abstraction). Proof in our paper.

Three Ingredients of Meta-Learning

1. **Meta-dataset**: pairs (x_c, x_q) with features f
2. **Meta-label**: $\ell_{cq} = \mathbb{I}[y_c = y_q]$ —class agreement
3. **Meta-learner**: logistic regression h_θ , trained with BCE

Method Landscape: 4 Families

Family	Test LLM	Training
Prompt Engineering	0	None
Heuristic	0	None
Model-Based	1	2nd-order
RL	$\gg 1$	Online
Meta-Sel (ours)	0	Offline

Key Distinction: Heuristics miss label structure; RL needs costly online exploration. Meta-Sel learns offline, deploys with **only 1 LLM call** for prediction — ~ 10 ms/query selection, 16/20 win/tie vs RDES.

§3 Meta-Sel: Feature Engineering & Meta-Learner

Feature & Meta-Learner

Feature: TF-IDF cosine similarity (CPU-computable)

$$\text{sim}(x_c, x_q) = \frac{\phi(x_c)^\top \phi(x_q)}{\|\phi(x_c)\|_2 \|\phi(x_q)\|_2}, \quad f \in \mathbb{R}^1$$

Meta-Learner: Logistic regression

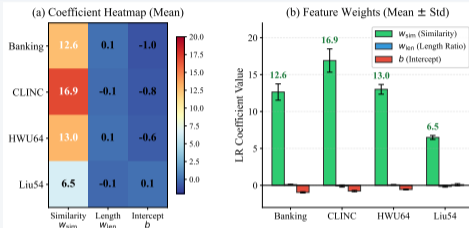
$$h_\theta(f) = \sigma(\theta \cdot \text{sim} + b), \quad \theta \in \mathbb{R}$$

Trained with BCE loss on pairwise data. Inference: $\mathcal{O}(N)$

Complexity vs. RL Baselines

	Training	Inference
Meta-Sel	$\mathcal{O}(Q C)$ CPU	$\mathcal{O}(N)$
RDES	$\mathcal{O}(kT_{ep}N)$ LLM	$\mathcal{O}(kN)$ LLM
AES (RL-based)	$\mathcal{O}(N)$ LLM	$\mathcal{O}(N)$ LLM

Feature Weight Visualization

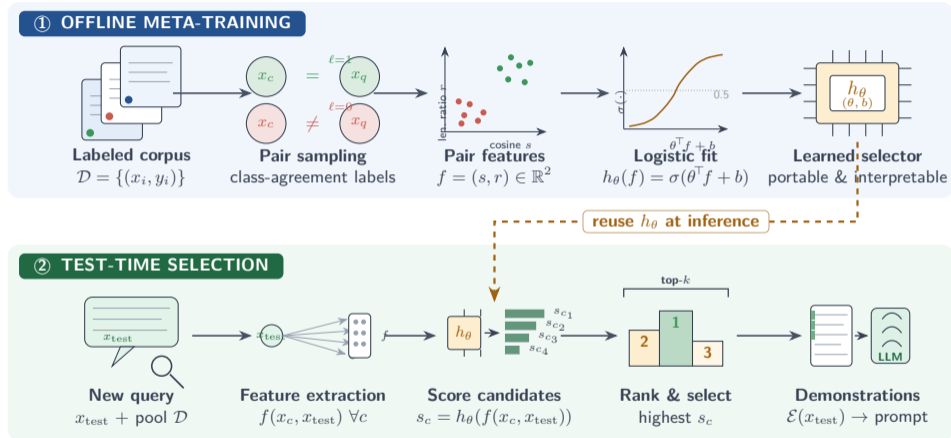


TF-IDF similarity dominates across all datasets.

Interpretability & Design Principle

- ▶ Learned weight **directly auditable** —no black-box
- ▶ **Explainable:** exactly *why* each demo was selected
- ▶ Single feature + 1-parameter model = maximum simplicity

§3 Meta-Sel: Framework Overview



A two-phase framework: train a lightweight selector offline, then reuse it for zero-cost LLM selection at inference.

§3 Benchmark: Setup

Experimental Protocol

- ▶ **4 datasets:** Banking77, CLINC150, HWU64, Liu54
- ▶ **5 LLMs:** 4B–20B (Gemma3, Qwen3, LLaMA2, DeepSeek-R1, GPT-OSS)
- ▶ **12 methods** across 4 families: Prompt Eng., Heuristic, Model-Based, RL
- ▶ $k=5$, challenge split 1k queries; paired t -test, $p < 0.05$

Why Meta-Sel Wins

- ▶ **Learns from labels** —heuristics miss label structure
- ▶ **Offline training** —RL needs costly exploration
- ▶ **Zero inference cost** —no LLM calls for selection
- ▶ **Deterministic** —no sampling variance

Performance Highlights

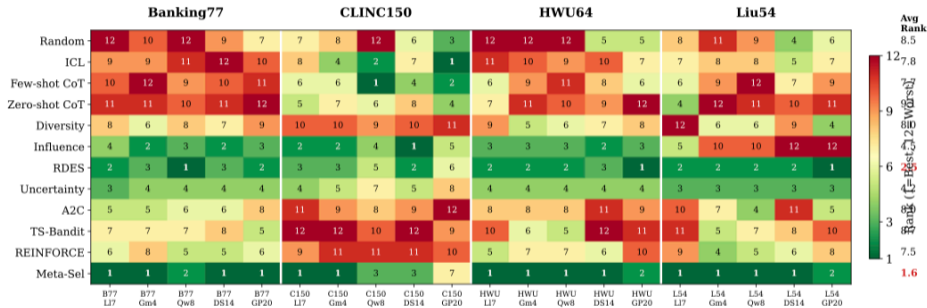
- ▶ **Top-3 in 18/20** model–dataset settings
- ▶ Matches/beats RDES in **16/20** (15W + 1T)
- ▶ Macro-avg accuracy: **82.5%** (vs RDES 80.8%)
- ▶ ~ 10 ms/query, < 5 s CPU, deterministic

Key Insight

A 1-parameter model with a single TF-IDF feature **matches or beats** RL-based selection (RDES) while eliminating all test-time LLM overhead.

§3 Benchmark: Ranking Heatmap

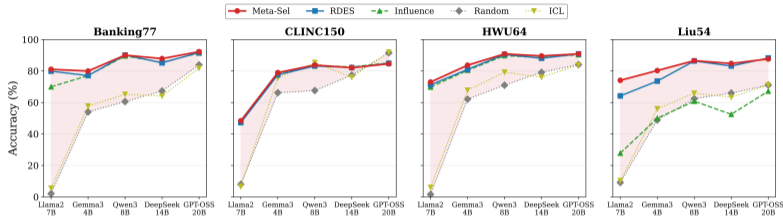
Method Ranking Across All Evaluation Settings



Ranking heatmap: Meta-Sel ranks candidates by predicted label agreement. Clear separation between high- and low-agreement candidates. Top-3 in 18/20 model-dataset settings confirms the method's consistency.

§3 Benchmark: Model-Scale Effect & Ablation Analysis

Model-Scale Effect (by Dataset)

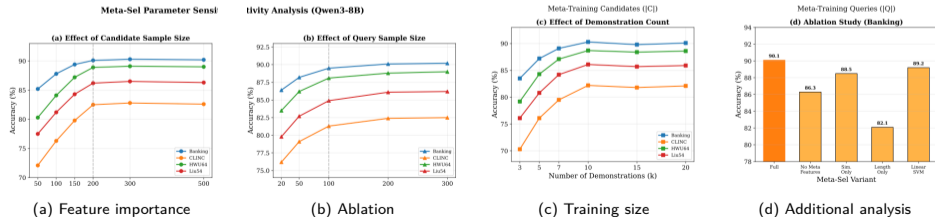


Key Findings

Model-Scale
Gap ↓ as model ↑;
4B–8B benefit most.

Ablation
TF-IDF dominant;
LR adds refinement;
saturates at ~18k.

Comprehensive Ablation Analysis



(a) Feature importance

(b) Ablation

(c) Training size

(d) Additional analysis

§3 The Efficiency–Effectiveness Sweet Spot

Meta-Sel vs RDES: Systematic Comparison

Property	RDES	Meta-Sel
Accuracy (macro-avg)	80.8%	82.5%
Inference LLM calls	1	1
Selection latency	~10 ms	~10 ms
Training regime	Online RL (LLM-in-loop)	Offline CPU (<5s)
Deterministic	✗	✓
Auditable	✗	✓

Why This Matters for Production

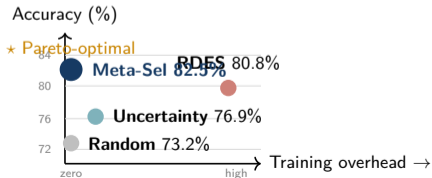
- ▶ **Predictable latency:** no RL rollouts at test time
- ▶ **Audit trail:** feature weights reveal selection rationale
- ▶ **Model-agnostic:** zero model-specific tuning
- ▶ **Small-model friendly:** biggest gains on 4B–8B models
- ▶ **Reproducible:** deterministic rankings enable CI/CD testing

Practical Deployment Principle

Train a lightweight offline selector first.

Add online optimization *only if* the residual quality gap justifies the cost — typically it does not.

Bottom line: cheap offline training yields better results than expensive online RL.



§4 Summary of Contributions

Phase A · 2019–22
Data-Level Selection

Phase B · 2022–26
Context-Level Selection

Phase C · 2025+
Deployment-Level

Feature Selection (Evol.)

- ▶ SaWDE · adaptive DE for high-dim FS (KBS'21)
- ▶ MEL · multi-level evolution, **TKDE'24**

ICL Demo Selection

- ▶ RDES · RL-based sequential MDP, **ICML'25**
- ▶ Meta-Sel · TF-IDF + LR, zero selection LLM cost

On-Device AI Deploy

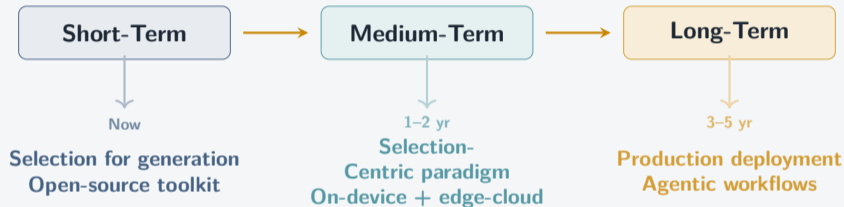
- ▶ **CSUR'25** Survey (ESI Hot Paper)
- ▶ Edge-cloud system selection paradigm

KPIs across the program: **+12.8%** RDES avg. gain | Meta-Sel **82.5%** macro-avg, **1**
LLM call (pred. only) | **14 papers** · **580+** cites | **CSUR'25** ESI Hot Paper

Unifying Principle: *Better Selection at Every Layer* ⇒ *More Reliable, Efficient & Deployable AI*

§4 Research Roadmap

Selection-Centric AI: A 3–5 Year Agenda



Driven by three guiding principles: **One Principle, Three Levels** | **Simplicity at Scale** | **Democratizing Access**

Selection-Centric AI: Learning to Select the Right Context for LLMs

Questions & Discussion

Xubin Wang

wangxubin@ieee.org | wangxubin.site

14 papers · 580+ citations · ICML / ACM CSUR / IEEE TKDE

RDES (ICML 2025) · Meta-Sel (EMNLP under review) · MEL (TKDE 2024) · On-Device AI (CSUR 2025)