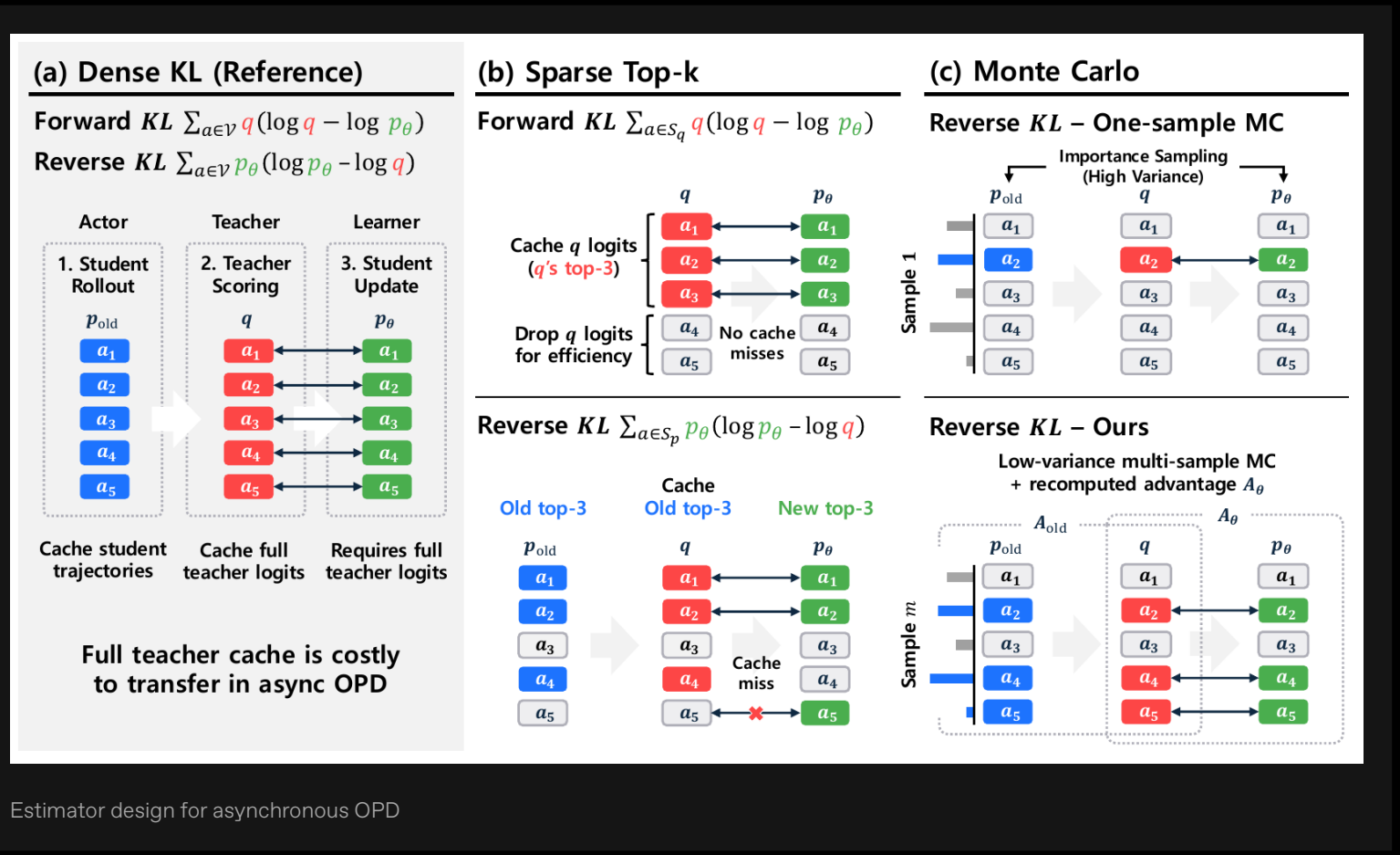
FURIOSA

AsyncOPD: How Stale Can On-Policy Distillation Be?

Wonjun Kang*, Kevin Galim*, Seunghyuk Oh, Minjun Kang, Sanghyun Park, Donghoon Kim, Minjae Lee, Minseo Kim, Rishabh Tiwari, Yuchen Zeng, Hyung Il Koo, Kangwook Lee

FuriosaAI · Ajou University · UC Berkeley · Microsoft Research · KRAFTON · Ludo Robotics

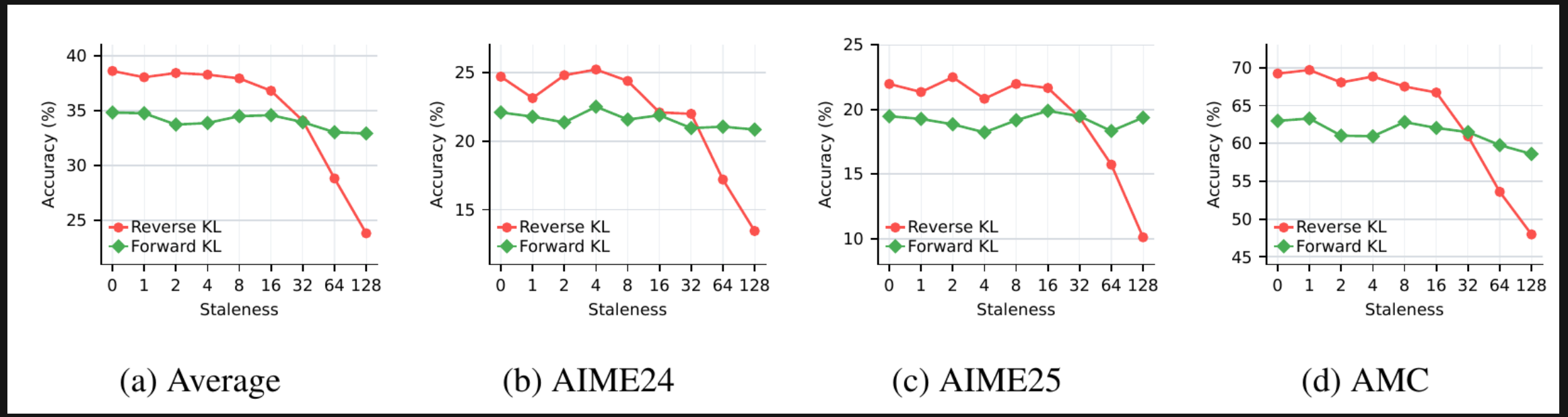
Asynchronous OPD Introduces Stale-policy Data



Estimator design for asynchronous OPD

- OPD trains a student on its own rollouts guided by teacher feedback, but rollouts can dominate training time for reasoning workloads.
- Asynchronous pipelines decouple rollout generation from learner updates, which introduces stale-policy data.
- Full-vocabulary teacher logits are too expensive to store or transfer, necessitating finite teacher-score caches.

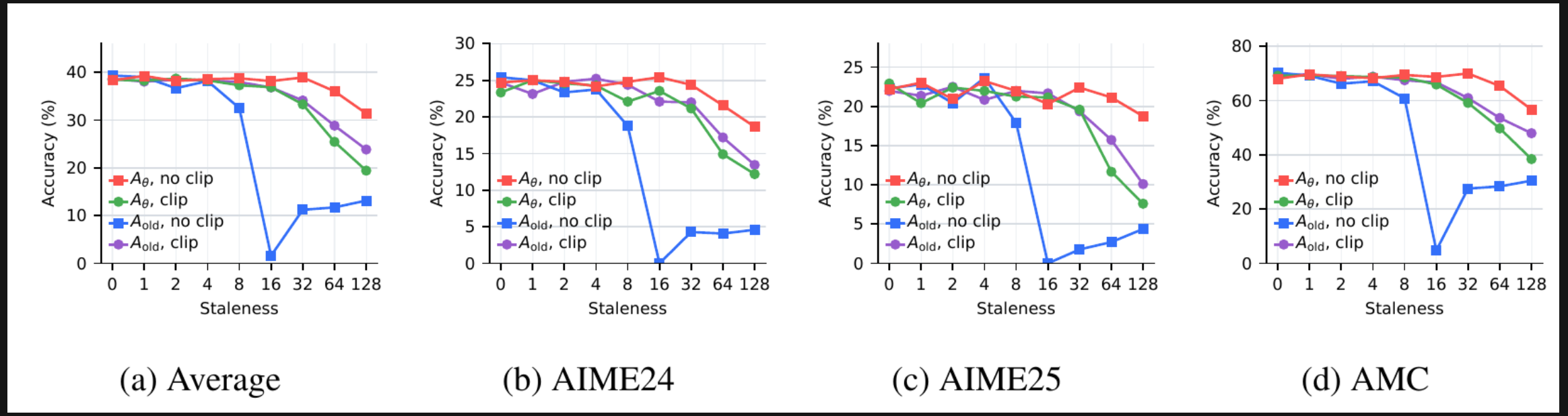
Finding 1: Forward KL Is Robust to Staleness, Reverse KL Is Vulnerable



Accuracy under staleness for forward- and reverse-KL OPD

- Forward KL is teacher-weighted; reverse KL is student-weighted.
- Reverse KL starts higher at zero staleness but degrades faster as staleness grows; forward KL is flatter across the sweep.

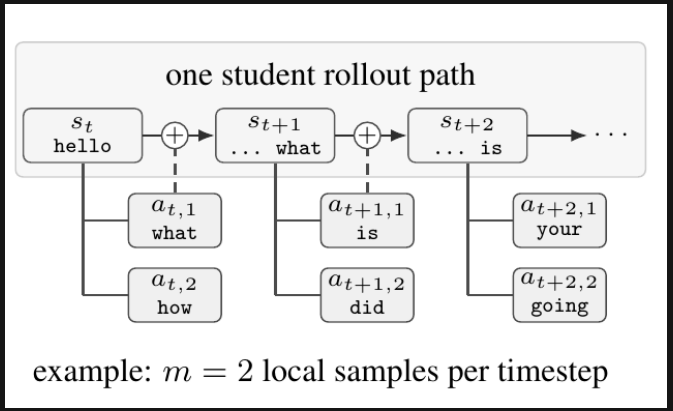
Finding 2: Recompute the Advantage at Learner Time, Without Clipping



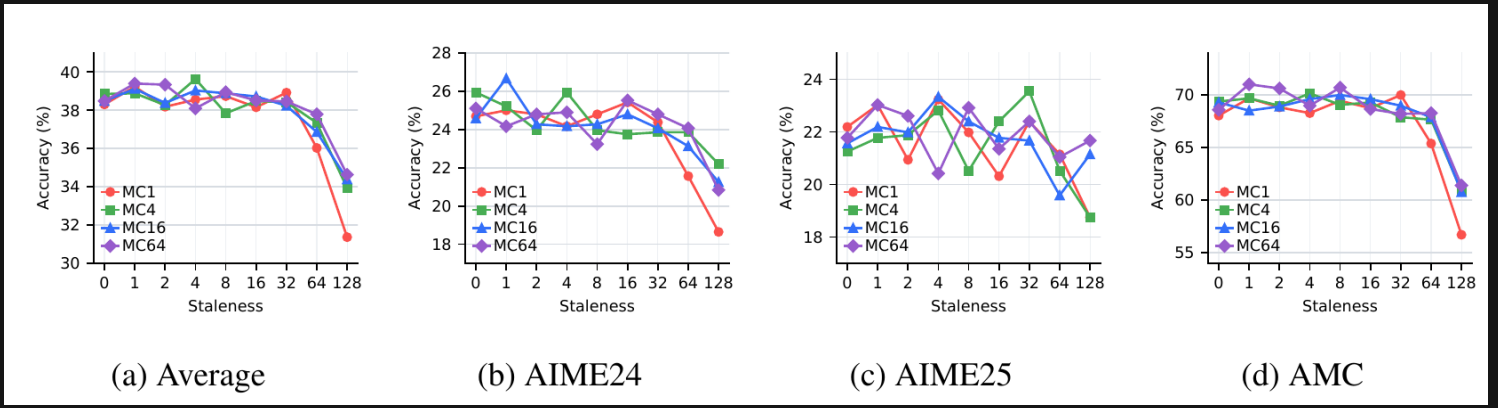
Accuracy under staleness for the advantage-and-clipping ablation

- Recomputing the advantage under the current student at learner time and avoiding clipping gives the most stable performance across the sweep.
- Advanced asynchronous RL surrogates such as decoupled PPO and M2PO do not improve over it.

Finding 3: Multi-sample MC Preserves Correctability While Reducing Variance



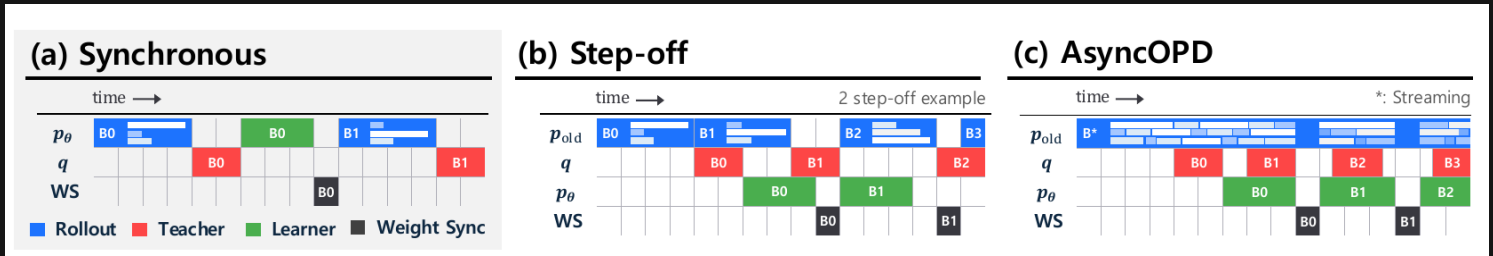
Multi-sample MC with m = 2 local samples per timestep



Accuracy under staleness for multi-sample MC

- Stale student top-k supports are support-mismatched; one-sample MC remains correctable but high-variance.
- Multi-sample MC caches and teacher-scores multiple stale-policy samples at each decoding step.
- Increasing the number of samples improves large-staleness behavior; m = 4 already gives a clear jump.

AsyncOPD: a Fully Asynchronous OPD Pipeline



Scheduler comparison for synchronous OPD, step-off scheduling, and AsyncOPD

- Overlaps student rollout, teacher scoring, and learner updates; streams rollout data to reduce long-tail waiting.
- Improves training throughput by 1.6× to 3.8× over strict synchronous training while reaching comparable accuracy.

Student (MC64, train tok/s)	Strict sync	Two-step-off	AsyncOPD (ours)
Qwen3-1.7B-Base	8.7k	14.2k	**23.4k (2.70×)**
Qwen3-4B-Base	9.5k	12.4k	**15.8k (1.66×)**
Qwen3-8B-Base	7.5k	11.6k	**14.5k (1.94×)**

Thank You

Code: https://github.com/furiosa-ai/async-opd