ORAL PRESENTATION

FoGen Workshop, ICML 2026 • July 10 • Room 318, Seoul

Within-Episode Failure Recovery in LLM Agents via Progress-Gated Dual-Process Routing

Ankush Kadu Aswanth Krishnan

QpiAI

released open-source as **ReflexGrad** • `github.com/qpiai/reflexgrad`

Scan for paper

An agent fails a task it could solve

ALFWorld: "put a clean mug on the coffee table."

An agent fails a task it could solve

ALFWorld: "put a clean mug on the coffee table."

- Step 3: agent decides the mug is **in the cabinet**. It isn't.

An agent fails a task it could solve

ALFWorld: "put a clean mug on the coffee table."

- Step 3: agent decides the mug is **in the cabinet**. It isn't.
- Steps 4–20: opens cabinet 1, cabinet 2, cabinet 3… *minor variations of the same wrong idea.*

```
step 4:  open cabinet 1 - nothing
step 6:  open cabinet 2 - nothing
step 9:  open cabinet 3 - nothing
step 12:  open cabinet 1 - again!
⋮
step 55:  FAIL
```

An agent fails a task it could solve

ALFWorld: "put a clean mug on the coffee table."

- Step 3: agent decides the mug is **in the cabinet**. It isn't.
- Steps 4–20: opens cabinet 1, cabinet 2, cabinet 3… *minor variations of the same wrong idea.*
- Step 55: **budget exhausted**. Task failed.

```
step 4:  open cabinet 1 - nothing
step 6:  open cabinet 2 - nothing
step 9:  open cabinet 3 - nothing
step 12:  open cabinet 1 - again!
⋮
step 55:  FAIL
```

An agent fails a task it could solve

ALFWorld: "put a clean mug on the coffee table."

- Step 3: agent decides the mug is **in the cabinet**. It isn't.
- Steps 4–20: opens cabinet 1, cabinet 2, cabinet 3… *minor variations of the same wrong idea.*
- Step 55: **budget exhausted**. Task failed.

```
step 4:  open cabinet 1 - nothing
step 6:  open cabinet 2 - nothing
step 9:  open cabinet 3 - nothing
step 12:  open cabinet 1 - again!
⋮
step 55:  FAIL
```

The fix was already in its own trajectory. The failure pattern — repeated no-progress steps — is visible *during* the episode. Nothing acts on it.

Three families of methods — none closes this loop

Reflexion

Strategic self-critique ... but only **after the episode ends**, on the *next* trial. Needs demos to bootstrap.

Three families of methods — none closes this loop

Reflexion

Strategic self-critique ... but only **after the episode ends**, on the *next* trial. Needs demos to bootstrap.

TextGrad

Tactical refinement within a session ... but gradients are **local**: they sharpen a *wrong* strategy more precisely.

Three families of methods — none closes this loop

Reflexion

Strategic self-critique ... but only **after the episode ends**, on the *next* trial. Needs demos to bootstrap.

TextGrad

Tactical refinement within a session ... but gradients are **local**: they sharpen a *wrong* strategy more precisely.

Search (ToT / LATS)

Widens the action space per step ... but **never updates the policy** as the episode progresses.

Three families of methods — none closes this loop

Reflexion

Strategic self-critique ... but only **after the episode ends**, on the *next* trial. Needs demos to bootstrap.

TextGrad

Tactical refinement within a session ... but gradients are **local**: they sharpen a *wrong* strategy more precisely.

Search (ToT / LATS)

Widens the action space per step ... but **never updates the policy** as the episode progresses.

Missing piece: start tactical, **escalate to strategic** when tactical fixes stop working — **within a single episode, without demonstrations.**

Key idea: a progress signal decides *which* kind of correction

A per-step progress score $s_t = E(o_t, a_t, o_{t+1}, \tau) \in [0, 10]$ drives a deterministic router:

FAST (TextGrad)
every $k{=}3$ steps
$\sim$85% of steps
local refinement

Key idea: a progress signal decides *which* kind of correction

A per-step progress score $s_t = E(o_t, a_t, o_{t+1}, \tau) \in [0, 10]$ drives a deterministic router:

FAST (TextGrad)
every k=3 steps
$\sim$85% of steps
local refinement

SLOW (Reflexion)
m=5 consecutive low scores fire the gate
causal diagnosis + plan

Key idea: a progress signal decides *which* kind of correction

A per-step progress score $s_t = E(o_t, a_t, o_{t+1}, \tau) \in [0, 10]$ drives a deterministic router:

FAST (TextGrad)
every $k{=}3$ steps
$\sim$85% of steps
local refinement

SLOW (Reflexion)
$m{=}5$ consecutive low scores fire the gate
causal diagnosis + plan

COOL (cooldown)
$c{=}5$ steps
execute the plan
no gradient interference

Key idea: a progress signal decides *which* kind of correction

A per-step progress score $s_t = E(o_t, a_t, o_{t+1}, \tau) \in [0, 10]$ drives a deterministic router:

FAST (TextGrad)
every k=3 steps
$\sim$85% of steps
local refinement

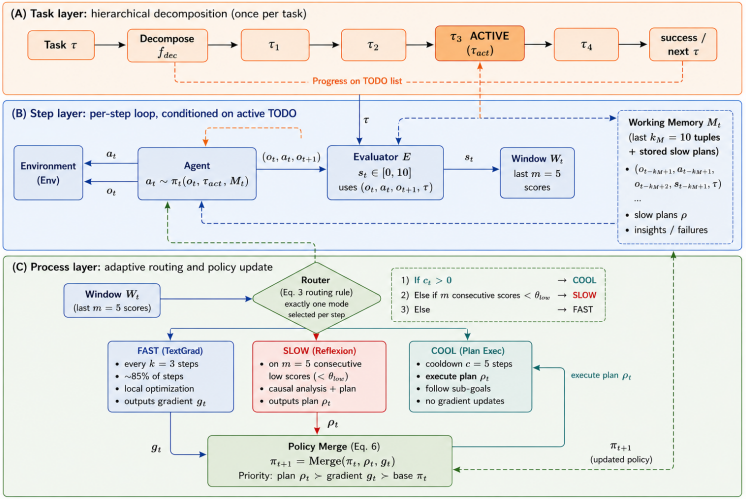
SLOW (Reflexion)
m=5 consecutive low scores fire the gate
causal diagnosis + plan

COOL (cooldown)
c=5 steps
execute the plan
no gradient interference

Priority merge: plan $\succ$ gradient $\succ$ base policy — keeps the natural-language policy coherent.

Each SLOW activation emits **3 observable artifacts**: (1) reproducible trigger (2) causal diagnostic (3) verified fix.

System architecture



Evaluator $\rightarrow$ score window W_t $\rightarrow$ router $\rightarrow$ priority merge $\rightarrow$ updated policy π_{t+1}.

The routing rule — deterministic, three lines

$$R_t = \begin{cases} \textsf{COOL}, & \text{if } c_t > 0 \\ \\ \\ \end{cases}$$

The routing rule — deterministic, three lines

$$R_t = \begin{cases} \textbf{COOL}, & \text{if } c_t > 0 \quad \text{(a plan is executing — protect it)} \\ & \\ & \end{cases}$$

The routing rule — deterministic, three lines

$$R_t = \begin{cases} \textbf{COOL}, & \text{if } c_t > 0 \quad \text{(a plan is executing — protect it)} \\ \textbf{SLOW}, & \text{if all last } m{=}5 \text{ scores} < \theta_{\text{low}} \quad \text{(sustained stall — not one noisy dip)} \end{cases}$$

The routing rule — deterministic, three lines

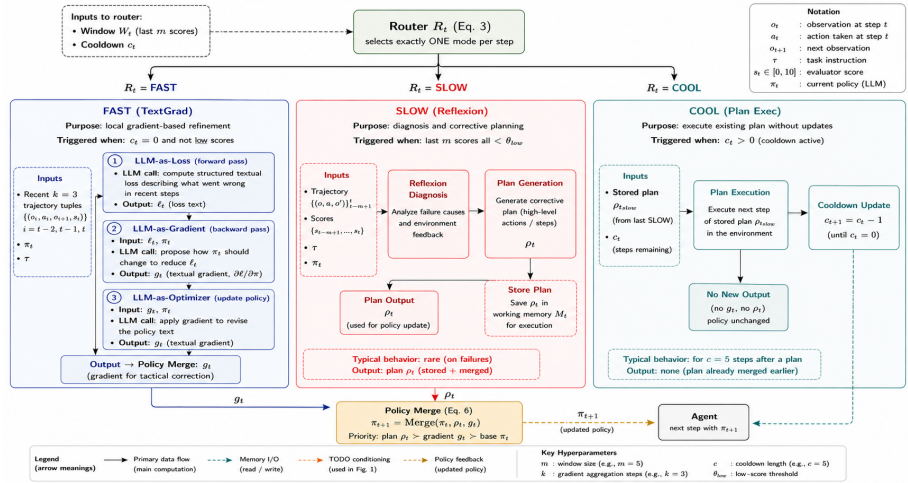
$$R_t = \begin{cases} \textbf{COOL}, & \text{if } c_t > 0 \quad \text{(a plan is executing — protect it)} \\ \textbf{SLOW}, & \text{if all last } m{=}5 \text{ scores} < \theta_{\text{low}} \quad \text{(sustained stall — not one noisy dip)} \\ \textbf{FAST}, & \text{otherwise} \quad \text{(default: cheap local refinement)} \end{cases}$$

The routing rule — deterministic, three lines

$$R_t = \begin{cases} \textbf{COOL}, & \text{if } c_t > 0 \quad \text{(a plan is executing — protect it)} \\ \textbf{SLOW}, & \text{if all last } m{=}5 \text{ scores} < \theta_{\text{low}} \quad \text{(sustained stall — not one noisy dip)} \\ \textbf{FAST}, & \text{otherwise} \quad \text{(default: cheap local refinement)} \end{cases}$$

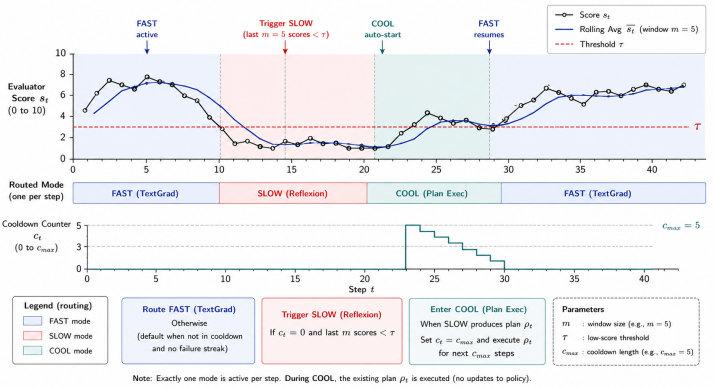
- **Consecutive**-low trigger, not average-low $\Rightarrow$ robust to single evaluator errors ($\sim$3% false-positive rate; 96% self-correct within 2 steps).
- Cooldown $c{=}5$: without it, gradients *rewrite the policy mid-plan* — recovery dies. A key design detail.
- No training, no demos, single model — the router is 100% inference-time.

Inside the two processes



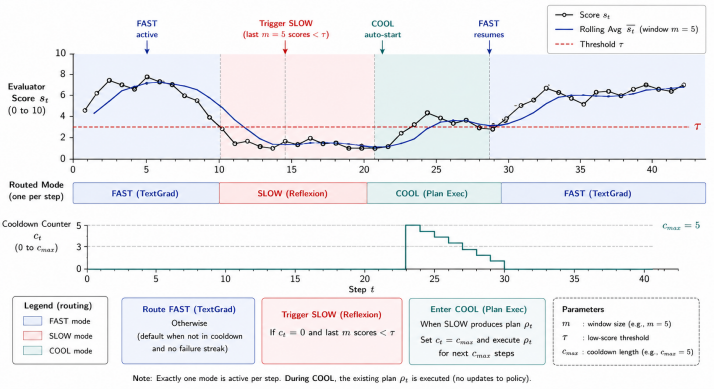
FAST: 4-stage local TextGrad loop. SLOW: 4-stage Reflexion causal reasoning → plan ρ_t.

What recovery actually looks like



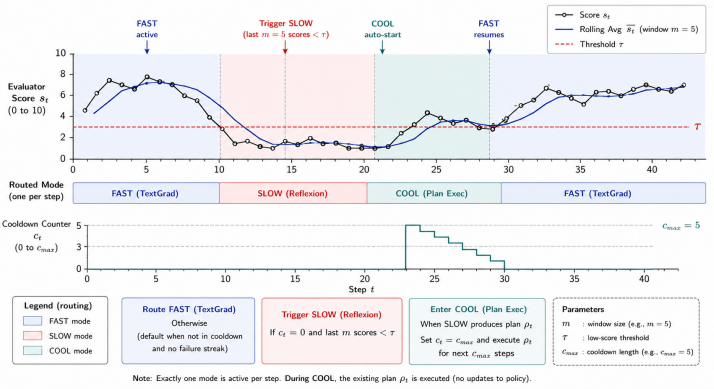
Note: Exactly one mode is active per step. **During COOL**, the existing plan ρ_t is executed (no updates to policy).

What recovery actually looks like



- Scores **dip and stay low** → gate fires SLOW → diagnosis: *"you are searching containers; the mug is on a surface"* → plan.

What recovery actually looks like



- Scores **dip and stay low** → gate fires SLOW → diagnosis: *"you are searching containers; the mug is on a surface"* → plan.
- COOL protects the plan for 5 steps → scores recover → FAST resumes. **Task solved inside the same episode.**

Results: the lift is architectural

ALFWorld, 134 tasks, n=10 seeds, **no demonstrations**

	GPT-5	Qwen-3-8B
Zero-shot	46.3	35.1
DPR	**88.1**	**75.4**
Lift	**+41.8pp**	**+40.3pp**

Results: the lift is architectural

ALFWorld, 134 tasks, $n{=}10$ seeds, **no demonstrations**

	GPT-5	Qwen-3-8B
Zero-shot	46.3	35.1
DPR	**88.1**	**75.4**
Lift	**+41.8pp**	**+40.3pp**

Cross-model gap: 1.5pp — within seed noise ($p{\approx}0.13$).

The gain does *not* come from model scale. An open-weight 8B model gets the **same architectural lift** as a frontier model.

Results: the lift is architectural

ALFWorld, 134 tasks, n=10 seeds, **no demonstrations**

	GPT-5	Qwen-3-8B
Zero-shot	46.3	35.1
DPR	**88.1**	**75.4**
Lift	**+41.8pp**	**+40.3pp**

Cross-model gap: 1.5pp — within seed noise ($p \approx 0.13$).

The gain does *not* come from model scale. An open-weight 8B model gets the **same architectural lift** as a frontier model.

Every model-specific gain separated from zero-shot at $p < 10^{-6}$.

Zero demonstrations beats one-shot search — compute-matched

Same model (Qwen-3-8B), baselines in their *most favorable* 1-shot setup:

Method	Demos	Calls/task	Success
Self-Refine	1-shot	~ 55	68.7
Tree of Thoughts	1-shot	~ 100	69.7
LATS	1-shot	~ 140	72.7
DPR	**None**	$\sim \mathbf{100}$	**75.4**

Zero demonstrations beats one-shot search — compute-matched

Same model (Qwen-3-8B), baselines in their *most favorable* 1-shot setup:

Method	Demos	Calls/task	Success
Self-Refine	1-shot	~ 55	68.7
Tree of Thoughts	1-shot	~ 100	69.7
LATS	1-shot	~ 140	72.7
DPR	**None**	$\sim$ **100**	**75.4**

Beats 1-shot LATS by **+2.7pp** at ∼30% *less* compute ($p \approx 0.01$); ToT **+5.7pp**; Self-Refine **+6.7pp**.

Zero demonstrations beats one-shot search — compute-matched

Same model (Qwen-3-8B), baselines in their *most favorable* 1-shot setup:

Method	Demos	Calls/task	Success
Self-Refine	1-shot	~ 55	68.7
Tree of Thoughts	1-shot	~ 100	69.7
LATS	1-shot	~ 140	72.7
DPR	**None**	$\sim$ **100**	**75.4**

Beats 1-shot LATS by **+2.7pp** at ∼30% *less* compute ($p \approx 0.01$); ToT **+5.7pp**; Self-Refine **+6.7pp**.

Why? Search assumes a demonstration-grounded prior. **DPR** builds its scaffolding *during* the episode — gradients, diagnosis, progress signal.

Where it works — and where it doesn't (yet)

Robust

- Threshold sweeps over $k, m, \theta_{\text{low}}$: success stays within **84.3–88.1%**.
- Synergy is super-additive: $+12.0$pp beyond the sum of parts (GPT-5).
- Cross-domain probes: TextWorld **+33pp**; OSWorld $14 \rightarrow 16/20$.

Where it works — and where it doesn't (yet)

Robust

- Threshold sweeps over $k, m, \theta_{\text{low}}$: success stays within **84.3–88.1%**.
- Synergy is super-additive: $+12.0$pp beyond the sum of parts (GPT-5).
- Cross-domain probes: TextWorld **+33pp**; OSWorld $14 \rightarrow 16/20$.

Honest limits

- **5.2pp gap** to 1-shot ReflAct — localized to Heat & Examine: world knowledge a demo transfers directly. *We do not claim parity.*
- Evaluator is an LLM ($\sim$3% false positives); success is always scored by ALFWorld's deterministic oracle, never by E.
- Router ablations (fixed-cadence, random gate) are the next experiment.

Where it works — and where it doesn't (yet)

Robust

- Threshold sweeps over $k, m, \theta_{\text{low}}$: success stays within **84.3–88.1%**.
- Synergy is super-additive: $+12.0$pp beyond the sum of parts (GPT-5).
- Cross-domain probes: TextWorld **+33pp**; OSWorld $14 \rightarrow 16/20$.

Honest limits

- **5.2pp gap** to 1-shot ReflAct — localized to Heat & Examine: world knowledge a demo transfers directly. *We do not claim parity.*
- Evaluator is an LLM ($\sim$3% false positives); success is always scored by ALFWorld's deterministic oracle, never by E.
- Router ablations (fixed-cadence, random gate) are the next experiment.

Failure modes on Qwen-3-8B (33 fails): world-knowledge 21, navigation 8, evaluator noise 4 — the first two are recoverable in principle.

Takeaways

- **Demo-free, in-episode recovery is achievable** — and beats compute-matched 1-shot search.

Takeaways

- **Demo-free, in-episode recovery is achievable** — and beats compute-matched 1-shot search.
- The mechanism is a **progress-gated router**: tactical refinement by default, strategic diagnosis exactly when progress stalls, a cooldown to let plans breathe.

Takeaways

- **Demo-free, in-episode recovery is achievable** — and beats compute-matched 1-shot search.
- The mechanism is a **progress-gated router**: tactical refinement by default, strategic diagnosis exactly when progress stalls, a cooldown to let plans breathe.
- The lift is **architectural** — it transfers across model scales (8B $\approx$ frontier, $\pm$seed noise).

Takeaways

- **Demo-free, in-episode recovery is achievable** — and beats compute-matched 1-shot search.
- The mechanism is a **progress-gated router**: tactical refinement by default, strategic diagnosis exactly when progress stalls, a cooldown to let plans breathe.
- The lift is **architectural** — it transfers across model scales (8B $\approx$ frontier, $\pm$seed noise).

Code, prompts, per-seed logs, sensitivity sweeps — all released as **ReflexGrad**:

`github.com/qpiai/reflexgrad`

Poster today • AIWILD tomorrow, Hall A

Scan for paper

Backup slides

Backup: Router ablation status

- Sensitivity sweeps over all three routing thresholds ($k \in \{2, 3, 5\}$, $m \in \{3, 5, 7\}$, $\theta_{\text{low}} \in \{3, 4, 7\}$) bound success within **84.3–88.1%** — the rule is not knife-edge tuned.
- Fixed-cadence (slow every m steps regardless of scores) and random-gate (15% rate) ablations are designed and queued — the cleanest way to isolate the routing decision from component availability.
- Design argument: fixed-cadence ignores the structural signature SLOW targets — consecutive low scores under continued local refinement *is* the trace of a wrong strategy.

Backup: Is the evaluator a hidden demonstration?

- E is the **routing signal only**. Task success in every table is measured by ALFWorld's deterministic completion oracle — never by E.
- Evaluator noise is characterized: $\sim$3% false-positive rate over $\sim$8,000 calls; 96% of isolated false positives self-correct within 2 steps; $m{=}5$ consecutive filtering suppresses spurious SLOW triggers (zero observed on GPT-5 run).
- Planned: redaction experiment (remove task description from E's prompt) and a non-LLM progress heuristic — directly tests whether E amortizes a demonstration.

Backup: Baseline fairness

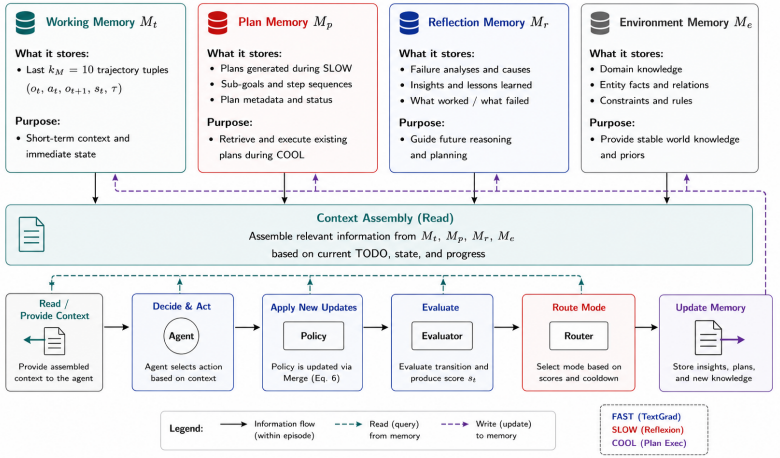
- Self-Refine, ToT, LATS re-implemented on Qwen-3-8B (none publish ALFWorld numbers); each anchored within $\sim$2pp of its published number on its original benchmark.
- Comparison is conservative by construction: baselines run their published *most favorable* 1-shot configuration; **DPR** runs zero-shot (hardest).
- All prompts, configs, and per-task logs released for independent re-runs.

Backup: Per-category breakdown (Qwen-3-8B)

Category	**DPR** success
Pick & Place	90.9%
Clean & Place	88.5%
Pick Two	81.8%
Cool & Place	77.3%
Heat / Examine	lower — world-knowledge bound

The 5.2pp ReflAct gap concentrates in Heat/Examine: verb→receptacle knowledge (heat→microwave, examine→lamp) that one demonstration transfers directly.

Backup: Memory subsystem



Working memory (k_M=10), plan memory, gradient archive, failure memory, environment scratchpad — supporting subsystem; the contribution is the router.

Backup: Cross-domain + stats detail

- **TextWorld** (9 cooking/treasure tasks, GPT-5): $56\% \rightarrow 89\%$ ($+33$pp) — applicability evidence on a different text-game family.
- **OSWorld** (20 visual GUI tasks, 7 app categories): zero-shot $14/20 \rightarrow$ **DPR** $16/20$; the 2 recoveries required multi-step error recovery that activated SLOW. Reported as cross-modality evidence, not a benchmark-wide claim.
- Stats: $n{=}10$ seeds $\{42, 123, 456, 789, 1024, 1337, 2025, 3141, 5926, 7531\}$; Welch's t; LATS gap $t{\approx}2.87$, $p{\approx}0.010$; ToT $p{\approx}1.7{\times}10^{-5}$; Self-Refine $p{\approx}1.0{\times}10^{-6}$.
- Policy growth bounded: $\sim$150 tokens (step 1) $\rightarrow\sim$380 (step 15), max 520 observed.