CCE-MOBILITY: COUNTERFACTUAL WORLD–AGENT CO-EVOLUTION FOR MOBILITY AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM) agents are increasingly used for mobility planning and tool-based assistance, yet the world may change after a plan or commitment has already been formed. Roads may be blocked, services may become unavailable, or acquired information may become stale, requiring the agent to revise its decisions while preserving the original user need. While self-evolving agents learn from interaction experience and language world models simulate environment dynamics, an important question remains underexplored: what plausible environmental change would invalidate an already-made decision while leaving the original task achievable? We introduce **CCE-Mobility**, a counterfactual world–agent co-evolution framework for recovery under dynamic world changes. Given an observed trajectory, the world model reasons about decision-supporting dependencies and constructs a localized, executable intervention that invalidates one while preserving the original goal. The resulting counterfactual world provides recovery experience for agent adaptation, while persistent failures guide subsequent world-side evolution. Thus, the agent learns to recover from challenging world changes, while the world model learns to construct counterfactuals targeting unresolved weaknesses. Experiments on TravelPlanner and MobilityBench show consistent recovery gains across all evaluated backbones. On TravelPlanner, CCE-Mobility improves Recovery by 5.1%–70.9% and reduces Low by 16.7%–51.0% relative to unevolved agents. Paired validation further shows that world-side evolution reduces Low by an additional 21.8% relative to agent-only adaptation. Code will be made publicly available at `https://anonymous.4open.science/r/CCE-Mobility-4254`.

1 INTRODUCTION

Mobility permeates everyday decision-making, from daily commuting and local services to long-distance travel (Barbosa et al., 2018; Alessandretti et al., 2020). Digital mobility systems have evolved from basic positioning and point-to-point routing toward context-aware assistance supported by maps, real-time traffic, and points of interest. Building on this infrastructure, large language model (LLM) agents combine natural-language interaction, external tools, and multi-step reasoning for travel planning, route planning, and local-service assistance (Xie et al., 2024; Song et al., 2026; Lan et al., 2025). Together, these capabilities motivate a shift from one-shot route generation toward persistent, context-aware mobility assistance.

Unlike static question answering, however, mobility decisions are made against a world that continues to change. After an agent has selected a route, a point of interest, or a service, a road may become blocked, the selected resource may become unavailable, or previously acquired information may become stale before the user's task is completed. The challenge is therefore not only to make a good decision once, but also to recognize when the assumptions supporting an existing decision no longer hold and revise it while still satisfying the user need.

Existing benchmarks evaluate mobility planning and tool use (Xie et al., 2024; Song et al., 2026), with recent work extending evaluation to disruptions and changing constraints (Karmakar et al., 2026; Liu et al., 2026). These settings reveal an important limitation of factual interaction alone as a learning signal. A successful trajectory shows what decision worked in the observed world, but provides little evidence about how the agent would respond if one of the conditions supporting that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: **Motivation for CCE-Mobility.** Under the same intent and world change, original-environment self-evolution (left) may retain a stale waypoint, whereas CCE-Mobility (right) learns evidence-grounded recovery through counterfactual World–Agent co-evolution.

decision no longer held. For example, observing that an agent selects a reachable gas station does not tell us whether it would re-query, verify, and reroute if that station became unavailable. Improving recovery therefore requires not only collecting more trajectories, but also asking controlled "what-if" questions around decisions that have already been made.

Language world models provide a natural basis for reasoning about such alternatives. Recent work uses them to simulate agent environments or augment policy learning with predictive world-modeling objectives (Zuo et al., 2026; Lu et al., 2026), while other methods adapt tasks or environments to evolving agent capabilities (Guo et al., 2025). Our setting asks a more targeted question: *given an already-instantiated decision, which environmental conditions support it, and what plausible local change would invalidate one while keeping the original task achievable?* We frame this as intervention-oriented world modeling: instead of modeling complete future dynamics, the world side reasons over the current trajectory, identifies decision-supporting dependencies, and constructs an alternative world by intervening on selected ones.

To study this problem, we introduce **C**ounterfactual **C**o-**E**volution for **Mobility** Agents (**CCE-Mobility**), a framework for counterfactual world–agent co-evolution. Starting from an observed mobility trajectory, the world side reasons about task-relevant dependencies underlying the agent's current decision, such as resource availability, route accessibility, temporal conditions, or supporting observations. It then constructs a localized and executable counterfactual intervention that invalidates a selected dependency while preserving the original user goal and task constraints, with candidate changes screened for recoverability. This creates a paired alternative to the observed world: the preceding interaction remains unchanged, but the agent must now respond to a world in which an assumption behind its previous decision no longer holds. The resulting branch exposes recovery behavior that would otherwise remain unobserved and turns counterfactual world changes into actionable recovery experience.

Counterfactual recovery also provides a natural signal for co-evolution. Agent-side adaptation learns reusable recovery knowledge from intervention- induced failures, but some failures remain unresolved afterward. Resolved failures reflect weaknesses the agent has learned to handle, whereas persistent ones expose recovery gaps that remain informative for world evolution. CCE-Mobility therefore re-evaluates the same counterfactual challenges after agent adaptation and routes only persistent failures to the world side, where they are abstracted into knowledge for constructing new challenges in other contexts. In this loop, *the agent learns how to recover, while the world side learns which counterfactuals remain informative*. Both sides evolve through external memories while the underlying foundation models remain fixed.

We evaluate CCE-Mobility on TravelPlanner (Xie et al., 2024) and MobilityBench (Song et al., 2026) across multiple LLM backbones. Across all evaluated backbones, CCE-Mobility consistently improves recovery quality over the unevolved agent. A paired validation-panel analysis on

TravelPlanner further indicates an incremental effect from world-side evolution, while additional experiments examine descriptive complementarity with SkillOpt.

Our contributions are summarized as follows:

- We formulate mobility recovery through *trajectory-conditioned counterfactual world intervention*, where environmental dependencies supporting an already-made decision are selectively altered while preserving the original user goal and task constraints.
- We introduce **CCE-Mobility**, a counterfactual world–agent co-evolution framework in which counterfactual recovery experience improves agent-side knowledge, while failures that persist after adaptation guide the construction of subsequent world challenges.
- We evaluate CCE-Mobility on TravelPlanner and MobilityBench across multiple LLM backbones, showing consistent recovery-quality gains over unevolved agents and further analyzing world-side evolution and descriptive complementarity with SkillOpt.

2 Related Work

Mobility Planning under Dynamic Conditions. Research on mobility agents has progressed from static plan generation toward tool-grounded planning and adaptation in changing environments. TravelPlanner and MobilityBench establish realistic settings for multi-step planning, tool use, and constraint satisfaction (Xie et al., 2024; Song et al., 2026). Recent studies have further examined itinerary revision and tool-use adaptation under dynamic disruptions (Karmakar et al., 2026; Liu et al., 2026). Together, these works highlight that mobility agents must remain effective when the external conditions underlying their previous decisions change. They primarily study how such adaptation should be evaluated, however, rather than how world changes can be systematically constructed around the agent's current decisions and subsequently turned into learning experience.

Experience-Driven Agent Adaptation. Agents can improve from interaction experience without updating their foundation models. Reflexion stores language feedback, while Mobile-Agent-E, SkillOpt, SkillOS, and MUSE-Autoskill convert past interactions into reusable procedures, skills, or memories (Shinn et al., 2023; Wang et al., 2025; Yang et al., 2026b; Ouyang et al., 2026; Lin et al., 2026). These methods primarily optimize agent-side knowledge, motivating the complementary question of whether the experience-generating world should also adapt as agent capabilities evolve.

World Modeling and Adaptive Co-Evolution. Recent language world models increasingly characterize the external environments in which agents operate. Qwen-AgentWorld learns to simulate agentic environments and their responses to agent actions; DynaWeb uses a learned web world model to support synthetic interaction and imagined rollouts; WebEvolver employs a world model both as a virtual environment for self-improvement and as an imagination module for look-ahead reasoning; and PaW incorporates action–next-observation world-modeling supervision directly into policy learning (Zuo et al., 2026; Ding et al., 2026; Fang et al., 2025; Lu et al., 2026). These works demonstrate how world modeling can support agent learning and decision making by capturing how environments respond and evolve.

A closely related line of work adapts the experience presented to an agent as its capabilities evolve. PAIRED and replay-guided environment design adjust environments according to policy competence (Dennis et al., 2020; Jiang et al., 2021). Recent LLM-agent methods extend to agent–data and agent–environment co-evolution: CoEvolve uses rollout-derived signals to guide task synthesis, GenEnv adapts environment difficulty, and Agent-World expands environment and task distributions around capability gaps (Yang et al., 2026a; Guo et al., 2025; Dong et al., 2026). These approaches mainly adapt task distributions, environment difficulty, or broader experience spaces.

CCE-Mobility instead focuses on localized counterfactual world changes grounded in observed tool-use trajectories. It identifies environmental dependencies supporting an already-instantiated decision and intervenes on them to construct targeted alternative worlds for recovery. The resulting experience alternately updates agent-side recovery knowledge and world-side intervention knowledge, while post-adaptation residual failures guide subsequent counterfactual challenges. Recovery is evaluated against the changes actually instantiated in the environment, grounding feedback for both sides of the co-evolution loop.

3 METHOD

CCE-Mobility converts observed mobility decisions into controlled counterfactual recovery experience for world–agent co-evolution. From a factual trajectory, the world side reasons about dependencies supporting the current decision and constructs a localized intervention that preserves the original task. The agent then attempts recovery in the resulting alternative world. Recovery failures drive agent-side adaptation, while persistent failures guide world-side evolution. We next formalize the framework and its three components: counterfactual intervention, agent-side recovery evolution, and residual-guided world–agent co-evolution.

3.1 PROBLEM FORMULATION

Task setting. We study recovery in tool-using mobility agents when a local world intervention invalidates a dependency supporting an existing plan or commitment. The user intent and task constraints remain unchanged, requiring the agent to revise affected decisions while completing the original task.

For benchmark B, we model agent–environment interaction as a finite-horizon partially observable Markov decision process (POMDP) (Kaelbling et al., 1998), denoted by $\mathfrak{P}_B$ with horizon H. Its latent state $s_t = (x_t^1, \ldots, x_t^m)$ contains task-relevant world factors, such as road status and resource availability, together with execution context such as time and task progress. These variables define where local interventions may act. The user goal g induces a trajectory-level criterion r_g for evaluating whether the original task requirements are satisfied.

The agent observes the environment only through interaction events and tool responses. At step t, its history is $h_t = (o_0, a_0, \ldots, a_{t-1}, o_t)$ and it acts as $a_t \sim \pi_\theta(\cdot \mid g, h_t)$, where o_t and a_t denote observations and actions, respectively. Retrieved skills from external memory $\mathcal{M}_{\mathrm{A}}$ may additionally condition the policy (Section 3.4).

World changes and recovery. Consider a nonterminal checkpoint $k < H$ in a factual prefix $\tau^{\mathrm{F}}_{\leq k}$ after a plan or commitment has been formed but before the world change. The observable history h_k records the evidence available to the agent, while the privileged state s_k is used only for challenge construction and environment execution.

A local intervention δ modifies selected post-checkpoint world factors:

$$(\mathfrak{P}_B^\delta, s_k^\delta, \chi_\delta) = \mathcal{I}_B(\mathfrak{P}_B, s_k; \delta), \qquad g^\delta = g, \quad h_k^\delta = h_k. \tag{1}$$

Here $\mathfrak{P}_B^\delta$ is the intervened environment, s_k^δ its post-intervention state, and χ_δ the instantiated before–after evidence. Benchmark-specific interfaces realize $\mathcal{I}_B$ by modifying only targeted world factors or subsequent tool behavior.

An intervention is admitted as a recovery challenge only if it invalidates a plan-supporting dependency and passes recoverability checks under the original constraints and available tools. For example, a blocked route segment is admissible when benchmark-specific checks indicate that a feasible alternative remains.

The agent then continues from the preserved factual history:

$$\tau^+_{\delta,k} \sim \mathrm{Continue}(\pi_\theta, \mathcal{M}_{\mathrm{A}}; g, h_k \mid \mathfrak{P}_B^\delta, s_k^\delta), \tag{2}$$

where $\mathcal{M}_{\mathrm{A}}$ is fixed during the rollout. Terms after $\mid$ initialize the environment rather than the policy context. The agent observes the change only through subsequent tool responses or environment notifications, yielding the complete interaction $\bar{\tau}_{\delta,k} = h_k \oplus \tau^+_{\delta,k}$.

We define *same-intent recovery* as revising decisions invalidated by the world change while satisfying the original criterion r_g without relying on invalidated resources or assumptions. Section 3.3 instantiates this requirement through execution-grounded recovery contracts. Our objective is to improve recovery on held-out world changes while retaining performance on unmodified tasks.

3.2 FRAMEWORK OVERVIEW

Figure 2 summarizes CCE-Mobility. For each run, the agent policy π_θ and world-side model W remain fixed while two external memories evolve: $\mathcal{M}_{\mathrm{A}}$ stores reusable recovery skills, and $\mathcal{M}_{\mathrm{W}}$

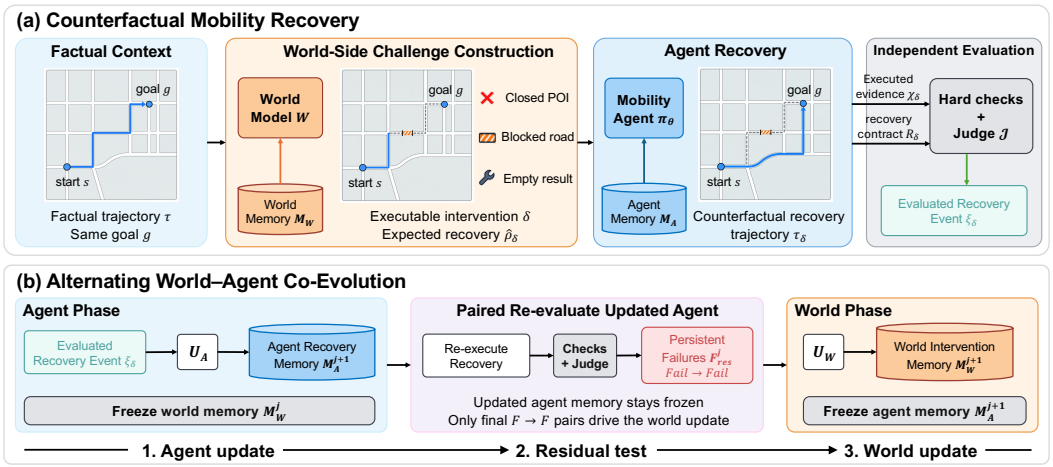

Figure 2: **Overview of CCE-Mobility.** (a) **Counterfactual recovery:** a trajectory-conditioned world intervention invalidates a dependency supporting the current decision while preserving the original task. (b) **Residual-guided co-evolution:** recovery failures improve agent memory, while failures that persist after adaptation guide world-side evolution.

stores intervention knowledge. Given a factual trajectory, the world side constructs δ and derives $\widehat{\rho}_\delta$; execution produces χ_δ, from which R_δ is grounded. Failed recoveries update $\mathcal{M}_{\mathrm{A}}$, while failures persisting after adaptation update $\mathcal{M}_{\mathrm{W}}$.

3.3 Trajectory-Conditioned Counterfactual World Intervention

A useful recovery challenge should invalidate a dependency supporting the factual decision without changing the underlying user intent. Starting from the factual execution prefix $\tau^{\mathrm{F}}_{\leq k}$, the world side reasons about task-relevant dependencies such as selected resources, route segments, reservations, timing constraints, and supporting observations. Conditioned on the factual context and current world memory $\mathcal{M}_{\mathrm{W}}$, the fixed world model proposes a local intervention $\delta \sim W(\cdot \mid \tau^{\mathrm{F}}_{\leq k}, g, \mathcal{M}_{\mathrm{W}})$. Each sampled δ specifies an affected entity and a changed world condition using an operator supported by the target benchmark. It remains a candidate proposal and must pass the validity and recoverability checks below before being instantiated.

Given the affected dependency and intervention type, the world side further derives an expected recovery specification $\widehat{\rho}_\delta$ from the supported recovery capabilities. Rather than prescribing an oracle action sequence, $\widehat{\rho}_\delta$ describes semantic recovery requirements, such as detecting the change, revoking an invalid commitment, reacquiring current evidence, revising the affected decision, and restoring the original goal. Intervention construction and recovery semantics are therefore coupled without requiring both to be produced by a single call to W.

Candidate interventions are screened using benchmark-specific checks for factual relevance, executability, locality, non-leakage, and recoverability. Only proposals that pass these checks are instantiated as recovery challenges. The benchmark adapter executes δ, and χ_δ records the change actually applied to the environment. This execution evidence is authoritative: evaluation is grounded in the instantiated change rather than solely in the original proposal.

The expected recovery specification is then grounded against the executed intervention and original task requirements as $R_\delta = G(g, \delta, \widehat{\rho}_\delta, \chi_\delta)$, where G aligns the semantic recovery expectation with the original task and execution evidence and introduces no additional learned model. The resulting contract R_δ is determined without conditioning on the agent's eventual recovery behavior and is never exposed to the acting policy.

After the agent produces the continuation in Section 3.1, recovery is assessed using deterministic constraint checks together with an evaluator J isolated from the world-side proposal process. Hard checks verify benchmark constraints and whether the agent relies on resources or assumptions invalidated by the intervention, while J evaluates semantic recovery quality against R_δ.

Algorithm 1 Residual-Guided World–Agent Co-Evolution

Require: training tasks $\mathcal{D}$, fixed models π_θ and W, co-evolution rounds K
1: Initialize $\mathcal{M}_{\mathrm{A}}^1$ and $\mathcal{M}_{\mathrm{W}}^1$
2: **for** round $j = 1, \ldots, K$ **do**
3: $\quad$ `challenges` $\leftarrow$ ConstructAndFreeze$(\mathcal{D}, W, \mathcal{M}_{\mathrm{W}}^j)$
4: $\quad$ `pre` $\leftarrow$ Evaluate$(\texttt{challenges}, \pi_\theta, \mathcal{M}_{\mathrm{A}}^j)$
5: $\quad$ $\mathcal{M}_{\mathrm{A}}^{j+1} \leftarrow U_A(\mathcal{M}_{\mathrm{A}}^j, \texttt{pre}; \texttt{challenges})$
6: $\quad$ `post` $\leftarrow$ Evaluate$(\texttt{challenges}, \pi_\theta, \mathcal{M}_{\mathrm{A}}^{j+1})$
7: $\quad$ $F_{\mathrm{res}}^j \leftarrow$ PersistentFailures$(\texttt{pre}, \texttt{post})$
8: $\quad$ $\mathcal{M}_{\mathrm{W}}^{j+1} \leftarrow U_W(\mathcal{M}_{\mathrm{W}}^j, F_{\mathrm{res}}^j)$
9: **end for**
Ensure: training checkpoints with evolved $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{W}}$

The resulting auditable event ξ_δ links the factual context, executed intervention, recovery requirements, continuation, and evaluation outcome. When benchmark-specific final-pass information is available, we define

$$\mathrm{Success}(\xi_\delta) = \big[\mathrm{Recovery}(\xi_\delta) \geq 3\big] \wedge \big[\mathrm{FinalPass}(\xi_\delta) = 1\big], \tag{3}$$

and $\mathrm{Fail}(\xi_\delta)$ as its complement, with missing recovery scores also treated as failures. When final-pass information is unavailable, the predicate reduces to the Recovery threshold alone. The benchmark-specific definition of $\mathrm{FinalPass}$ follows the corresponding task constraints.

Importantly, this co-evolution failure predicate is distinct from the Low metric reported in Section 4: Low uses the diagnostic threshold $\mathrm{Recovery} < 2$, whereas Fail also includes moderate-quality recoveries with scores below 3 and recoveries that fail benchmark-specific final acceptance.

3.4 Agent-Side Recovery-Skill Evolution

Agent-side evolution is driven by recovery failures. Given evaluated recovery events, U_A identifies recurring failure patterns according to affected capability, world-change type, invalidated dependency, and recovery stage. Failed cases are used to synthesize candidate corrective skills specifying an observable trigger, recovery procedure, failure guards, success conditions, and provenance.

The resulting skills are stored in external agent recovery memory $\mathcal{M}_{\mathrm{A}}$ rather than encoded through updates to π_θ. World-side metadata may be used offline to organize recovery experience during skill construction, but is neither stored as a deployment-time retrieval key nor exposed to the acting agent.

Let `pre` denote recovery records produced by the current agent memory on a frozen set of `challenges`. The round-level update is

$$\mathcal{M}_{\mathrm{A}}^{j+1} = U_A\Big(\mathcal{M}_{\mathrm{A}}^j, \texttt{pre}; \texttt{challenges}\Big). \tag{4}$$

U_A combines failure-driven candidate-skill synthesis with paired curation on the same frozen challenges. Previously failed cases test whether candidate skills repair the targeted weaknesses, whereas previously successful cases serve as retention anchors for detecting regressions. Paired outcomes and hard constraint violations guide retain, repair, or reject decisions; if no candidate change is retained, the previous memory is carried forward.

The resulting memory is then held fixed and re-evaluated on the same frozen challenge set. These post-update records are used for subsequent world-side adaptation, ensuring that world evolution reflects the behavior of the curated agent memory rather than discarded candidate changes.

At inference time, the policy retrieves skills from $\mathcal{M}_{\mathrm{A}}$ using only g and observable history h_t, without access to the intervention label, $\widehat{\rho}_\delta$, χ_δ, R_δ, evaluator outputs, or oracle actions.

3.5 Residual-Guided World–Agent Co-Evolution

CCE-Mobility alternates agent-side and world-side adaptation. At round j, the current world memory $\mathcal{M}_{\mathrm{W}}^j$ conditions a set of recovery challenges. For paired comparison, each challenge preserves

the same factual history, executed intervention, execution evidence, and recovery requirements before and after the agent update.

The current agent memory $\mathcal{M}_{\mathrm{A}}^{j}$ is first evaluated on these frozen challenges. Agent-side evolution produces $\mathcal{M}_{\mathrm{A}}^{j+1}$ according to Equation 4, after which the same challenges are re-executed. The paired outcomes distinguish recovery improvement (failure-to-success), retained competence (success-to-success), regression (success-to-failure), and persistent weakness (failure-to-failure).

Only persistent failures are routed to world-side adaptation:

$$F_{\mathrm{res}}^{j} = \left\{ \xi_{i,\mathrm{post}}^{j} \;\middle|\; \mathrm{Fail}(\xi_{i,\mathrm{pre}}^{j}) \wedge \mathrm{Fail}(\xi_{i,\mathrm{post}}^{j}) \right\}. \tag{5}$$

Success-to-failure cases are treated as agent-side regressions and excluded from world updates.

The world memory is updated by

$$\mathcal{M}_{\mathrm{W}}^{j+1} = U_W\left(\mathcal{M}_{\mathrm{W}}^{j}, F_{\mathrm{res}}^{j}\right). \tag{6}$$

If F_{res}^{j} is empty, $\mathcal{M}_{\mathrm{W}}$ remains unchanged. Otherwise, U_W abstracts persistent failures into reusable challenge knowledge, including affected dependencies, intervention patterns, operator applicability, and expected-recovery schemas. These abstractions are instantiated in new factual contexts in subsequent rounds rather than by replaying the same failed case.

Checkpoint selection and deployment. CCE-Mobility produces multiple training checkpoints. A checkpoint is selected using the predefined validation protocol and then frozen for held-out testing. Its held-out challenge set is excluded from CCE-Mobility memory evolution and checkpoint selection. Baseline-specific evaluation protocols, including the prequential updates used by Reflexion and Voyager, are described in Section 4.1. At deployment, CCE-Mobility uses only the fixed policy π_θ, observable interaction history, and optional retrieval from $\mathcal{M}_{\mathrm{A}}$; world-side challenge construction and evaluation remain offline.

4 Experiments

Our experiments address three questions: (1) whether CCE-Mobility improves recovery under counterfactual world changes across model backbones and how these gains relate to factual-task quality; (2) whether world-side evolution provides additional benefit beyond agent-only adaptation; and (3) whether CCE-Mobility complements an existing skill-optimization method.

4.1 Experimental Setup

Benchmarks and splits. We evaluate CCE-Mobility on TravelPlanner in its solo-planning setting (Xie et al., 2024) and MobilityBench (Song et al., 2026). Both are augmented with counterfactual recovery challenges that preserve factual history and user goals while changing a task-relevant world dependency. TravelPlanner main results use the held-out test1000 split, with training data used for memory evolution and validation used for checkpoint selection. MobilityBench main results use the fixed historical heldout500 panel for final within-panel comparison under a frozen challenge and evaluator configuration. Mechanism analysis additionally uses a frozen TravelPlanner validation panel of 247 challenges from 149 tasks.

Compared methods and backbones. We compare No Evolution, SkillOpt (Yang et al., 2026b), Reflexion (Shinn et al., 2023), Voyager (Wang et al., 2024), and CCE-Mobility. No Evolution uses the base agent without evolved recovery memory; Reflexion and Voyager are adapted to our recovery setting following their experience-reuse principles, while SkillOpt uses its official implementation. TravelPlanner includes complete five-method comparisons for Qwen3.6-35B-A3B (Qwen Team, 2026a) and Qwen3.8-Max (Qwen Team, 2026b), plus No Evolution–CCE-Mobility comparisons for GPT-5.6-sol (OpenAI, 2026), Claude Opus 4.8 (Anthropic, 2026), and GLM-5.2 (GLM-5 Team et al., 2026). MobilityBench includes complete five-method comparisons for Qwen3.6-35B-A3B and DeepSeek-V4-Flash-0731 (DeepSeek-AI, 2026). Methods within each backbone block share the same frozen challenge set and evaluator configuration.

Table 1: TravelPlanner test1000 results. Counterfactual recovery and official factual-planning metrics are reported separately.

		Counterfactual recovery (%)			Official TravelPlanner (%)		
Backbone	Method	Recovery (+)	Poor Recovery (-)	CF Pass (+)	CS-μ (+)	Hard-μ (+)	TP-F (+)
A. Five-method comparisons							
Qwen3.6 35B-A3B	No Evolution	61.40	31.94	37.59	73.40	7.77	**1.20**
	Reflexion	53.86	45.64	34.46	73.28	7.34	**1.20**
	Voyager	55.31	43.05	35.33	73.31	**7.99**	0.70
	SkillOpt	72.01	21.69	37.13	73.46	5.24	0.60
	CCE-Mobility	**73.12**	**20.90**	**37.79**	**74.48**	7.86	1.00
Qwen3.8-Max	No Evolution	48.67	46.91	37.19	82.21	**34.37**	**6.50**
	Reflexion	50.54	44.71	**38.99**	81.33	32.18	5.60
	Voyager	62.11	30.14	36.66	81.56	31.88	5.50
	SkillOpt	55.79	38.86	36.33	81.75	33.45	5.90
	CCE-Mobility	**68.33**	**23.00**	36.46	**83.65**	34.23	5.80
B. Additional backbone comparisons							
GPT-5.6-sol	No Evolution	65.19	27.21	**2.33**	60.84	0.44	0.10
	CCE-Mobility	**70.12**	**22.22**	2.20	**70.15**	**20.83**	**2.30**
Claude Opus 4.8	No Evolution	53.68	41.78	1.73	60.61	**9.35**	1.30
	CCE-Mobility	**56.41**	**34.80**	**1.80**	**71.90**	8.86	**1.40**
GLM-5.2	No Evolution	35.46	58.75	26.01	72.84	8.65	0.60
	CCE-Mobility	**60.59**	**32.47**	**28.68**	**72.88**	**9.56**	**1.00**

Evaluation protocol. Within each comparison, methods share factual tasks, frozen challenges, interventions, contracts, and evaluator. No Evolution, SkillOpt, and CCE-Mobility use fixed memory snapshots. Reflexion and Voyager use task-level prequential evaluation: all counterfactual cases of a factual task are evaluated before one memory update, which is retained for later tasks. Agents observe only interaction history and tool responses; intervention metadata, recovery contracts, and evaluator outputs remain hidden. TravelPlanner test cases are excluded from CCE-Mobility evolution and checkpoint selection; MobilityBench heldout500 is treated as a fixed historical comparison panel rather than an untouched development-independent test set.

Metrics and statistics. The main tables report three counterfactual recovery metrics and three representative official metrics from each benchmark. The raw Recovery score ranges from 0 to 4 and is normalized to a percentage for reporting. **Poor Recovery** is the percentage of challenges with raw Recovery below 2, equivalently below 50% of the full score, and captures the low-quality tail rather than task failure. **CF Pass** denotes benchmark-specific final acceptance under the counterfactual world. The co-evolution failure predicate is stricter, additionally treating raw Recovery below 3 (75% of the full score) or failed final acceptance as failure (Section 3.3).

For TravelPlanner, we report official commonsense (CS-μ), hard-constraint (Hard-μ), and final-pass (TP-F) metrics. For MobilityBench, we report final-pass rate (FPR), tool selection (TS-R), and decomposition recall (DEC-R). Complete results across all benchmark metrics, together with detailed metric definitions, are provided in the supplementary material. All rate-based quality metrics are reported as percentages. Bold marks the best value within each backbone block, including ties. Paired mechanism analyses use task-cluster bootstrap 95% confidence intervals from 10,000 resamples.

4.2 Main Results

Table 1 shows that CCE-Mobility improves both Recovery and Poor Recovery over No Evolution across all five TravelPlanner backbones. Relative Recovery gains range from 5.1% to 70.9%, while Poor Recovery is reduced by 16.7%–51.0%. In both complete five-method comparisons, CCE-Mobility achieves the highest Recovery and lowest Poor Recovery. Among the selected official metrics, it also achieves the highest CS-μ across all five backbone blocks, while Hard-μ and TP-F show more mixed behavior.

A similar pattern appears on MobilityBench (Table 2). With Qwen3.6, CCE-Mobility improves Recovery by 30.5% and reduces Poor Recovery by 75.0% relative to No Evolution; with DeepSeek, the corresponding improvements are 11.1% and 16.4%. On Qwen3.6, CCE-Mobility achieves the strongest Recovery, Poor Recovery, CF Pass, and FPR, whereas SkillOpt leads TS-R. With

Table 2: MobilityBench heldout500 results. Counterfactual recovery and official factual-interaction metrics are reported separately.

		Counterfactual recovery (%)			**Official MobilityBench (%)**		
Backbone	Method	**Recovery** (+)	**Poor Recovery** (-)	CF Pass (+)	FPR (+)	TS-R (+)	DEC-R (+)
Qwen3.6 35B-A3B	No Evolution	74.16	5.60	98.80	56.00	60.58	**76.99**
	Reflexion	75.05	3.40	96.80	58.00	59.63	75.34
	Voyager	74.61	8.80	99.00	56.20	60.04	76.58
	SkillOpt	84.84	2.20	99.20	57.00	**61.12**	75.34
	CCE-Mobility	**96.75**	**1.40**	**100.00**	**59.60**	61.01	76.16
DeepSeek-V4 Flash-0731	No Evolution	68.09	12.20	**99.60**	45.20	45.14	71.70
	Reflexion	75.25	5.20	98.60	48.00	46.99	70.74
	Voyager	69.96	4.40	98.60	45.80	46.39	69.97
	SkillOpt	**77.88**	**1.60**	**99.60**	49.20	**47.37**	70.52
	CCE-Mobility	75.66	10.20	**99.60**	**51.20**	45.72	**72.59**

Table 3: World–agent co-evolution on the frozen TravelPlanner validation panel (247 challenges from 149 tasks).

Comparison	Baseline Poor Recovery (%)	CCE-Mobility Poor Recovery (%)	Improved	Regressed	Δ Poor Recovery [95% CI] (pp)
No Evolution $\rightarrow$ CCE-Mobility	21.46	**14.57**	20	3	$-6.88\,[-10.61, -3.24]$
Agent-only $\rightarrow$ CCE-Mobility	18.62	**14.57**	14	4	$-4.05\,[-7.35, -0.83]$

DeepSeek, SkillOpt obtains the strongest Recovery, Poor Recovery, and TS-R, while CCE-Mobility achieves the best FPR and DEC-R. Overall, the two benchmarks consistently show improved counterfactual recovery over No Evolution, while factual-task quality does not improve uniformly.

4.3 Analysis of World–Agent Co-Evolution

We isolate the effect of world-side evolution by comparing Agent-only with CCE-Mobility on the frozen TravelPlanner validation panel. Improved and Regressed denote transitions across the Poor Recovery threshold (raw Recovery < 2, i.e., $< 50\%$), distinct from the stricter Fail predicate used for persistent residuals. CCE-Mobility reduces Poor Recovery from 18.62% to 14.57%, a 21.8% relative reduction, with the paired 95% confidence interval excluding zero. As this panel is part of development/validation, we treat it as mechanism evidence rather than an independent test estimate. The supplementary MobilityBench analysis shows only a 1.00-point reduction with an interval spanning zero, so the incremental world-side effect there remains inconclusive.

4.4 Complementarity with Skill Optimization

Using Qwen3.6-35B-A3B, combining CCE-Mobility with SkillOpt increases Recovery from 72.01% to 73.52%, reduces Poor Recovery from 21.69% to 19.89%, and reduces invalid-resource use from 2.79% to 0.70%, while CF Pass remains similar (37.13% vs. 36.73%). Full results are reported in supplementary Table 4. Since paired uncertainty is unavailable, we report this comparison descriptively rather than as evidence of statistically significant improvement.

5 Conclusion

We introduced CCE-Mobility, which constructs trajectory-conditioned counterfactual interventions around dependencies supporting already-made mobility decisions and uses the resulting recovery experience for residual-guided world–agent co-evolution. Across TravelPlanner and MobilityBench, CCE-Mobility consistently improves recovery quality over unevolved agents across the evaluated backbones, although factual-task gains are not uniform. A paired TravelPlanner validation analysis indicates an incremental world-side effect beyond agent-only adaptation, whereas the corresponding MobilityBench mechanism result remains inconclusive. Overall, the results support counterfactual recovery as a useful learning signal while highlighting the need to distinguish recovery quality from final task acceptance and ordinary factual performance.

AI USE STATEMENT

Generative AI tools were used to assist with research ideation and methodology discussion, literature retrieval, code development and debugging, manuscript drafting, and language polishing. Generative AI tools were not used to prove mathematical claims. All scientific figures were manually designed and created by the authors, although AI tools were occasionally used for brainstorming visual presentation ideas.

In addition, LLMs are an explicit component of the proposed methodology for constructing counterfactual interventions and intervention-derived data, as described in the main paper. All AI-assisted code, research ideas, citations, experimental analyses, and manuscript content were reviewed and verified by the authors. The authors take full responsibility for the final content, claims, code, and experimental artifacts of this work.

REFERENCES

Laura Alessandretti, Ulf Aslak, and Sune Lehmann. The scales of human mobility. *Nature*, 587: 402–407, 2020. doi: 10.1038/s41586-020-2909-1.

Anthropic. Claude Opus 4.8 System Card. Anthropic, May 2026.

Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R. James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J. Ramasco, Filippo Simini, and Marcello Tomasini. Human mobility: Models and applications. *Physics Reports*, 734:1–74, 2018. doi: 10.1016/j.physrep.2018.01.001.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13049–13061, 2020.

Hang Ding, Peidong Liu, Junqiao Wang, Ziwei Ji, Meng Cao, Rongzhao Zhang, Lynn Ai, Eric Yang, Tianyu Shi, and Lei Yu. DynaWeb: Model-based reinforcement learning of web agents. *arXiv preprint arXiv:2601.22149*, 2026.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, et al. Agent-World: Scaling real-world environment synthesis for evolving general agent intelligence. *arXiv preprint arXiv:2604.18292*, 2026.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. WebEvolver: Enhancing web agent self-improvement with co-evolving world model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8959–8975, 2025. doi: 10.18653/v1/2025.emnlp-main.454.

GLM-5 Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chengxing Xie, et al. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Jiacheng Guo, Ling Yang, Peter Chen, Qixin Xiao, Yinjie Wang, Xinzhe Juan, Jiahao Qiu, Ke Shen, and Mengdi Wang. GenEnv: Difficulty-aligned co-evolution between LLM agents and environment simulators. *arXiv preprint arXiv:2512.19682*, 2025.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1884–1897, 2021.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Priyanshu Karmakar, Soumyabrata Chaudhuri, Shubhojit Mallick, Manish Gupta, Abhik Jana, and Shreya Ghosh. TripTide: A benchmark for adaptive travel planning under disruptions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40269–40292. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.2002.

Xiaochong Lan, Jie Feng, Jiahuan Lei, Xinlei Shi, and Yong Li. Benchmarking and advancing large language models for local life services. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4566–4577, 2025. doi: 10.1145/3711896.3737196.

Huawei Lin, Peng Li, Jie Song, Fuxin Jiang, and Tieying Zhang. MUSE-Autoskill: Self-evolving agents via skill creation, memory, management, and evaluation. *arXiv preprint arXiv:2605.27366*, 2026.

Jiayu Liu, Cheng Qian, Zhaochen Su, Qing Zong, Shijue Huang, Bingxiang He, and Yi R. Fung. CostBench: Evaluating multi-turn cost-optimal planning and adaptation in dynamic environments for LLM tool-use agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12826–12858. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.584.

Ning Lu, Baijiong Lin, Shengcai Liu, Jiahao Wu, Haoze Lv, Yanbin Wei, Lingting Zhu, Shengju Qian, Xin Wang, Ying-Cong Chen, Qi Wang, and Ke Tang. Policy and world modeling co-training for language agents. *arXiv preprint arXiv:2606.02388*, 2026.

OpenAI. GPT-5.6 System Card. OpenAI Deployment Safety Hub, July 2026.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, et al. SkillOS: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*, 2026.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Qwen Blog, April 2026a.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork. Qwen Blog, August 2026b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023.

Zhiheng Song, Jingshuai Zhang, Chuan Qin, Chao Wang, Chao Chen, Longfei Xu, Kaikui Liu, Xiangxiang Chu, and Hengshu Zhu. MobilityBench: A benchmark for evaluating route-planning agents in real-world mobility scenarios. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 9801–9812, 2026. doi: 10.1145/3770855.3817498.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. Mobile-Agent-E: Self-evolving mobile assistant for complex tasks. In *NeurIPS 2025 Workshop on Scaling Environments for Agents*, 2025.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54590–54613. PMLR, 2024.

Shidong Yang, Ziyu Ma, Tongwen Huang, Yiming Hu, Yong Wang, and Xiangxiang Chu. CoEvolve: Training LLM agents via agent-data mutual evolution. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23015–23036. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1055.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026b.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, et al. Qwen-AgentWorld: Language world models for general agents. *arXiv preprint arXiv:2606.24597*, 2026.