Structure-Driven Agentic Reasoning for Large Language Models

Siheng Xiong

Advisor: Faramarz Fekri

School of Electrical and Computer Engineering
Georgia Institute of Technology

November 2026

Georgia Tech

Agenda

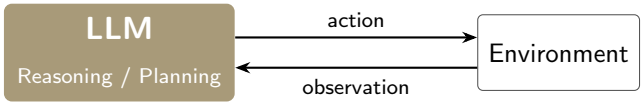
- Motivation and research goal

Technical contributions

- TC 1: Learning temporal reasoning through explicit graph structure
- TC 2: Planning and verifying reasoning through entailment graphs
- TC 3: Learning long-horizon reasoning through multi-level policies
- TC 4: Adaptive information control for search-augmented reasoning
- TC 5: Self-improvement through evolving skill populations

- Conclusion and future work

LLMs are becoming agentic reasoners



The model reasons step by step, interacts with the environment, and uses feedback to decide what to do next.

Within a step	Which inference to make next? Is it supported by previous steps?
Within a trajectory	Which subgoal to pursue next? How to execute it?
Across tasks	What to learn from previous experience? How to reuse it?

Georgia Tech

Unstructured reasoning leaves these decisions implicit

✗ Unchecked dependencies

The model generates reasoning steps, but it is not clear which earlier steps they depend on, or whether those dependencies are valid.

✗ Long-horizon drift

The model handles planning and execution in a single token stream, so it may lose track of high-level goals.

✗ Greedy skill refinement

The model repeatedly refines the same skill, which may lead to overfitting or ineffective optimization.

The model has to make these decisions anyway. The key question is how to structure these decisions.

Georgia Tech

Research question

How can explicit structure organize and improve agentic reasoning?

During learning

Structure helps us supervise, compare, and revise reasoning.

During inference

Structure guides reasoning and verification.

Georgia Tech

Structure across different scopes







	Graphs		Hierarchies		Populations
	TG-LLM [ACL'24]	**SWAP** [ACL'25]	**MLR** [ICLR'26]	**DeepControl** [EMNLP'26]	**DIVE** [preprint'26]
Structure	temporal graph	entailment graph	plan–execution hierarchy	evidence hierarchy	evolving skill populations
Organizes	task information	reasoning steps	planning and execution	information acquisition	experience across tasks
Improves via	graph-driven supervision	inference-time planning	hierarchical policy learning	utility-guided policy learning	multi-skill optimization

Georgia Tech

Outline

Planning and verifying reasoning through entailment graphs

SWAP: Deliberate Reasoning in Language Models as Structure-Aware Planning with an Accurate World Model
Xiong, Payani, Yang, Fekri. ACL 2025

Learning long-horizon reasoning through multi-level policies

Self-improvement through evolving skill populations

Additional contributions

Conclusion and future work

Georgia Tech

Multi-step reasoning with unstructured CoT

Logic (FOLIO; Han et al., 2022)

All people who regularly drink coffee are dependent on caffeine. People regularly drink coffee, or they don't want to be addicted to caffeine, or both. No one who doesn't want to be addicted to caffeine is unaware that caffeine is a drug. Rina is either a student who is unaware that caffeine is a drug, or she is not a student and she is aware that caffeine is a drug. Rina is either a student who is dependent on caffeine, or she is not a student and not dependent on caffeine. *Does it follow that Rina doesn't want to be addicted to caffeine or is unaware that caffeine is a drug?*

Let's think step by step. Rina is either a student or not. If she is a student, she is unaware that caffeine is a drug, so the conclusion holds. If she is not a student, she is not dependent on caffeine. ... The premises say nothing about whether someone who is not dependent on caffeine wants to be addicted to it. ... Therefore, the answer is Uncertain.

Answer: Uncertain ✗ (correct: True)

Math (MATH500; Hendrycks et al., 2021)

Solve for a: $\sqrt{4+\sqrt{16+16a}}+\sqrt{1+\sqrt{1+a}}=6$.

Let's solve this step by step. First, notice that $16+16a = 16(1+a)$, so $\sqrt{16+16a} = 16\sqrt{1+a}$. Let $u = \sqrt{1+a}$. Then the equation becomes $\sqrt{4+16u}+\sqrt{1+u}=6$. To get rid of the square roots, we can isolate one of them: $\sqrt{4+16u} = 6-\sqrt{1+u}$. Squaring both sides gives $4+16u = 37+u-12\sqrt{1+u}$, so $15u-33 = -12\sqrt{1+u}$. Squaring again would give a quadratic, but it may introduce extraneous roots, so let's instead try some values. If $u=1$: $\sqrt{20}+\sqrt{2} \approx 4.47+1.41 = 5.88$, slightly below 6. If $u=1.1$: $\sqrt{21.6}+\sqrt{2.1} \approx 4.65+1.45 = 6.10$, slightly above 6. So u is between 1 and 1.1; trying $u=1.05$ gives about 5.99, so $u \approx 1.054$. Then $1+a = u^2 \approx 1.111$, which gives $a \approx 0.11$.

Answer: $a \approx 0.11$ ✗ (correct: $a = 8$)

Georgia Tech

Limitations of unstructured CoT

✗ Unchecked dependencies

The model does not explicitly track which earlier statements each reasoning step depends on, so errors may propagate unnoticed.

✗ Ineffective planning

Without explicit structure, it is difficult to compare different reasoning paths and select the best one.

SWAP makes dependencies explicit through an entailment graph.

Georgia Tech

Related work

Line of work	What it leaves open
Sampling and voting Self-consistency (Wang et al., 2023)	More reasoning paths, but only final answers are aggregated; individual steps are neither checked nor compared.
Search over reasoning states Tree of Thoughts (Yao et al., 2023) RAP (Hao et al., 2023)	Explores alternative states, but the states are unstructured text, so they are hard to compare.

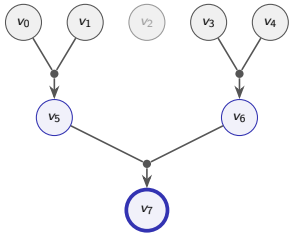
Georgia Tech

Multi-step reasoning as entailment-graph transitions

State $\mathcal{G}_t$: an entailment graph

Action a_t: a proposed update

Transition: applying a_t to $\mathcal{G}_t$ gives $\mathcal{G}_{t+1}$



Entailment graph $\mathcal{G} = (V, E)$

- nodes: statements (evidence, assumptions, lemmas)
- hyperedges: premises $\rightarrow$ conclusions

Every new statement records its premises, so each step can be checked against what it builds on.

Georgia Tech

Multi-step reasoning as entailment-graph transitions

Example (FOLIO): Does the conclusion follow from the premises?

Conclusion: Rina doesn't want to be addicted to caffeine or is unaware that caffeine is a drug.

Initial graph $\mathcal{G}_0$ (premises):

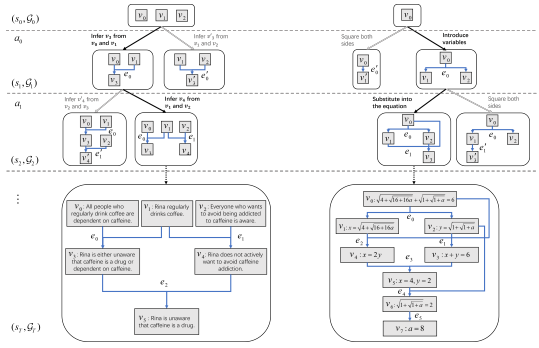
- v_0: All people who regularly drink coffee are dependent on caffeine.
- v_1: People regularly drink coffee, or they don't want to be addicted to caffeine, or both.
- v_2: No one who doesn't want to be addicted to caffeine is unaware that caffeine is a drug.
- v_3: Rina is either a student who is unaware that caffeine is a drug, or she is not a student and she is aware that caffeine is a drug.
- v_4: Rina is either a student who is dependent on caffeine, or she is not a student and not dependent on caffeine.

t	**Action** a_t	**Added to** $\mathcal{G}_{t+1}$	
0	combine v_0, v_1	v_5: Rina is dependent on caffeine, or she doesn't want to be addicted to caffeine.	$\{v_0, v_1\} \to v_5$
1	combine v_3, v_4	v_6: If Rina is dependent on caffeine, she is unaware that caffeine is a drug.	$\{v_3, v_4\} \to v_6$
2	combine v_5, v_6	v_7: Rina doesn't want to be addicted to caffeine, or she is unaware that caffeine is a drug. = conclusion ✓	$\{v_5, v_6\} \to v_7$

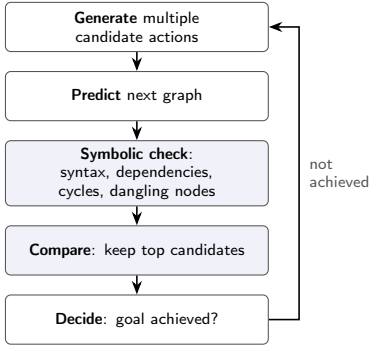
Georgia Tech

Structure-aware planning (SWAP)

Idea: planning over entailment graphs.

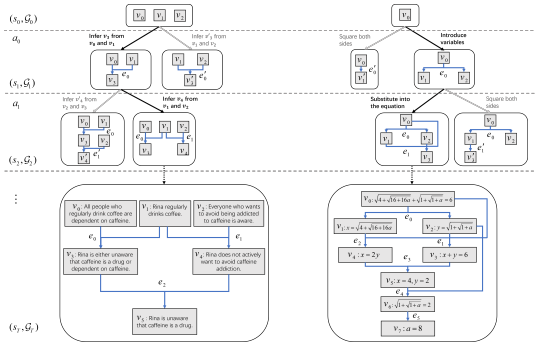


Left: FOLIO Right: MATH500

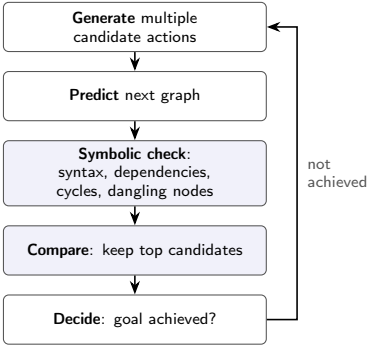


Structure-aware planning (SWAP)

Idea: planning over entailment graphs.



Left: FOLIO Right: MATH500



Planning only helps if candidates are meaningfully different and selection can tell them apart.

Georgia Tech

Challenge 1: generation diversity

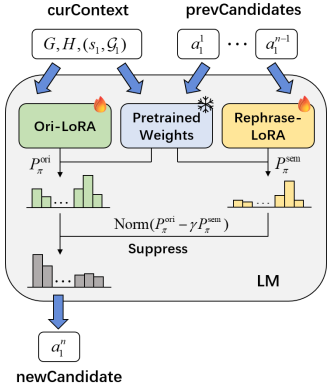
Observation: sampled candidates often differ only in wording.

Diversity-based Modeling (DM)

Subtract the paraphrase distribution of previous candidates from the original policy:

$$P_\pi \propto \max\Big(P_\pi^{\text{ori}}(a_{t,l}^n \mid \cdot) - \gamma_l\, P_\pi^{\text{sem}}(a_{t,l}^n \mid a_t^{1..n-1}, \cdot),\, 0\Big)$$

- P_π^{sem}: trained on semantically equivalent action pairs
- $\gamma_l = \gamma_0 \alpha^l$: push for a new direction early, then let the original policy take over



Georgia Tech

Challenge 2: discrimination accuracy

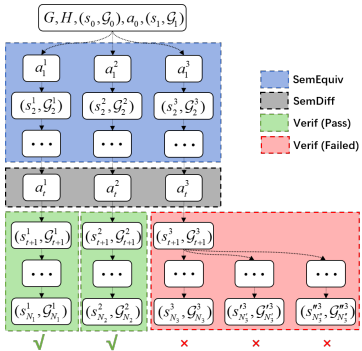
Observation: scoring candidates individually may miss important differences.

Contrastive Ranking (CR)

The discriminator sees all candidates side by side, explains their differences, then picks one:

$$E,\ a_t^{\text{best}} \sim P_d\big(\cdot \mid G, H, s_t, \mathcal{G}_t, \{a_t^j, s_{t+1}^j, \mathcal{G}_{t+1}^j\}_{j=1}^K\big)$$

Automatically construct training data: branch from correct trajectories and pair correct and incorrect continuations.



Georgia Tech

Challenge 2: discrimination accuracy

Example: $p(x) = 2x + 3$, $q(x) = x^2 - x + 4$. Find the coefficient of x^2 in $p(x)q(x)$.
Current graph: v_0: $p(x) = 2x + 3$; v_1: $q(x) = x^2 - x + 4$

Option 1

a^1: multiply the leading terms
v_2: $2x \cdot x^2 = 2x^3$,
so the coefficient is 2

✗

Option 2

a^2: collect the products that give x^2
v_2': $2x \cdot (-x) + 3 \cdot x^2 = x^2$,
so the coefficient is 1

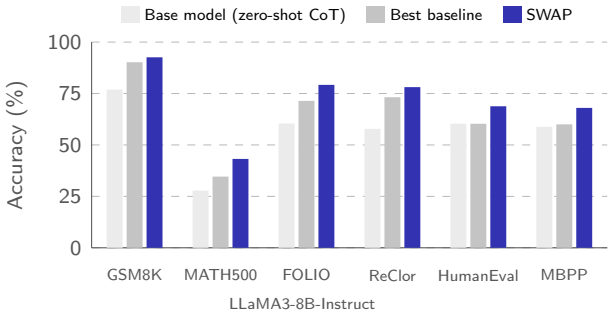
✓

Comparison: Both actions target a single coefficient, but the leading terms give the x^3 term, not x^2. Option 2 collects exactly the products of degree 2.

Conclusion: Option 2

Georgia Tech

Consistent gains across benchmarks and backbones



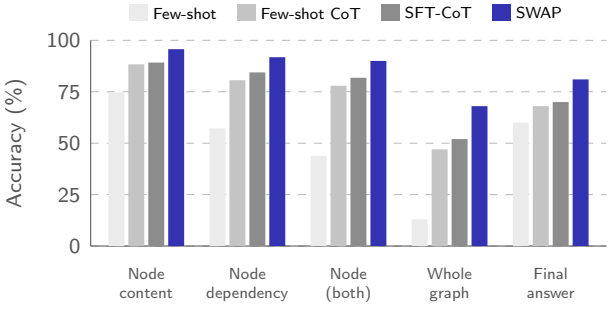
Average accuracy over six benchmarks

	Base	Best baseline	**SWAP**
LLaMA3-8B	57.0	65.0	**71.6**
Mistral-7B	37.2	52.6	**59.0**

Best baseline per benchmark among CoT, SFT-CoT, SC@32, ToT, RAP, PRM (PRM800K / Math-Shepherd); equal 32-rollout budget for search methods.

Georgia Tech

Process quality and ablation



Graph quality on FOLIO, LLaMA3-8B-Instruct.
Node (both): content and dependencies correct; whole graph: every node correct.

SWAP improves not only the final answers, but also the reasoning process itself.

Ablation (avg. over six benchmarks)

SWAP	**71.6**
w/o structure (inference)	69.7
w/o structure (training)	68.4
w/o CR (individual scoring)	67.5
w/o DM (independent sampling)	67.2
w/o planning (single pass)	62.0

LLaMA3-8B-Instruct

Georgia Tech

Takeaways

1. Making dependencies explicit allows us to check and compare reasoning steps
2. Planning is only as good as its candidates and its judge
3. Improving the reasoning process is as important as improving the final answer

Next: long-horizon reasoning needs multi-level policies.

Georgia Tech

Outline

Planning and verifying reasoning through entailment graphs

Learning long-horizon reasoning through multi-level policies

MLR: Enhancing Language Model Reasoning with Structured Multi-Level Modeling
Xiong, Payani, Fekri. ICLR 2026

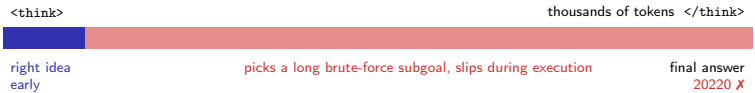
Self-improvement through evolving skill populations

Additional contributions

Conclusion and future work

DeepSeek-R1 on MATH500

Problem: What is the least positive integer multiple of 30 that can be written with only the digits 0 and 2?



early	The number must end in 0, and its digit sum must be a multiple of 3. With three 2s, the sum is 6, so 2220 works. ✓
then	To be sure it's the smallest, let me list every number made of 0s and 2s in increasing order and test each one.
later	20 ✗, 22 ✗, 200 ✗, 202 ✗, 220 ✗, 222 ✗, 2000 ✗, 2002 ✗, 2020 ✗, 2022 ✗, 2200 ✗, 2202 ✗, 2222 ✗ (2220 skipped), . . .
later	. . . 20220: digit sum 6, ends in 0 ✓
end	So the answer is 20220. (correct: 2220)

Georgia Tech

Limitations of long-horizon single-policy reasoning

✗ Long-horizon drift

One policy plans and executes in a single token stream. Over long traces, high-level decisions get buried in low-level details.

Problem: Least multiple of 30 using only the digits 0 and 2?

What a high-level policy would decide

(1) Conditions: ends in 0, digit sum a multiple of 3

(2) Smallest such number: 2220

(3) Verify directly: $2220/30 = 74$

(4) State the answer

The single-policy model gets distracted and produces the wrong answer

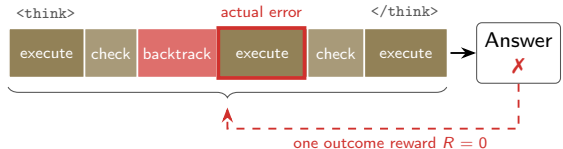
- ✓ Must end in 0; digit sum must be a multiple of 3.
- ✓ Three 2s give sum 6, so 2220 works.
- ✗ To be sure, let me list every number made of 0s and 2s...
- ✗ 20, 22, 200, ..., 2202, 2222 (2220 skipped), ...
- ✗ So the answer is 20220. (correct: 2220)

Georgia Tech

Limitations of long-horizon single-policy reasoning

✗ Credit assignment

A final-answer reward scores the whole trajectory. It does not tell which intermediate decision, a subgoal or its execution, made the difference.



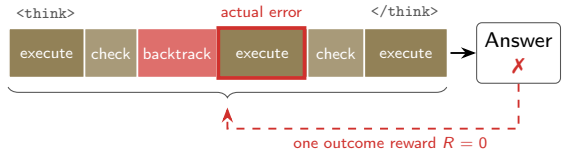
Which part of the trace was actually wrong?

Georgia Tech

Limitations of long-horizon single-policy reasoning

✗ Credit assignment

A final-answer reward scores the whole trajectory. It does not tell which intermediate decision, a subgoal or its execution, made the difference.



Which part of the trace was actually wrong?

Separate high-level planning from low-level execution,
and train both levels with process supervision.

Georgia Tech

Related work

Line of work	What it leaves open
Plan then solve Plan-and-Solve (Wang et al., 2023) ReWOO (Xu et al., 2023)	The plan is written once up front and does not adapt to what execution finds.
Process supervision PRM800K (Lightman et al., 2024) Math-Shepherd (Wang et al., 2023)	Step-level signals, but for a single flat policy: no separate signal for high-level planning.

Georgia Tech

Multi-level reasoning (MLR)

A trajectory of M subtasks: subgoals $d^{(m)}$ (what) and executions $c^{(m)}$ (how); a summary $c'^{(m)}$ compresses each completed subtask.

$$\text{High level} \quad p_\theta^H(d) = \prod_{m=1}^{M} p_\theta^H\left(d^{(m)} \mid d^{(1:m-1)},\, c'^{(1:m-1)}\right)$$

$$\text{Low level} \quad p_\theta^L(c) = \prod_{m=1}^{M} p_\theta^L\left(c^{(m)} \mid d^{(1:m)},\, c^{(1:m-1)}\right)$$

This allows the planner to maintain a high-level view without being overwhelmed by low-level details.

Georgia Tech

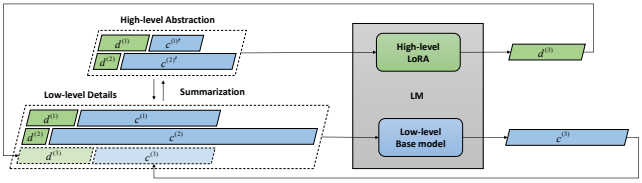
Example: MLR on MATH500

Problem: The set of points (x, y, z) that satisfy $2x = 3y = -z$ is a line; the set that satisfies $6x = -y = -4z$ is another line. Find the angle between these lines, in degrees.

	What the planner reads (284 tokens) subgoal $d^{(m)}$ + summary $c'^{(m)}$	**What the executor writes** (2,477 tokens) execution $c^{(m)}$, length in tokens
1	Understanding the problem The angle is determined by the direction vectors.	112
2	Finding direction vectors Direction vector of the first line is $(3, 2, -6)$.	509
3	Finding the second direction vector Direction vector is $(2, -12, -3)$.	301
4	Calculating the angle using the dot product formula Dot product is zero, so the vectors are perpendicular.	576
5	Verifying the result using another method Their direction vectors are perpendicular.	969
6	Extract the final answer Thus, the final answer is 90°.	10

Georgia Tech

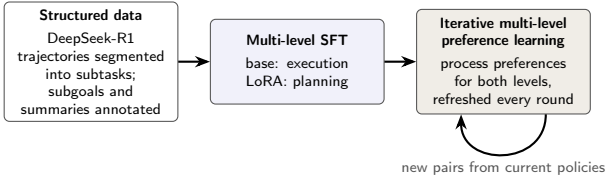
Architecture and training



Low level: backbone only, for execution

High level: backbone + LoRA adapter, for planning

One backbone for both levels: only a lightweight adapter is added.



Low level: preference pairs update *only* the backbone

High level: preference pairs update *only* the adapter

Mini-batches from the two levels are interleaved.

Georgia Tech

Process preference data

Problem: What is the area of a triangle with vertices $A(1,1)$, $B(6,1)$, $C(3,7)$?

Shared prefix: subtasks 1–3 (understand the problem, recall the formula, substitute the coordinates)

High-level pair (trains the adapter)

✓ **Chosen** d_+
Perform the calculations
$\Rightarrow$ area $= 15$

✗ **Rejected** d_-
Take the absolute value of each term, then add
$\Rightarrow$ area $= 21$

Low-level pair (trains the base model) current subgoal: Perform the calculations

✓ **Chosen** c_+
$1(1-7) = -6,\ \ 6(7-1) = 36,\ \ 3(1-1) = 0$
$\Rightarrow$ area $= 15$

✗ **Rejected** c_-
$1(1-7) = -6,\ \ 6(7-1) = 36,\ \ 3(1-1) = 3$
$\Rightarrow$ area $= 16.5$

Each pair shares the same prefix; the chosen continuation is the one more likely to reach the correct answer.

Georgia Tech

Process preferences via survivability

Idea: evaluate each intermediate decision by how likely it is to lead to a correct final answer.

Survival probability of a prefix under the target policy:

$$g(x^{(m)}) = P_{p_\theta}\left(R = 1 \mid x^{(m)}\right)$$

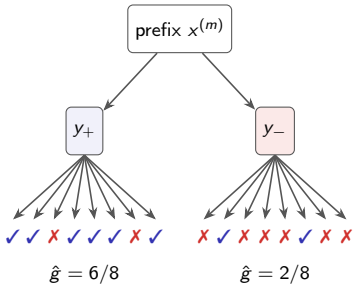
Utility of a candidate step $y^{(m)}$:

$$U(y^{(m)}) = \log \tilde{g}(x^{(m)}, y^{(m)}) - \log \tilde{g}(x^{(m)})$$

Keep a pair (y_+, y_-) when

$$U(y_+) - U(y_-) > \delta$$

Same construction for subgoals and for executions.



$K = 8$ rollouts per candidate

Georgia Tech

Cheap rollouts, reweighted for the target

Rollouts come from a small model; each one is reweighted by how likely the target model is to produce it, using Twisted Sequential Monte Carlo (TSMC).

Particle k: sample $y_k^{(m)} \sim p_{\text{roll}}(\cdot \mid x_k^{(m-1)})$, then:

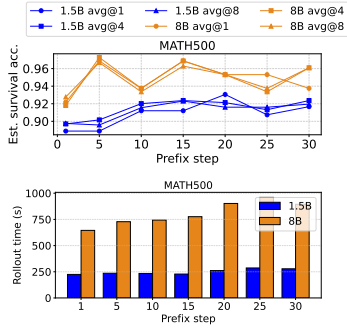
$$W_k^{(m)} = W_k^{(m-1)}\, \tilde{w}_k^{(m)}$$

$$\tilde{w}_k^{(m)} = G_m\big(x_k^{(m)}\big)\, \frac{p_\theta\big(x_k^{(m)} \mid x_k^{(m-1)}\big)}{p_{\text{roll}}\big(x_k^{(m)} \mid x_k^{(m-1)}\big)}\, \frac{\phi_m\big(x_k^{(m)}\big)}{\phi_{m-1}\big(x_k^{(m-1)}\big)}$$

$$\hat{g}_K\big(x^{(m)}\big) = \frac{1}{K}\sum_{k=1}^{K} W_k^{(M_k)}$$

G_M: final-answer correctness ($G_m = 1$ for $m < M$); p_θ: target model; p_{roll}: small rollout model; ϕ_m: survivability critic.

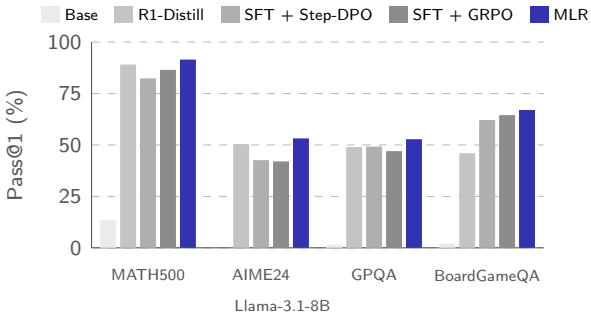
Comparable label accuracy at much lower runtime.



Survival-label accuracy (top) and rollout time on one A100 (bottom): R1-Distill-Qwen-1.5B vs. R1-Distill-Llama-8B. AIME24 and GPQA use the target model for rollouts.

Georgia Tech

MLR improves across benchmarks and backbones



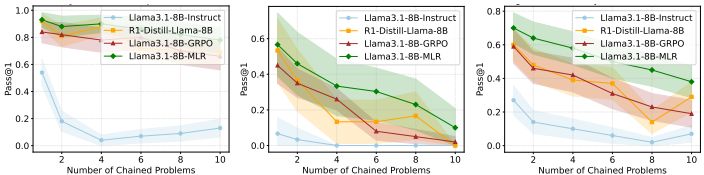
Average Pass@1 over four benchmarks

	Qwen2.5 1.5B	Qwen2.5-Math-7B	Llama-3.1 8B
Base	21.5	26.9	4.3
R1-Distill	47.7	60.0	58.6
SFT + Step-DPO	47.8	60.3	59.1
SFT + GRPO	48.4	59.9	60.0
MLR	**54.2**	**66.2**	**66.1**

MLR uses only 10% of the SFT data and 5% of the preference data compared with R1-Distill.

Georgia Tech

Long-horizon robustness and ablation



Llama-3.1-8B; accuracy vs. number of concatenated problems in one context

Trained only on single questions, MLR degrades more slowly as problems are chained.

Ablation (avg. Pass@1, Llama-3.1-8B)

MLR: multi-level + process pref.	**66.1**
Single policy + process pref.	59.1
Multi-level + outcome pref.	55.3
Single policy + outcome pref.	51.6
SFT only	42.2

Each choice helps more when the other is present: hierarchy $+3.7 \rightarrow +7.0$, process preferences $+7.5 \rightarrow +10.8$.

Georgia Tech

Takeaways

1. Separating planning from execution helps with long-horizon reasoning
2. Supervising both levels with process preferences improves reasoning performance
3. Process supervision can be made reliable and cost-efficient

Next: self-improvement needs evolving skill populations.

Georgia Tech

Outline

Planning and verifying reasoning through entailment graphs

Learning long-horizon reasoning through multi-level policies

Self-improvement through evolving skill populations

DIVE: Unlocking Self-Improvement in Frozen Language Models Through Diversity-Driven Skill Evolution
Xiong, Payani, Gungordu, Fekri. arXiv 2026

Additional contributions

Conclusion and future work

Georgia Tech

Self-improvement via skills

A **skill** s: a compact natural-language artifact with reasoning procedures, verification strategies, recurring failure patterns, and output constraints.

A frozen model solves tasks and a verifier labels its attempts; skill optimization turns that experience into reusable skills.

$$V_{\mathcal{T}}(s) = \mathbb{E}_{(x,y)\sim\mathcal{T}}\left[r(x, y, f_\theta(x; s))\right] \quad \text{expected performance with skill } s$$

$$\mathcal{E}(D; s) = \{(x, f_\theta(x; s), r)\}_{(x,y)\in D} \quad \text{verifier-labeled experience}$$

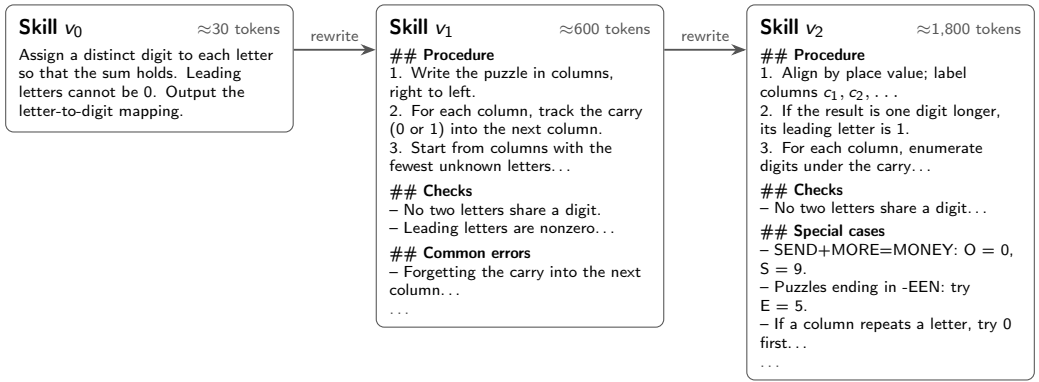
$$s' = \textsc{UpdateSkill}_{f_\theta}\left(s, \mathcal{E}(D; s)\right) \quad \text{the same frozen model revises the skill}$$

Learning happens in the text, not the weights:
skills can be inspected, edited, and transferred across models.

Georgia Tech

Greedy single-skill refinement

Task: Cryptarithm puzzles, e.g., SEND + MORE = MONEY (answer: $9567 + 1085 = 10652$); one skill, revised from verifier feedback.



Georgia Tech

Limitations of greedy single-skill refinement

✗ Overfitting

A complex dataset contains many patterns. A single skill that covers them all becomes long and complicated, and easily overfits the training data.

✗ Ineffective optimization

Greedy refinement always builds on the latest version, which is not always the best. Also, a skill can be improved in many ways (rewriting, merging, simplifying), so no single operator covers them all.

DIVE gives skill optimization a structure: a revision graph that can branch from any earlier version, and multiple populations evolving in parallel.

Georgia Tech

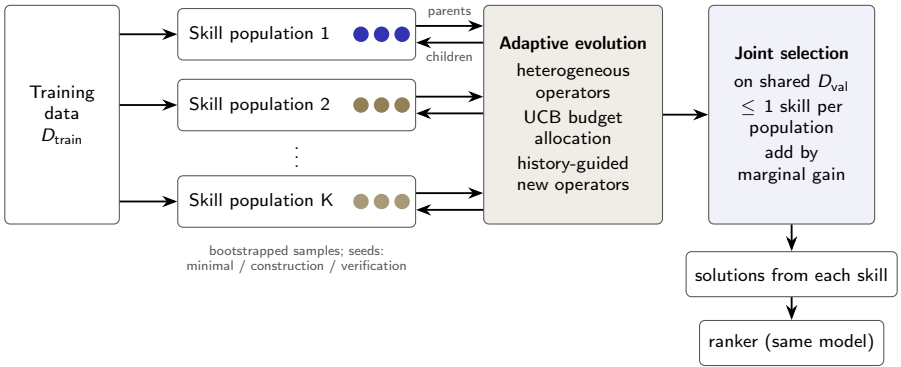
Related work

Line of work	What it leaves open
Learning from experience Self-Refine (Madaan et al., 2023) Reflexion (Shinn et al., 2023) ExpeL (Zhao et al., 2024)	Feedback is reused as reflections or insights, but without systematic population-based optimization.
Text-based optimization MIPROv2 (Opsahl-Ong et al., 2024) GEPA (Agrawal et al., 2025) SkillOpt (Yang et al., 2026)	Guidance is optimized as text, but the final result is a single prompt or skill. They do not consider adaptive operators.

Georgia Tech

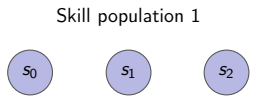
Diversity-driven skill evolution (DIVE)

DIVE uses adaptive evolution with multiple skill populations to increase diversity in skill evolution, and then selects jointly to benefit from that.



Georgia Tech

Each population grows its own revision graph

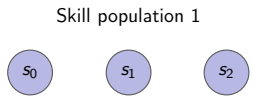


Start: three seed skills

s_0: minimal
s_1: construction
s_2: verification

Georgia Tech

Each population grows its own revision graph

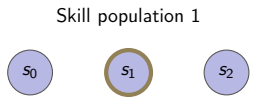


Iteration 1

1. Choose an operator (UCB): Reflective repair

Georgia Tech

Each population grows its own revision graph

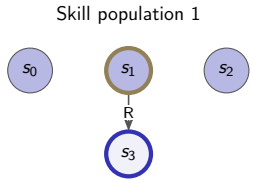


Iteration 1

1. Choose an operator (UCB): Reflective repair
2. Choose parent(s): s_1 (a strong skill)

Georgia Tech

Each population grows its own revision graph

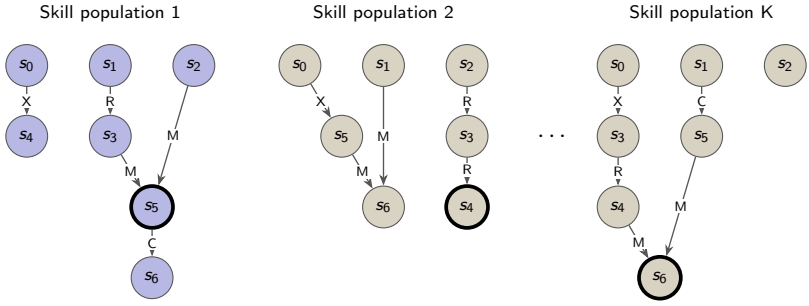


Iteration 1

1. Choose an operator (UCB): Reflective repair
2. Choose parent(s): s_1 (a strong skill)
3. Generate the child: s_3

Georgia Tech

Each population grows its own revision graph



s_0, s_1, s_2: seed skills **R**: reflective repair **X**: exploration **C**: compression **M**: recombination (two parents)

Parents are kept, so any earlier version can be revised again. **Bold**: each population's pick for joint selection.

Georgia Tech

Heterogeneous operators with adaptive allocation

A skill can be improved in many ways, and no single operator fits every case, so DIVE uses a portfolio of operators, from local fixes to new strategies.

Reflective repair
fix recurring errors, keep useful guidance
parent: a strong skill

Exploration
propose a substantially different strategy
parent: a weaker skill

Compression
consolidate an overly long skill
parent: effective but long

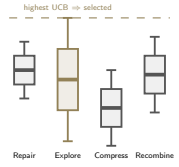
Recombination
merge complementary procedures, resolve conflicts
parents: strong + complementary

At step t_{new} the model reads the evolution history and writes new operators.

Reward: improvement over the best parent,
$R_t = U(s'_t) - \max_{s \in P_t} U(s)$

Operator choice (UCB): $a_t = \arg\max_{a \in \mathcal{A}} \left[\hat{\mu}_a(t) + \beta\sqrt{\log t / N_a(t)}\right]$

- Parent-relative credit puts single-parent and multi-parent operators on a common scale
- UCB gives more chances to promising operators while still exploring others



mean reward; top whisker: UCB

Georgia Tech

Select skills that work well together

$$\max_{S} \ \hat{V}_{D_{\text{val}}}(S) \quad \text{s.t. } |S| \le M, \ \ |S \cap S_B^{(k)}| \le 1 \ \forall k$$

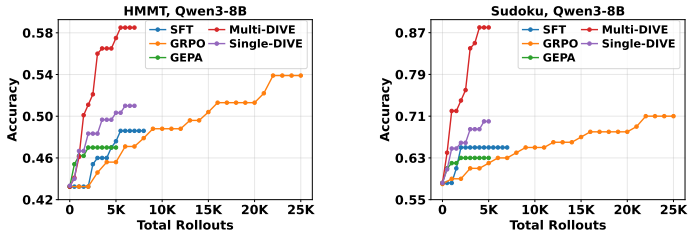
- $\hat{V}$ scores the *whole* pipeline: every skill generates a solution, the ranker picks one
- At most one skill per population
- Selection: start from the best single skill, then add the one with the largest positive marginal gain

	s_1	s_2	s_3	s_4
inst. 1	✓	✓	✗	✓
inst. 2	✓	✓	✓	✗
inst. 3	✓	✓	✗	✗
inst. 4	✓	✓	✗	✗
inst. 5	✗	✓	✗	✗
inst. 6	✓	✗	✓	✓
inst. 7	✗	✗	✓	✗
inst. 8	✗	✓	✓	✗
acc.	5/8	**6/8**	4/8	2/8

Top-2 by accuracy: $\{s_2, s_1\}$ covers 7;
start from s_2, add s_3 (+2): $\{s_2, s_3\}$ covers all 8.

Georgia Tech

More from less experience

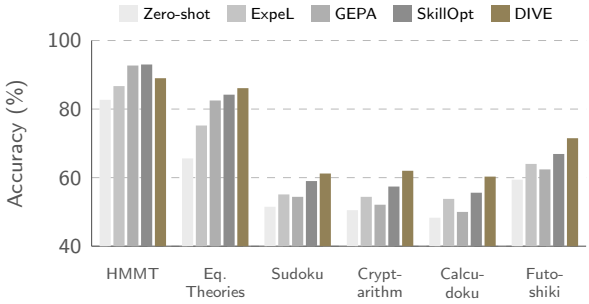


Accuracy vs. total optimization rollouts on HMMT (left) and Sudoku (right), Qwen3-8B

DIVE reaches higher accuracy with fewer rollouts than SFT, GRPO, and GEPA.
Multi-DIVE outperforms Single-DIVE, showing the benefits of multiple populations.

Georgia Tech

Evolved skills improve single runs



Qwen3.5-27B, 1 candidate per instance; logical tasks use hard subsets

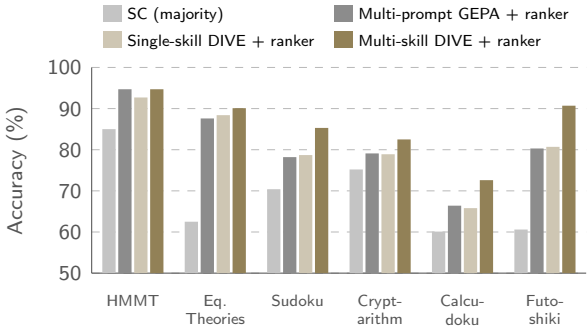
Average accuracy over six tasks (1 candidate)

	GPT-5 nano	DeepSeek v4-flash	Qwen3.5 27B
Zero-shot	52.3	73.8	59.7
ExpeL	56.3	78.5	64.9
GEPA	55.1	77.6	65.7
SkillOpt	59.7	81.2	69.4
DIVE	**60.7**	**81.9**	**71.7**

Best average for every model. The larger gains come from combining skills.

Georgia Tech

The real gain: multiple runs



Qwen3.5-27B, 10 candidates per instance; ranked methods share the same frozen-model ranker

Average accuracy over six tasks (10 candidates)

	GPT-5 nano	DeepSeek v4-flash	Qwen3.5 27B
SC (majority)	67.7	86.8	69.0
Multi-prompt GEPA + ranker	75.9	92.1	81.0
Single-skill DIVE + ranker	76.2	91.1	80.9
Multi-skill DIVE + ranker	**81.4**	**96.4**	**86.0**

For every model, multiple skills beat a single skill and multiple GEPA prompts.

Georgia Tech

Skill transfer, cost efficiency, and ablation

Cross-model transfer

Target	Zero-shot	Skills from 9B	Evolved on target
Qwen3.5-27B	59.6	81.2	86.0
DeepSeek-v4-flash	73.9	90.5	96.4

Skills transfer; evolving on the target adds more.

Small model vs. large model

	Avg. acc.	Cost / instance
GPT-5, few-shot	79.8	\$0.153
GPT-5-nano + DIVE	81.4	\$0.088

Higher accuracy at ∼58% of the cost.

Ablation: evolution mechanisms

	HMMT	Sudoku
Single operator	53.2	80.0
Uniform allocation	52.2	76.0
UCB allocation	55.8	86.0
Full (+ generated operators)	**58.5**	**88.0**

Qwen3-8B, accuracy at the final rollout budget

Ablation: skill-set construction

	Avg. acc.
Zero-shot	27.7
Best single skill	37.2
Random M skills	49.3
Top-M individual skills	54.1
Joint M-skill selection	**56.6**

Qwen3.5-9B, 10 populations, $M = 10$, six tasks

Georgia Tech

Takeaways

1. We can improve frozen models through skill evolution
2. Maintaining diversity and selecting complementary skills improve overall performance
3. Skill optimization can achieve better performance with fewer optimization rollouts

Georgia Tech

Outline

Planning and verifying reasoning through entailment graphs

Learning long-horizon reasoning through multi-level policies

Self-improvement through evolving skill populations

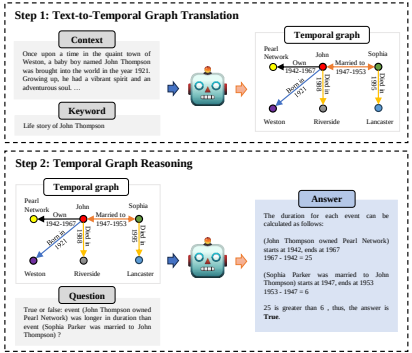
Additional contributions

TG-LLM: Large Language Models Can Learn Temporal Reasoning. Xiong, Payani, Kompella, Fekri. ACL 2024
DeepControl: Adaptive Information Control for Search-Augmented LLM Reasoning. Xiong, Gungordu, Kerce, Fekri. EMNLP 2026

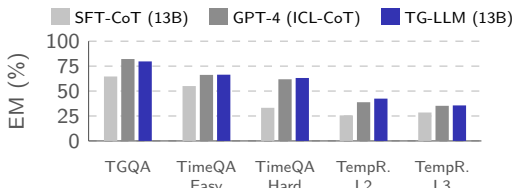
Conclusion and future work

Georgia Tech

Temporal reasoning through explicit graph structure (TG-LLM)



Main idea: translate the text into a temporal graph, then reason over the graph.

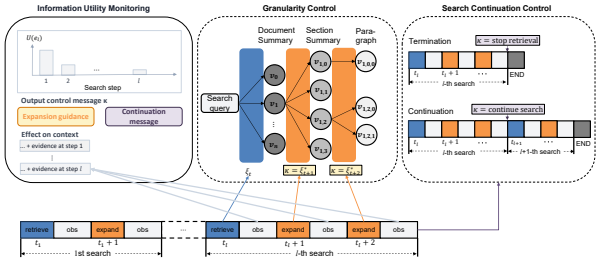


A Llama2-13B model matches GPT-4 on most benchmarks.

Georgia Tech

Adaptive information control for search-augmented reasoning (DeepControl)

Main idea: organize retrieved evidence as a hierarchy, and use an information-utility signal to decide how deep to expand it and when to stop searching; the guidance is annealed during training, so the policy learns to make these decisions on its own.



Avg. EM over 7 QA benchmarks

	7B	3B
RAG	30.4	27.0
Search-R1	43.1	32.5
DeepControl	**47.9**	**41.1**

Qwen2.5 backbones

Best results at both model sizes.

Georgia Tech

Outline

Planning and verifying reasoning through entailment graphs

Learning long-horizon reasoning through multi-level policies

Self-improvement through evolving skill populations

Additional contributions

Conclusion and future work

Georgia Tech

Conclusion

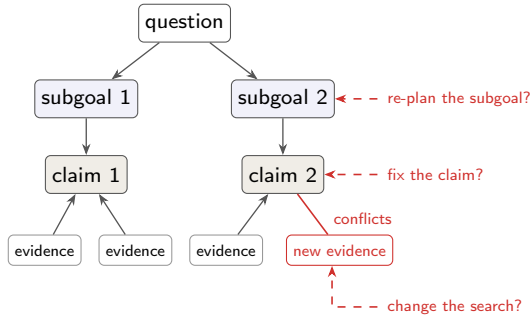
	Structure	During learning	During inference
SWAP	entailment graph	train the generator and discriminator on graph-structured data	check and compare each step against its premises
MLR	planning–execution hierarchy	learn both levels from process preferences	high-level policy plans subgoals, then low-level policy executes them
DIVE	evolving skill populations	revise and select skills	guide the model with complementary skills

The same structure that organizes reasoning also tells us how to improve it.

Georgia Tech

Future work #1: evidence-grounded structured reasoning

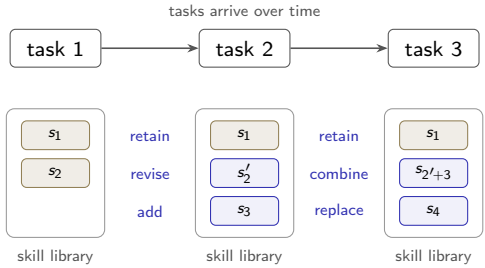
- Combine planning, evidence gathering, and verification into a single framework
- Organize subgoals, claims, and evidence into a structured graph
- Each subgoal guides the search, and new evidence can trigger re-planning
- **Key question**: when new evidence conflicts, should the system fix a claim, re-plan a subgoal, or change how it searches?



Georgia Tech

Future work #2: continual skill evolution across tasks

- DIVE focuses on a single fixed task, but in practice, new tasks arrive over time
- Reuse and combine skills learned from previous tasks, while avoiding forgetting
- **Key question**: when to retain, revise, combine, or replace existing skills



Georgia Tech

Publications resulting from thesis

Conferences

1. **Xiong, S.**, Payani, A., Kompella, R., and Fekri, F. "Large Language Models Can Learn Temporal Reasoning." *ACL*, 2024.
2. **Xiong, S.**, Payani, A., Yang, Y., and Fekri, F. "Deliberate Reasoning in Language Models as Structure-Aware Planning with an Accurate World Model." *ACL*, 2025.
3. **Xiong, S.**, Payani, A., and Fekri, F. "Enhancing Language Model Reasoning with Structured Multi-Level Modeling." *ICLR*, 2026.
4. **Xiong, S.**, Gungordu, O., Kerce, J. C., and Fekri, F. "Adaptive Information Control for Search-Augmented LLM Reasoning." *Findings of EMNLP*, 2026.

Workshop

5. **Xiong, S.**, Gungordu, O., Johnson, B., Kerce, J. C., and Fekri, F. "Scaling Search-Augmented LLM Reasoning via Adaptive Information Control." *ICLR Workshop on Scaling Post-training for LLMs*, 2026.

Under Review

6. **Xiong, S.**, Payani, A., Gungordu, O., and Fekri, F. "DIVE: Unlocking Self-Improvement in Frozen Language Models through Diversity-Driven Skill Evolution." 2026.

Georgia Tech

Other publications

Conferences

1. **Xiong, S.**, Yang, Y., Fekri, F., and Kerce, J. C. "TILP: Differentiable Learning of Temporal Logical Rules on Knowledge Graphs." *ICLR*, 2023.
2. **Xiong, S.**, Yang, Y., Payani, A., Kerce, J. C., and Fekri, F. "TEILP: Time Prediction over Knowledge Graphs via Logical Reasoning." *AAAI*, 2024.
3. Yang, Y., **Xiong, S.**, Payani, A., Kerce, J. C., and Fekri, F. "Temporal Inductive Logic Reasoning over Hypergraphs." *IJCAI*, 2024.
4. Yang, Y., **Xiong, S.**, Payani, A., Shareghi, E., and Fekri, F. "Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation." *ACL*, 2024.
5. Yang, Y., **Xiong, S.**, Payani, A., Shareghi, E., and Fekri, F. "Can LLMs Reason in the Wild with Programs?" *Findings of EMNLP*, 2024.
6. Saz, A. F., **Xiong, S.**, and Fekri, F. "DISCD: Distributed Lossy Semantic Communication for Logical Deduction of Hypothesis." *ICMLCN*, 2025.
7. Saz, A. F., **Xiong, S.**, and Fekri, F. "Lossy Semantic Communication for the Logical Deduction of the State of the World." *WCNC*, 2025.
8. Saz, A. F., **Xiong, S.**, and Fekri, F. "Analysis of Semantic Communication for Logic-based Hypothesis Deduction." *GLOBECOM*, 2025.
9. Saz, A. F., **Xiong, S.**, and Fekri, F. "On the Theory of Semantic Information and Communication for Logical Inference." *WCNC*, 2026.

Other publications (cont.)

Conferences

10. **Xiong, S.**, Zou, J., Fekri, F., and Cho, Y. J. "Long-Context Modeling with Dynamic Hierarchical Sparse Attention for Memory-Constrained LLM Inference." *ICML*, 2026.
11. Gungordu, O., **Xiong, S.**, and Fekri, F. "Pathwise: Planning through World Model for Automated Heuristic Design via Self-Evolving LLMs." *ICML*, 2026.

Workshop

12. **Xiong, S.**, Zou, J., Fekri, F., and Cho, Y. J. "Long-Context Modeling with Dynamic Hierarchical Sparse Attention for On-Device LLMs." *NeurIPS Workshop on Efficient Reasoning*, 2025.

Under Review

13. Yang, Y., **Xiong, S.**, Shareghi, E., and Fekri, F. "The Compressor-Retriever Architecture for Language Model OS." 2024.
14. **Xiong, S.**, Payani, A., and Fekri, F. "Enhancing Long Chain-of-Thought Reasoning through Multi-Path Plan Aggregation." 2025.
15. Gungordu, O., **Xiong, S.**, and Fekri, F. "MOSAIC: Adversarial Co-evolution of Specialist Heuristics and Problem Instances for LLM-based Automated Heuristic Design." 2026.
16. Hu, H., Gungordu, O., **Xiong, S.**, and Fekri, F. "Evaluator-in-the-Loop Monte Carlo Tree Search via LLM Agents for Motif Scaffolding in Protein Design." 2026.
17. Gungordu, O., Tursucular, A., **Xiong, S.**, and Fekri, F. "EPOCH: Coverage-Aware Evolution with Experiential Preference Optimization for LLM-based Automated Heuristic Design." 2026.

Thank you!

SENTINEL Lab

Advisor: Dr. Faramarz Fekri

Members: Dr. Yuan Yang, A. Faruk Saz, Oguzhan Gungordu, Haotian Hu

Committee Members

Dr. Ghassan AlRegib
Dr. Suguman Bansal
Dr. Kartik Goyal
Dr. Karthikeyan Sundaresan

Collaborators

TG-LLM, SWAP, MLR, DIVE: Dr. Ali Payani

DeepControl: Dr. James C. Kerce

Other projects: Dr. James C. Kerce, Dr. Ehsan Shareghi

Funding

NSF, DARPA, Cisco

Friends and family

Georgia Tech