Alibaba Qwen Application Business Group

STAR: Similarity-guided Teacher-Assisted Refinement for Super-Tiny Function Calling Models

Jiliang Ni, Jiachen Pu, Zhongyi Yang, Jingfeng Luo, Conggang Hu

conggang.hcg@alibaba-inc.com

The Challenge: Agentic Capabilities on the Edge

The Deployment Gap

LLM Agents (> 7B Params)

Agentic Capability, High Latency, Prohibitive Cost

Tiny Models (< 1B Params)

Function Calling, On-Device, Limited Capacity

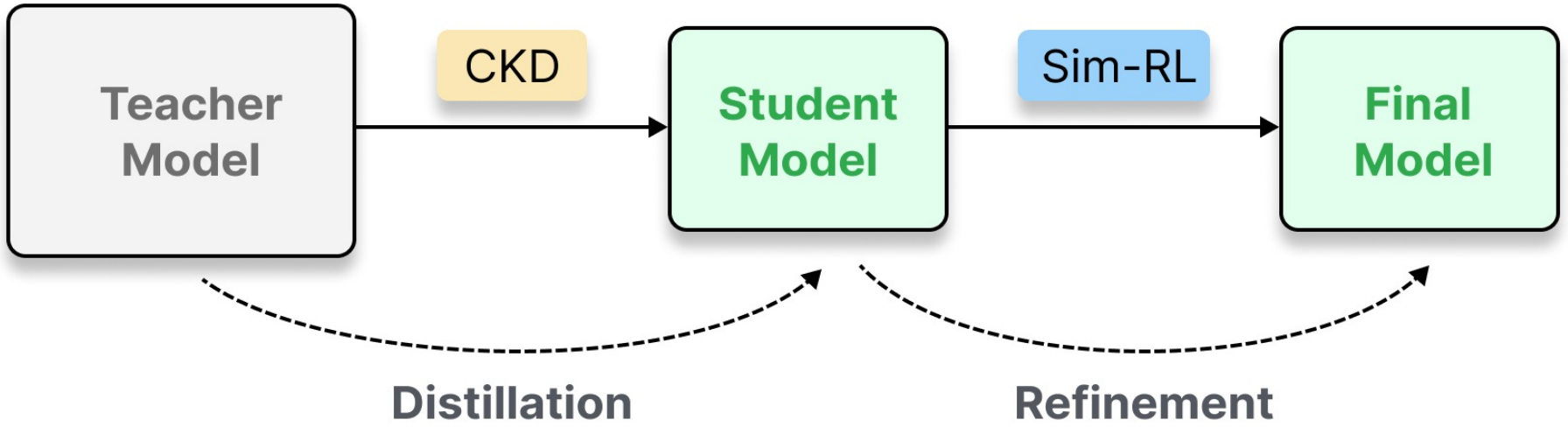
Why Conventional Training Fails?

SFT+RL framework: It fails at the sub-1B scale. SFT causes severe overfitting and poor generalization with memorizing specific tool patterns instead of learning reasoning, which is unfavorable to subsequent RL training.

KD+RL framework: While superior to SFT with less overfitting, it introduces training instability, loss of exploration capacity, and ineffective binary rewards for multi-solution tasks.

We need a framework that effectively transfers reasoning capabilities, not just output patterns.

The STAR Methodology



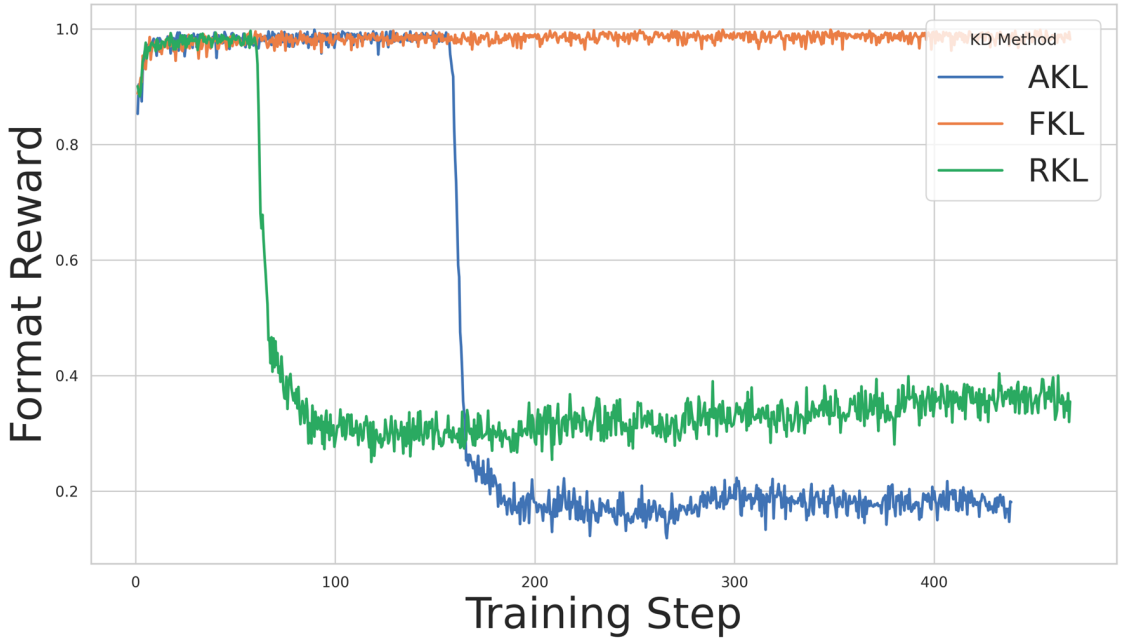
Phase 1: Distillation (CKD)

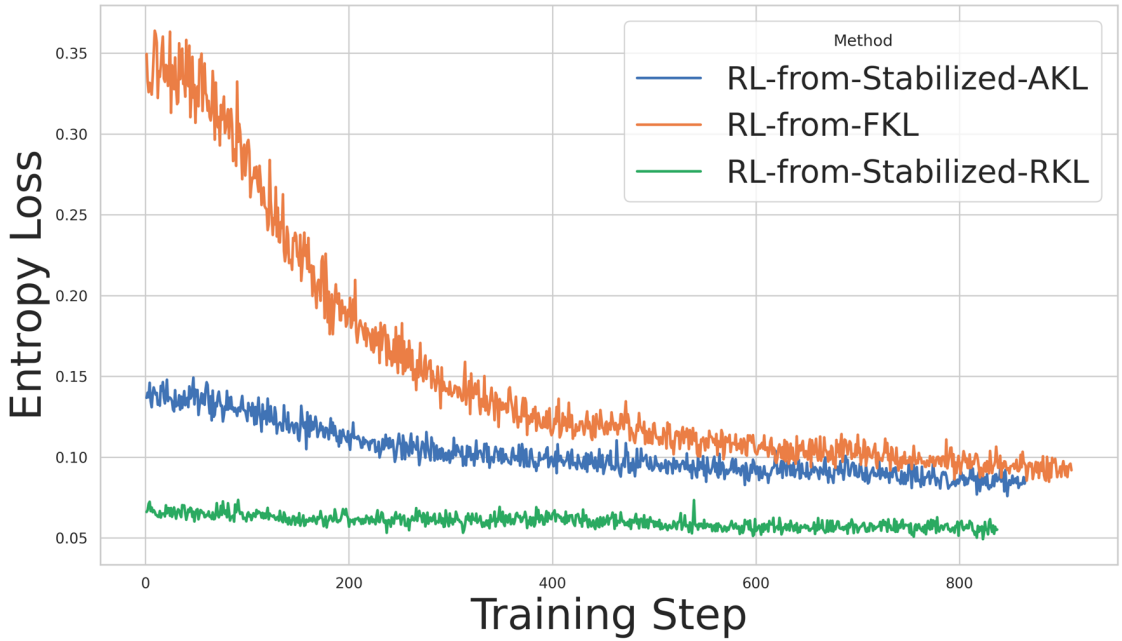
Provides training stability and preserves policy entropy for exploration in the downstream RL.

Phase 2: Refinement (Sim-RL)

Provides a robust learning signal and enhances policy optimization.

Constrained Knowledge Distillation (CKD)





The problem with Top-k **RKL / AKL**

Problem 1: Instability with top-k truncation.

Problem 2: Low entropy during RL training

$$\mathcal{L}_{\text{CKD}} = \mathcal{L}_{\text{FKL-k}} + \underbrace{\lambda_{\text{tail}}\mathcal{L}_{\text{tail}}}_{\textit{Tail penalty}}$$

Penalizes over-confidence

Extends exploration space

Similarity-guided Reinforcement Learning (Sim-RL)

Format Reward

Binary check for valid JSON format, function name

R = {0, 1}

Function Call Reward

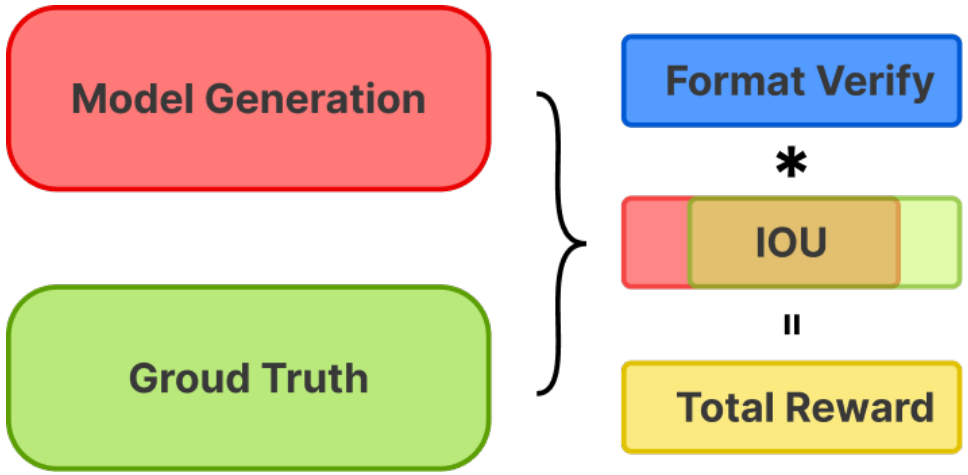
IoU-style matching with argument-level Similarity

R = Sim(P, G) / |P ∪ G|

Response Reward

ROUGE-L F1 score for natural language responses

R = ROUGE-L(P, G)



$$R = \underbrace{(R_{\text{format}} - 1)}_{\text{format term}} + \underbrace{R_{\text{format}} \cdot (R_{\text{fc}} + R_{\text{response}})}_{\text{answer term}}$$

Main Results: New SOTA for Super-Tiny Models

Method	Overall Acc	Non-Live Acc	Live Acc	Multi Turn Acc
Standard methods				
Base-model	47.33	71.81	65.66	1.88
SFT	44.58	66.29	62.15	1.62
SFT-think	47.59	71.54	64.46	4.50
FKL	49.51	76.44	65.93	5.12
Recent methods				
ToolRL	47.35	64.81	66.55	6.75
LUFFY	49.23	76.75	64.59	5.48
GKD	47.32	67.62	67.61	3.25
Our methods				
CKD	49.84	75.92	66.15	5.62
Sim-RL	49.35	75.21	67.39	3.25
SFT+Sim-RL*	50.41	76.27	66.99	6.13
CKD+Sim-RL	**51.70**	**78.65**	**68.19**	**7.00**

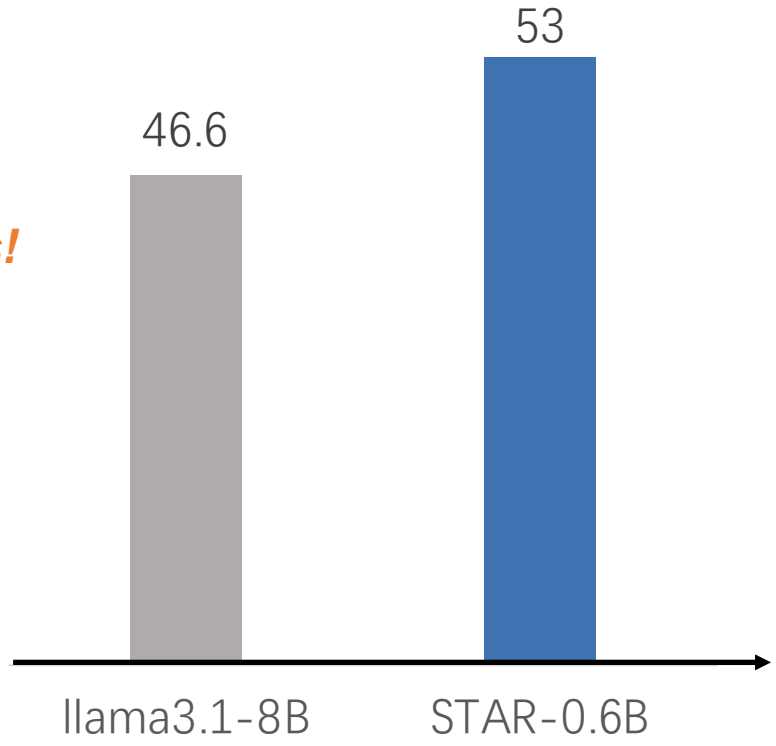
BFCLv3

Method	Summary	Atom	Single-Turn	Multi-Turn	Similar API	Preference
Standard methods						
Base-model	27.20	37.70	19.50	10.00	36.00	6.00
SFT	2.10	1.70	0.50	0.00	14.00	0.00
SFT-think	28.70	42.30	14.00	9.00	34.00	10.00
FKL	36.80	52.30	16.00	16.00	42.00	**22.00**
Recent methods						
ToolRL	29.40	45.00	12.50	10.00	34.00	4.00
LUFFY	44.40	59.30	26.50	26.00	50.00	**22.00**
GKD	40.10	54.00	21.50	23.00	46.00	**22.00**
Our methods						
CKD	39.00	55.00	21.00	19.00	48.00	10.00
Sim-RL	39.30	53.30	23.50	21.00	52.00	10.00
SFT+Sim-RL*	38.90	53.00	21.50	21.00	46.00	18.00
CKD+Sim-RL	**53.00**	**69.30**	**35.00**	**32.00**	**62.00**	20.00

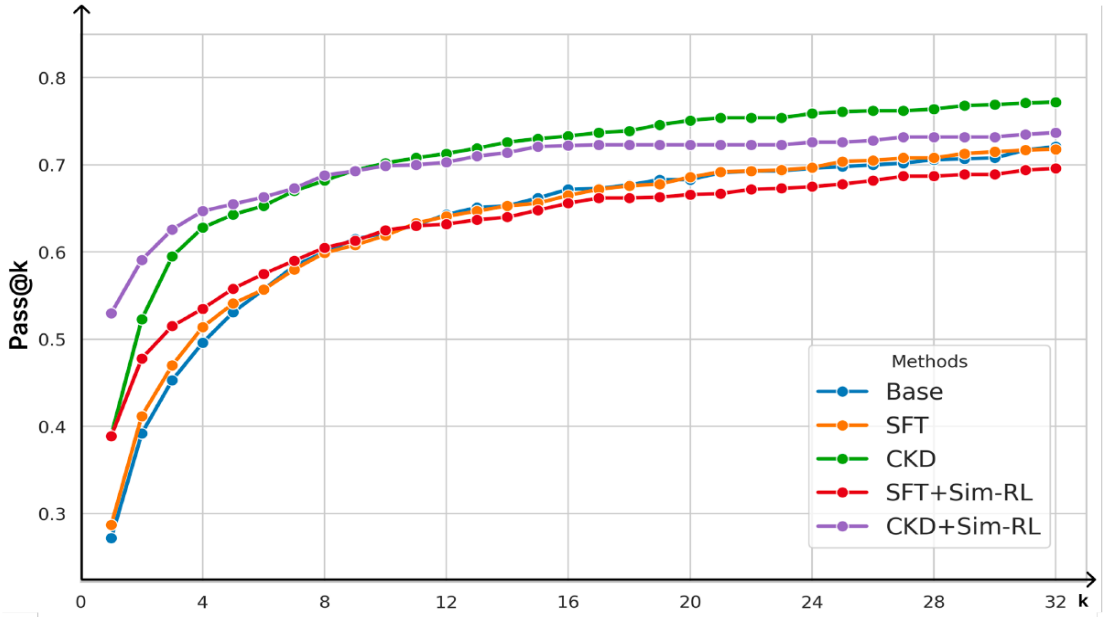
ACEBench Normal

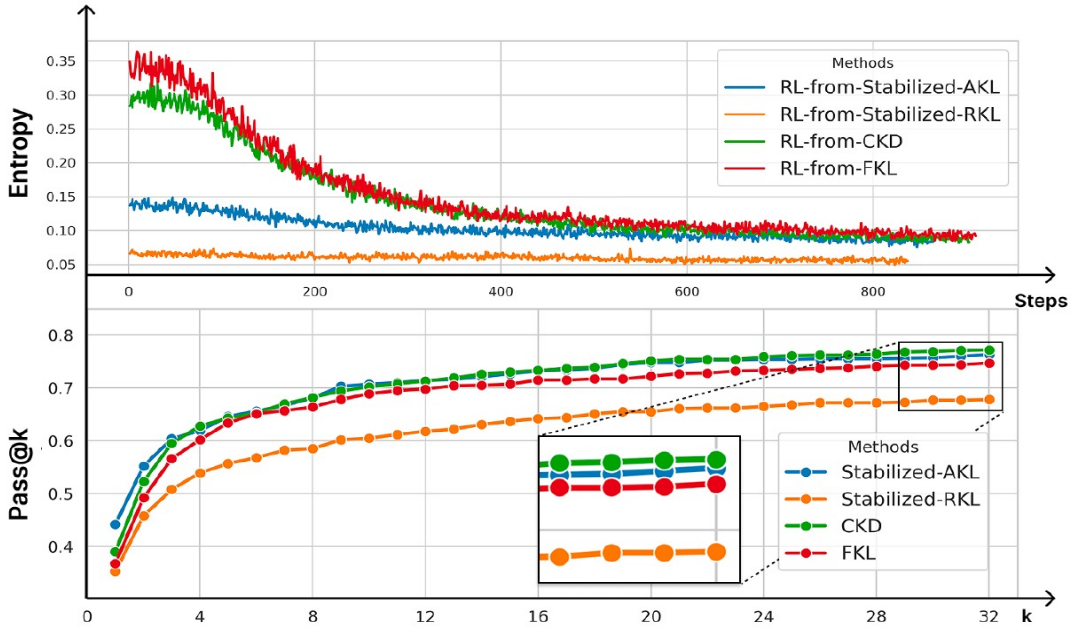
Significant Improvements!

Outperform larger models



Why CKD+Sim-RL works better on super-tiny models?



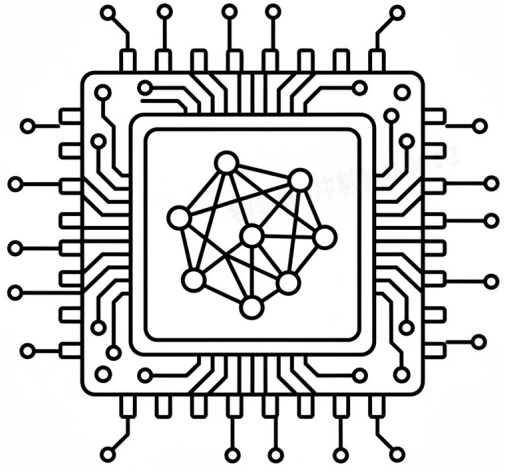


CKD is a Vastly Superior Initializer than SFT

With higher *Pass@K* and higher *Entropy*

(High Potential) (High Explorability)

Paving the Way for Ubiquitous AI Agents



- **Democratization:** High-quality agentic capabilities on edge devices.
- **Efficiency:** Massive reduction in inference cost and latency.
- **Specialization:** Small, specialized models can exceed generalist large models.

Thank You