Post-training LLM Agents

Policy Optimization, Credit Assignment, Self-Distillation, and Evaluation

Seven Topics and the Statistical Questions Behind Them

Yuyao Wang

Department of Mathematics and Statistics, Boston University

Outline

Overview

LLM agents are now trained by letting them interact with environments (search engines, code repositories, websites, tools) and updating the model from the outcomes.

Much of this is a statistics problem in disguise:

- gradients are Monte Carlo estimates from sparse, delayed rewards;
- the data distribution changes as the policy changes;
- supervision often comes from imperfect sources (LLM judges, self-teachers);
- evaluations are small and rarely report uncertainty.

Plan for this talk. Survey the main research directions in agent post-training and, for each one, the statistical questions I find interesting.

Setup

Compared with single-turn RL on reasoning tasks (RLVR):

	RLVR	Agentic RL
Decision	one step	K turns
Actions	a response	tool calls, code, clicks
Transitions	none	environment
Reward	verifier	terminal, often sparse
Context	fixed	grows over turns

See Zhang et al. (2026) for a survey.

A task $x \sim \mathcal{D}$ produces a trajectory

$$p_\theta(\tau \mid x) = \prod_{k=1}^{K} \pi_\theta(a_k \mid h_k)\, P_{\mathrm{env}}(o_{k+1} \mid h_k, a_k),$$

and the objective is

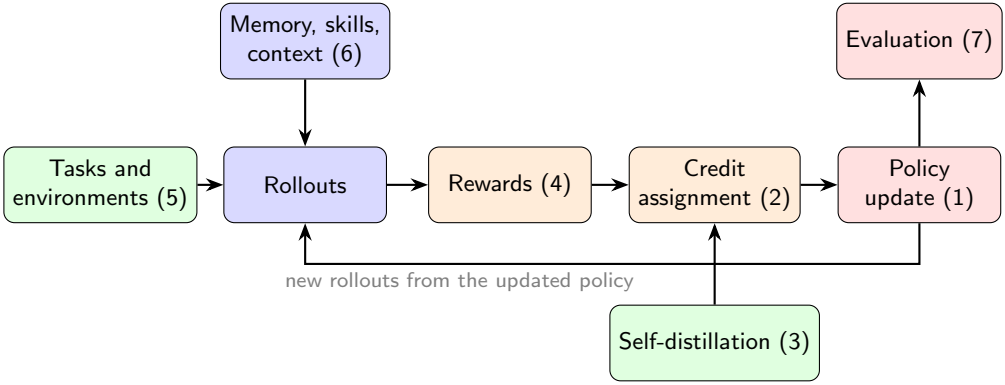
$$J(\theta) = \mathbb{E}_x\, \mathbb{E}_{\tau \sim p_\theta(\cdot|x)}[R(x, \tau)].$$

Because P_{env} does not depend on θ,

$$\nabla J = \mathbb{E}\Big[(R - b) \sum_t \nabla_\theta \log \pi_\theta(y_t \mid s_t)\Big],$$

where the sum runs over the tokens generated by the policy.

The training loop



Numbers refer to the seven topics in the rest of the talk.

Seven topics

	Question	Related statistics
1. Policy optimization	How to update stably from long, stale rollouts	importance sampling, bias of normalization
2. Credit assignment	Which turn deserves credit for the outcome	control variates, stratification
3. Self-distillation	How to learn from the model itself given privileged context	KL estimation, privileged information, pooling
4. Rewards	What to optimize without a verifier	measurement error
5. Environments	Which tasks to train on	experimental design, hierarchical sampling
6. Memory and skills	Knowledge in context vs. in weights	covariate shift
7. Evaluation	Whether an improvement is real	variance components, paired comparisons

1. Policy optimization

Most agent training uses GRPO-style updates (Shao et al., 2024). **Recent work:**

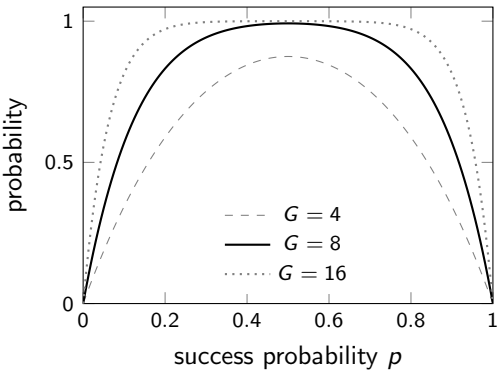
- corrects biases from normalizing by group standard deviation and by length (Liu et al., 2025), and adds clipping and sampling changes (DAPO; Yu et al., 2025);
- moves the importance ratio from the token level to the sequence level (GSPO; Zheng et al., 2025) and studies convergence for agentic settings (Hu et al., 2026);
- studies exploration: entropy collapse (Cui et al., 2025), and whether RL improves on the base model beyond sharpening it (Yue et al., 2025).

The variance problem. In agent training, rollouts are long and often generated asynchronously by an older policy. The trajectory-level ratio

$$w = \prod_{t=1}^{T} \frac{\pi_\theta(y_t \mid s_t)}{\pi_{\text{old}}(y_t \mid s_t)}$$

has variance $e^{T\sigma^2} - 1$ if $\log w \sim \mathcal{N}(-T\sigma^2/2,\ T\sigma^2)$. Clipping controls the variance at the cost of bias. How to choose this trade-off as a function of T and staleness is not well understood.

1. How often a GRPO group gives any gradient



GRPO uses $A_i = (R_i - \bar{R})/s_R$ over G rollouts. If all rewards in a group are equal, every A_i is zero. With i.i.d. Bernoulli(p) rewards,

$$\mathbb{P}(\text{some } A_i \neq 0) = 1 - p^G - (1-p)^G.$$

For $G = 8$ and $p = 0.05$ this is about 0.34.

Two side remarks:

- $R_i - \bar{R} = \frac{G-1}{G}(R_i - \bar{R}_{-i})$, so the group mean is a scaled leave-one-out baseline.
- Dividing by $s_R \approx \sqrt{\hat{p}(1-\hat{p})}$ gives more weight to very easy and very hard tasks.

2. Credit assignment

With one reward at the end of a long trajectory, every token gets the same advantage.

Approaches include:

- turn-level rewards within multi-turn GRPO and PPO (Wei et al., 2025);
- step-level advantages computed from states that repeat across rollouts in a group (GiGPO; Feng et al., 2025);
- hierarchical critics at the turn level (ArCHer; Zhou et al., 2024);
- branching rollouts from shared prefixes, and process reward models.

A 2026 survey organizes about 47 methods by granularity (token, segment, step, turn, multi-agent) and by method (Monte Carlo, temporal difference, model-based, game-theoretic, information-theoretic).

One open question. Heuristic turn rewards can change the optimal policy. Potential-based shaping does not (Ng et al., 1999), but most turn rewards used for agents are not of that form.

2. Baselines as control variates

For the action a_k taken at turn k, any baseline that depends only on the history h_k leaves the gradient unbiased:

$$\mathbb{E}\big[b(h_k)\,\nabla_\theta \log \pi_\theta(a_k \mid h_k)\big] = 0.$$

Several credit assignment methods can be read as different choices of $b(h_k)$:

- *Learned critic*, $b = \hat{V}_\phi(h_k)$: lower variance, but biased when $\hat{V}_\phi$ is wrong.
- *Grouping by state* (as in GiGPO): average the returns of rollouts that reach the same state. This is a stratified, nonparametric estimate of $V(h_k)$.
- *Branching from a shared prefix*: sibling rollouts differ only after a_k, which is the idea of common random numbers.

A natural question. Given a fixed number of rollouts, how should they be split across tasks, turns, and branch points to minimize the variance of the gradient estimate?

3. On-policy self-distillation: background

2015	Knowledge distillation: a student matches a teacher's output distribution (Hinton et al., 2015).
2022	Context distillation: a model given a long prompt teaches the same model without it (Snell et al., 2022).
2024	On-policy distillation (OPD): the student generates, and a teacher scores the student's tokens (Agarwal et al., 2024; Gu et al., 2024).
2025	OPD is used in open-model post-training, e.g. Qwen3's strong-to-weak distillation (Yang et al., 2025), and popularized by a blog post (Lu and Thinking Machines Lab, 2025).
2026	The teacher becomes the model itself with privileged context: reference solutions (Zhao et al., 2026), environment feedback (Hübotter et al., 2026), demonstrations (Shenfeld et al., 2026). Many follow-ups, including combinations with RL, multi-turn agents, surveys, and critical studies.

Reasons for the interest:

- it gives a token-level signal, even when every rollout in a group fails;
- it needs no larger teacher, only one extra forward pass with more context;
- it trains on the student's own trajectories, which avoids the train–test mismatch of offline distillation.

3. On-policy self-distillation: formulation

The same parameters play two roles. For a sampled trajectory $y \sim \pi_\theta(\cdot \mid x)$,

$$\text{student: } \pi_\theta(\cdot \mid x, y_{<t}), \qquad \text{teacher: } \pi_{\bar{\theta}}(\cdot \mid x, c, y_{<t}),$$

where c is available only during training and $\bar{\theta}$ is a stop-gradient (or moving-average) copy of θ. The loss is

$$\mathcal{L}(\theta) = \mathbb{E}_x \, \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \sum_t D\big(\pi_\theta(\cdot \mid x, y_{<t}) \,\big\|\, \pi_{\bar{\theta}}(\cdot \mid x, c, y_{<t})\big).$$

Choices include the divergence D (reverse KL, forward KL, JSD), full-vocabulary versus sampled-token estimates, and how the teacher copy is updated.

Privileged context c	**Examples**
reference solution	math reasoning (Zhao et al., 2026)
textual feedback, or a successful rollout	code, science, tool use (Hübotter et al., 2026)
demonstrations	continual learning (Shenfeld et al., 2026)
the model's own revision	binary-reward tasks (He et al., 2026)
skill descriptions	multi-turn agents (Wang et al., 2026a; Lu et al., 2026a)

3. On-policy self-distillation: known problems

- **Confidence the student cannot justify.** When the teacher sees a full solution, it reasons with little expressed uncertainty, and the student learns that style. Kim et al. (2026) report shorter outputs and accuracy drops of up to 40% on math reasoning, worse on unseen problems.
- **Teacher scored on prefixes it would not write.** The teacher is evaluated on the student's trajectories. In one agent study, more than half of the sampled tokens had a lower teacher than student log-probability (Lu et al., 2026a).
- **Long horizons.** In multi-turn agents, early mismatches compound across turns. Standalone self-distillation reached near-zero accuracy on search QA with a 3B model (Lu et al., 2026a); turn-level curricula are one response (Wang et al., 2026b).
- **Unclear source of gains.** In the same agent study, even randomly retrieved skills improved on RL alone, so it is not always clear how much the teacher's information matters.

See also Fu et al. (2026) for failure modes of on-policy distillation more generally.

3. A statistical reading of the self-distillation signal

Write $\Delta(y) = \log \pi(y \mid s, c) - \log \pi(y \mid s)$ for the same model with and without c.

The average gap on student samples is never positive.

$$\mathbb{E}_{y \sim \pi(\cdot \mid s)}[\Delta(y)] = -D_{\mathrm{KL}}\big(\pi(\cdot \mid s) \,\|\, \pi(\cdot \mid s, c)\big) \leq 0.$$

A negative average gap is therefore expected, and does not by itself show that the teacher is worse.

The gap is a pointwise information measure. If the model were a coherent joint distribution over (c, y), Bayes' rule would give

$$\Delta(y) = \log \pi(c \mid s, y) - \log \pi(c \mid s),$$

so self-distillation rewards tokens that make the privileged information more likely.

The student cannot condition on c. When c cannot be recovered from s, the best the student can do is match an average of the teachers $\pi(\cdot \mid s, c)$ over c. The next slide shows which average. This is one way to read the overconfidence result of Kim et al. (2026).

3. What a context-free student can learn

Let $c \sim p(c \mid s)$ and $p_c = \pi_{\mathrm{T}}(\cdot \mid s, c)$.

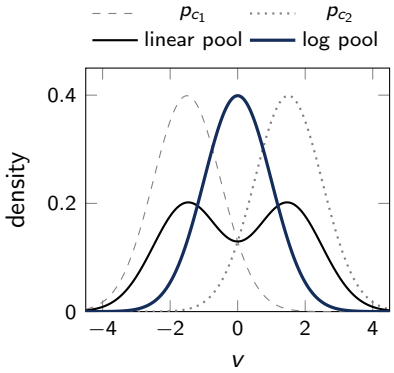
Proposition

$$\arg\min_q \mathbb{E}_c \, D_{\mathrm{KL}}(p_c \,\|\, q) = \mathbb{E}_c[p_c],$$

$$\arg\min_q \mathbb{E}_c \, D_{\mathrm{KL}}(q \,\|\, p_c) \propto \exp\left(\mathbb{E}_c \log p_c\right).$$

For the second,
$\mathbb{E}_c D_{\mathrm{KL}}(q \| p_c) = D_{\mathrm{KL}}(q \| g/Z) - \log Z$ with
$g = \exp(\mathbb{E}_c \log p_c)$.

Forward KL gives the *linear pool* (a mixture).
Reverse KL, the usual choice, gives the *logarithmic pool*, which puts little mass on anything one of the teachers rules out (Genest and Zidek, 1986).



Example with $p_{c_1} = \mathcal{N}(-1.5, 1)$ and $p_{c_2} = \mathcal{N}(1.5, 1)$.

3. Combining self-distillation with RL

In agent settings most recent work keeps an RL objective and adds the teacher signal.

Approach	**Idea**	**Examples**
Reweight the advantage	scale A by a function of the teacher-student gap	RLSD (Yang et al., 2026)
Auxiliary loss	$\mathcal{L}_{\mathrm{RL}} + \lambda\, \mathcal{L}_{\mathrm{distill}}$	Skill-SD (Wang et al., 2026a), SDAR (Lu et al., 2026a)
Schedules	decide when to distill	TCOD (Wang et al., 2026b), HDPO (Ding, 2026)
Token selection	distill only on some tokens	TIP (Xu et al., 2026), GATES (Stein et al., 2026)

Open questions from a statistical point of view. How much bias the added term introduces into the RL objective, and whether the teacher's reliability on a given token can be estimated instead of fixed by a hand-set gate.

3. What the distillation gradient is

Let $\Delta(v) = \log \pi_{\mathrm{T}}(v \mid s) - \log \pi_\theta(v \mid s)$. Treating the teacher as fixed,

$$\nabla_\theta D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s) \,\|\, \pi_{\mathrm{T}}(\cdot \mid s)\big) = \sum_v \nabla \pi_\theta(v) \Big[\log \frac{\pi_\theta(v)}{\pi_{\mathrm{T}}(v)} + 1 \Big] = -\mathbb{E}_{v \sim \pi_\theta} \big[\Delta(v) \, \nabla_\theta \log \pi_\theta(v) \big].$$

So reverse-KL distillation is a policy gradient with per-token reward Δ.

Some consequences:

- The sampled estimate $-\Delta(y_t)$ is unbiased for the KL, but differentiating it with y_t held fixed gives $\nabla_\theta \log \pi_\theta(y_t)$, which has mean zero. The value estimate and the gradient estimate are different objects.
- Δ is unbounded below, so a few tokens the teacher strongly disagrees with can dominate. Gating and clipping act like bounded influence functions in robust statistics.
- With privileged context, Δ is a noisy proxy for token quality, which connects to learning using privileged information (Vapnik and Vashist, 2009; Lopez-Paz et al., 2016).

4. Rewards beyond verifiers

Verifiable rewards (tests, exact match) are the default where available. For open-ended tasks, recent work scores trajectories with an LLM judge following a task-specific rubric (Gunjal et al., 2025).

The main concern is reward hacking: policies learn to exploit judge biases such as length or flattery.

Treat the judge as a noisy classifier of true success $R \in \{0, 1\}$, with sensitivity se and specificity sp.

Proposition

If se *and* sp *do not depend on the trajectory beyond* R*, then*

$$\mathbb{E}_\theta[\tilde{R}] = (\mathrm{se} + \mathrm{sp} - 1)\, J(\theta) + (1 - \mathrm{sp}).$$

So non-differential judge error only shrinks the gradient. Error that depends on features the policy controls can move the optimum, which is one way to describe reward hacking.

Open question. Can we detect differential judge error during training, for example with latent-class models over several judges (Dawid and Skene, 1979)?

5. Environments and task selection

Environments are a bottleneck. Recent work generates them automatically: Agent World Model builds about 1,000 code-based environments and reports that training on 10 overfits while 100 to 526 keeps improving (Wang et al., 2026). Similar efforts include ScaleEnv (Tu et al., 2026) and EnvCraft (Zeng et al., 2026).

Within a fixed pool, methods skip tasks with no signal (DAPO) or allocate rollouts by current ability (Yao et al., 2026).

Which tasks are informative? With $S_j = \partial_j \log p_\theta(\tau \mid x)$, Cauchy–Schwarz gives

$$|\partial_j p_x(\theta)| = |\mathrm{Cov}(R, S_j)| \le \sqrt{p_x(1-p_x)}\,\sqrt{\mathbb{E}S_j^2},$$

so tasks with p_x near $1/2$ carry the most signal.

How much does the number of environments matter? With m environments and n tasks each,

$$\mathrm{Var}(\bar{Y}) = \frac{\sigma^2_{\mathrm{env}}}{m} + \frac{\sigma^2_{\mathrm{task}}}{mn},$$

so generalization to new environments is limited by m.

6. Memory, skills, and context

Agents need information that does not fit in, or is not available in, the prompt.

Current work:

- trains the model to manage its own memory with RL (MemAgent; Yu et al., 2025);
- retrieves skill descriptions from a library during training (SkillRL; Xia et al., 2026);
- tries to move this knowledge into the weights so it is not needed at test time (Lu et al., 2026a, 2026b; Ye et al., 2026).

The gap can be large. In one ALFWorld experiment with a 3B model, training with skills in the prompt gave 80.5% success when skills were provided at test time and 60.2% when they were not (Lu et al., 2026a).

A statistical reading. The training inputs (x, c^+) and test inputs (x) come from different distributions, and when c^+ is not determined by x the student can only learn some average of the context-specific teachers. Topic 3 shows which average.

7. Evaluation

Many agent benchmarks report a single run per method with no standard errors. Recent papers measure run-to-run inconsistency directly (e.g. with intraclass correlation) and recommend clustered standard errors and paired comparisons (Miller, 2024).

For scale. On 128 tasks with accuracy around 0.8, the standard error is about 3.5 points, which is larger than many reported gains.

Reliability also matters. τ-bench reports pass^k, the probability of succeeding on all of k attempts (Yao et al., 2024).

With n attempts and c_i successes on task i, unbiased estimates are

$$\widehat{\text{pass@}k} = 1 - \binom{n - c_i}{k} \Big/ \binom{n}{k},$$

$$\widehat{\text{pass}^k} = \binom{c_i}{k} \Big/ \binom{n}{k}.$$

A benchmark score varies because of training seeds, environments, tasks, and repeated attempts. Only the last three are visible from one trained model, so seed variance is usually ignored.

Hierarchical models and paired designs seem well suited here, but are rarely used.

Common threads

Theme	Topics	Tools
Variance reduction	1, 2, 5	control variates, stratification, common random numbers
Distribution shift	1, 3, 6	importance sampling, covariate shift
Noisy supervision	3, 4	measurement error, latent-class models, robust estimation
Allocation and design	2, 5	optimal design, bandits
Uncertainty	7	variance components, bootstrap, paired tests

Questions I would like to work on

1. **Reliability of teachers and judges.** Model self-teachers and LLM judges as noisy raters of an unobserved quality, estimate their reliability token by token, and weight their signals accordingly.
2. **Bias of combined objectives.** Stationary points of $J(\theta) - \lambda \mathcal{L}_{\text{aux}}(\theta)$ are not maximizers of J. How large is the bias, and which schedules $\lambda_k \to 0$ remove it?
3. **Rollout allocation.** Choose how many rollouts to spend on each task, turn, and branch point to minimize gradient variance under a fixed budget.
4. **Inference for agent evaluation.** Hierarchical models over seeds, environments, tasks, and attempts, and confidence statements for reliability metrics like pass^k.

Summary

- Training agents with RL means estimating gradients from sparse, delayed, and sometimes noisy rewards over long trajectories.
- Current research tries to stabilize updates, assign credit more finely, add dense signals through (self-)distillation, broaden rewards, scale environments, and move knowledge into the weights.
- Each of these involves a bias-variance trade-off or a new source of measurement error, which is where I think statistical methods can help.

Thank you.

yuyaow@bu.edu

References (1/4)

Agarwal, R., et al. (2024). On-policy distillation of language models: Learning from self-generated mistakes. *ICLR*.

Cui, D., et al. (2025). The entropy mechanism of reinforcement learning for reasoning language models. arXiv:2505.22617.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *JRSS C*, 28(1).

Ding, K. (2026). HDPO: Hybrid distillation policy optimization via privileged self-distillation. arXiv:2603.23871.

Feng, L., et al. (2025). Group-in-group policy optimization for LLM agent training. arXiv:2505.10978.

Fu, Y., et al. (2026). Revisiting on-policy distillation: Empirical failure modes and simple fixes. arXiv:2603.25562.

Genest, C. and Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1).

Gu, Y., Dong, L., Wei, F. and Huang, M. (2024). MiniLLM: Knowledge distillation of large language models. *ICLR*.

Gunjal, A., et al. (2025). Rubrics as rewards: Reinforcement learning beyond verifiable domains. arXiv:2507.17746.

He, Y., et al. (2026). Self-distillation zero: Self-revision turns binary rewards into dense supervision. arXiv:2604.12002.

Hinton, G., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network. arXiv:1503.02531.

Hu, T., et al. (2026). SeeUPO: Sequence-level agentic-RL with convergence guarantees. arXiv:2602.06554.

References (2/4)

Hübotter, J., et al. (2026). Reinforcement learning via self-distillation. arXiv:2601.20802.

Kim, J., et al. (2026). Why does self-distillation (sometimes) degrade the reasoning capability of LLMs? arXiv:2603.24472.

Liu, Z., et al. (2025). Understanding R1-Zero-like training: A critical perspective. arXiv:2503.20783.

Lopez-Paz, D., Bottou, L., Schölkopf, B. and Vapnik, V. (2016). Unifying distillation and privileged information. *ICLR*.

Lu, K. and Thinking Machines Lab (2025). On-policy distillation. *Connectionism* (blog).

Lu, Z., et al. (2026a). Self-distilled agentic reinforcement learning. arXiv:2605.15155.

Lu, Z., et al. (2026b). Skill0: In-context agentic reinforcement learning for skill internalization. arXiv:2604.02268.

Miller, E. (2024). Adding error bars to evals: A statistical approach to language model evaluations. arXiv:2411.00640.

Ng, A. Y., Harada, D. and Russell, S. (1999). Policy invariance under reward transformations. *ICML*.

Shao, Z., et al. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300.

Shenfeld, I., Damani, M., Hübotter, J. and Agrawal, P. (2026). Self-distillation enables continual learning. arXiv:2601.19897.

Snell, C., Klein, D. and Zhong, R. (2022). Learning by distilling context. arXiv:2209.15189.

References (3/4)

Stein, A., Huang, F. and Goldstein, T. (2026). GATES: Self-distillation under privileged context with consensus gating. arXiv:2602.20574.

Tu, D., et al. (2026). ScaleEnv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training. arXiv:2602.06820.

Vapnik, V. and Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6).

Wang, H., et al. (2026a). Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. arXiv:2604.10674.

Wang, J., et al. (2026b). TCOD: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. arXiv:2604.24005.

Wang, Z., et al. (2026). Agent World Model: Infinity synthetic environments for agentic reinforcement learning. arXiv:2602.10090.

Wei, Q., et al. (2025). Reinforcing multi-turn reasoning in LLM agents via turn-level reward design. arXiv:2505.11821.

Xia, P., et al. (2026). SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. arXiv:2602.08234.

Xu, Y., et al. (2026). TIP: Token importance in on-policy distillation. arXiv:2604.14084.

Yang, A., et al. (2025). Qwen3 technical report. arXiv:2505.09388.

Yang, C., et al. (2026). Self-distilled RLVR. arXiv:2604.03128.

Yao, S., Shinn, N., Razavi, P. and Narasimhan, K. (2024). τ-bench: A benchmark for tool-agent-user interaction in real-world domains. arXiv:2406.12045.

References (4/4)

Yao, Z., et al. (2026). CoBa-RL: Capability-oriented budget allocation for reinforcement learning in LLMs. arXiv:2602.03048.

Ye, T., et al. (2026). On-policy context distillation for language models. arXiv:2602.12275.

Yu, H., et al. (2025). MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. arXiv:2507.02259.

Yu, Q., et al. (2025). DAPO: An open-source LLM reinforcement learning system at scale. arXiv:2503.14476.

Yue, Y., et al. (2025). Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? arXiv:2504.13837.

Zeng, Y., et al. (2026). EnvCraft: Synthesizing executable environments in agentic RL for claw-like agents. arXiv:2609.05576.

Zhang, G., et al. (2026). The landscape of agentic reinforcement learning for LLMs: A survey. *TMLR*. arXiv:2509.02547.

Zhao, S., et al. (2026). Self-distilled reasoner: On-policy self-distillation for large language models. arXiv:2601.18734.

Zheng, C., et al. (2025). Group sequence policy optimization. arXiv:2507.18071.

Zhou, Y., Zanette, A., Pan, J., Levine, S. and Kumar, A. (2024). ArCHer: Training language model agents via hierarchical multi-turn RL. *ICML*.

Credit assignment survey: *From reasoning to agentic: Credit assignment in reinforcement learning for large language models* (2026). arXiv:2604.09459.

Run-to-run inconsistency: *Stochasticity in agentic evaluations: Quantifying inconsistency with intraclass correlation* (2025). arXiv:2512.06710.

My specific research interest

LLM Agents for Spatio-Temporal Data

What the area is, how active it is, and where statistics fits

What "agent + spatio-temporal" means

Spatio-temporal data. Observations indexed by location and time, $Y(s, t)$: traffic sensors, GPS trajectories, weather grids, remote sensing, air quality, epidemics.

Classic tasks: forecasting, interpolation (kriging), anomaly detection, trajectory mining, simulation.

LLM agents. A language model that works in a loop: it plans a multi-step analysis, calls tools (databases, GIS operations, forecasting models, code), reads the results, and decides what to do next.

The output is an answer, a map, a forecast, or an action.

The combination. An agent that answers questions or makes decisions about the world *in space and time* by reasoning over spatio-temporal data and calling the right tools. Example: "Which districts will exceed the PM2.5 limit tomorrow, and how confident are we?"

Why agents, rather than a single model

Real tasks are workflows. A spatial analysis usually chains many steps: load layers, reproject, join, buffer, aggregate, fit a model, map the result. GeoAgentBench, for example, gives agents 117 GIS tools and 53 analysis tasks (Yu et al., 2026).

LLMs are weak at the computation. Text-trained models do not reliably encode temporal structure or do numerical work. A common design separates planning (the LLM) from computation (tools and specialized models).

Evidence is heterogeneous. Sensor series, maps, satellite images, and text reports often have to be combined. An agent can route each to an appropriate tool.

Conditions change. Spatio-temporal systems are non-stationary: new observations arrive, regimes shift, rare events matter. Decisions must be updated, which fits a loop better than a fixed pipeline.

Four lines of work

Role of the agent	What it does	Examples
Analyst	plans and executes spatial or time-series analyses with tools	GIS agents and benchmarks (Yu et al., 2026); time-series agents (survey, 2026)
Reasoner	a model trained to reason about spatio-temporal inputs	UrbanGPT (Li et al., 2024); UrbanLLaVA (Feng et al., 2025a); STReasoner (2026); Urban-R1 (2025)
Simulated person	generates human behaviour such as daily mobility	LLM agents as urban residents (Wang et al., 2024); multi-agent traffic simulation (ICLR 2026)
Decision maker	chooses actions in a spatial system	urban activity planning (Jiang et al., 2024); traffic signal control

The "reasoner" line increasingly borrows the RL recipe from math and code: verifiable rewards and GRPO-style training, now with spatial structure in the reward or the data (e.g. STReasoner, Urban-R1).

A formulation

As a sequential decision problem. A query x refers to a region $\mathcal{S}$ and time window $\mathcal{T}$. The agent observes a subset of the field

$$\mathcal{O} = \{\, Y(s_i, t_j) \,\} \subset \{ Y(s,t) : s \in \mathcal{S},\ t \in \mathcal{T} \},$$

and at turn k chooses an action a_k: query data, call a GIS operation, fit or run a model, or answer.

Trajectory $\tau = (a_1, o_2, \ldots, a_K)$, reward $R(x, \tau)$.

Where rewards come from:

- **Forecasts:** error against the realized $Y(s, t+h)$, available after the fact.
- **Analyses:** whether the executed workflow and its parameters match a reference (execution-based checking).
- **Questions:** exact match on locations, times, counts.
- **Simulation:** distance between simulated and observed mobility statistics.

So many spatio-temporal tasks have *verifiable* rewards, which makes them natural targets for the agent training methods developed for code and search.

Is the area active?

Signals of activity (2025–2026):

- **Surveys:** agentic AI for spatio-temporal data (Hashemi and Züfle, 2026); LLM agents for time series (2026).
- **Tutorials:** KDD 2025 (Liang et al., 2025); WWW 2026 (Zhang et al., 2026).
- **Workshops:** UrbComp at KDD 2026; STRL at IJCAI 2026 (fifth edition).
- **Benchmarks:** CityBench (Feng et al., 2025b); GeoAgentBench (Yu et al., 2026).
- **Trained models:** R1-style RL for urban and spatio-temporal reasoning (Urban-R1, 2025; STReasoner, 2026).

A realistic reading:

- Clearly growing, especially in data mining, urban computing, and GIScience venues (KDD, WWW, SIGSPATIAL).
- Smaller than core agent research on code, web, and search, which dominates NeurIPS/ICML agent papers.
- Much current work builds prompted pipelines. Trained agents, careful evaluation, and uncertainty quantification are less developed.
- This leaves room for methodological work, and spatial statistics is directly relevant.

Where statistics fits

Dependence. Nearby locations and times are correlated. Random train/test splits leak information; spatial or temporal blocking is needed (Roberts et al., 2017). Correlated tasks also mean fewer effectively independent training signals.

Uncertainty. Forecasts and interpolations need calibrated uncertainty. Agent outputs rarely provide it. Proper scoring rules can serve as rewards (Gneiting and Raftery, 2007).

Models as tools. Kriging, Gaussian processes, and hierarchical spatio-temporal models (Cressie and Wikle, 2011) can be tools the agent calls. Choosing among them is a model selection problem made in the loop.

Shift across places and times. A model trained on one city or season is applied to another. This is covariate shift with spatial structure, and it affects both the agent's policy and the tools it relies on.

Two quantitative points

Effective sample size. For an AR(1) series with lag-one correlation ρ, the mean of n observations has the variance of about

$$n_{\text{eff}} \approx n\,\frac{1-\rho}{1+\rho}$$

independent ones. With $\rho = 0.8$ and $n = 1000$, $n_{\text{eff}} \approx 111$.

The same issue applies to benchmark tasks drawn from neighbouring regions or days: the standard error of a reported score is larger than $\sqrt{p(1-p)/n}$ suggests.

A proper reward for probabilistic forecasts. The continuous ranked probability score

$$\mathrm{CRPS}(F, y) = \int \big(F(z) - \mathbf{1}\{y \le z\}\big)^2\, dz$$

is minimized in expectation by reporting the true predictive distribution. For $F = \mathcal{N}(\mu, \sigma^2)$ and $\omega = (y-\mu)/\sigma$,

$$\mathrm{CRPS} = \sigma\Big[\omega\big(2\Phi(\omega) - 1\big) + 2\varphi(\omega) - \tfrac{1}{\sqrt{\pi}}\Big]$$

(Gneiting et al., 2005). Using $R = -\mathrm{CRPS}$ rewards calibration, not only point accuracy.

Connecting to agent training

Hindsight as privileged information. In forecasting, the realized future $Y(s, t+h)$ is unknown at prediction time but known at training time. This fits the on-policy self-distillation setup: a teacher that sees the future, a student that does not.

Which divergence? Many futures are consistent with the student's information. Minimizing forward KL to future-conditioned teachers gives their mixture, a proper predictive distribution. Reverse KL, the usual choice, gives a normalized geometric mean, which is too concentrated for forecasting.

Other open questions:

- credit assignment over long tool chains in GIS workflows;
- evaluation with spatial and temporal blocking;
- transfer from simulated cities to real ones.

The divergence point is my own suggestion, based on the pooling result for context-free students.

Summary

- Agents for spatio-temporal data combine LLM planning with tools and models that handle space and time: as analysts, reasoners, simulated people, and decision makers.
- The area is growing, with 2026 surveys, tutorials, workshops, and benchmarks, mostly in data mining and urban computing. It is smaller and less crowded than general agent research.
- Many tasks have verifiable rewards, so current agent training methods apply. The spatio-temporal setting adds dependence, uncertainty, and shift, which are statistical questions.

Thank you.

yuyaow@bu.edu

Spatio-temporal references (1/2)

Cressie, N. and Wikle, C. K. (2011). *Statistics for Spatio-Temporal Data*. Wiley.

Feng, J., et al. (2025a). UrbanLLaVA: A multi-modal large language model for urban intelligence with spatial reasoning and understanding. arXiv:2506.23219.

Feng, J., et al. (2025b). CityBench: Evaluating the capabilities of large language models for urban tasks. *KDD*.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *JASA*, 102(477).

Gneiting, T., Raftery, A. E., Westveld, A. H. and Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133(5).

Hashemi, M. and Züfle, A. (2026). A comprehensive survey of agentic AI for spatio-temporal data. *Preprints* 2026012236.

Jiang, Y., et al. (2024). UrbanLLM: Autonomous urban activity planning and management with large language models. arXiv:2406.12360.

Li, Z., et al. (2024). UrbanGPT: Spatio-temporal large language models. *KDD*.

Liang, Y., et al. (2025). Foundation models for spatio-temporal data science: A tutorial and survey. *KDD*. arXiv:2503.13502.

Roberts, D. R., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8).

Spatio-temporal references (2/2)

Wang, J., et al. (2024). Large language models as urban residents: An LLM agent framework for personal mobility generation. *NeurIPS*.

Yu, B., et al. (2026). GeoAgentBench: A dynamic execution benchmark for tool-augmented agents in spatial analysis. arXiv:2604.13888.

Zhang, Z., Miao, H., Liang, Y., Zhao, Y. and King, I. (2026). LLM-enhanced web-centric spatio-temporal intelligence: Methods, applications, and frontier research. *WWW Companion*.

LLM agents for time-series: A survey (2026). arXiv:2608.26226.

STReasoner: Empowering LLMs for spatio-temporal reasoning in time series via spatial-aware reinforcement learning (2026). arXiv:2601.03248.

Urban-R1: Reinforced MLLMs mitigate geospatial biases for urban general intelligence (2025). arXiv:2510.16555.

Advancing multi-agent traffic simulation via R1-style reinforcement fine-tuning (2026). *ICLR*.

Backup: derivations

Judge error. With $\mathbb{P}(\tilde{R}=1 \mid R=1)=\mathrm{se}$ and $\mathbb{P}(\tilde{R}=1 \mid R=0)=1-\mathrm{sp}$,

$$\mathbb{E}_\theta[\tilde{R}] = \sum_\tau p_\theta(\tau)\big[\mathrm{se}\, R(\tau) + (1-\mathrm{sp})(1-R(\tau))\big] = (\mathrm{se}+\mathrm{sp}-1)J(\theta) + (1-\mathrm{sp}).$$

If se or sp depend on τ, extra terms appear that are not proportional to ∇J.

KL estimators. For $v \sim \pi_\theta$ and $\rho = \pi_{\mathrm{T}}(v)/\pi_\theta(v)$, both $k_1 = -\log\rho$ and $k_3 = (\rho-1)-\log\rho$ are unbiased for $D_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{T}})$, since $\mathbb{E}[\rho]=1$. Also $k_3 \geq 0$ because $\log\rho \leq \rho-1$.

Importance weights. If $\log w \sim \mathcal{N}(\mu, s^2)$ with $\mathbb{E}w = 1$, then $\mu = -s^2/2$ and $\mathrm{Var}(w) = e^{s^2}-1$.