CONSTRAINT-CARRYING SEARCH: WHEN EXPLICIT CONTRACTS IMPROVE WORLD-MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

When does an explicit constraint representation improve world-model search over learned masking, and when is direct symbolic filtering already sufficient? We answer this as a two-axis regime question: specifications range from executable DSLs to mixtures of language and schema fields, while constraint-relevant state ranges from fully observed to latent. The Compiled Admissibility Interface (CAI) canonicalizes entities, dependencies, units, precedence, and progress once, then reuses their identifiers in bounded-continuation supervision, a horizon-conditioned graph compiler, runtime masking, and branch diagnosis; belief disagreement conservatively retains actions at uncertain constraint boundaries. Against a capacity-, protocol-, and search-budget-matched flat encoder on WebArena, WorkArena, a typed-API sandbox, and compositional gridworld, the complete CAI interface improves macro success by $+1.7$ points with paired 95% interval $[+0.8, +2.6]$, with positive point estimates in all four domains, 0.06 fewer violation events per episode, and 2.4 seconds lower planning time. At this operating point CAI removes 34% of expansions relative to unpruned search while maintaining WM-FBR of at least 0.98. The factorial phase diagram localizes the premium: it reaches $+2.5$ points for mixed specifications under partial observability, whereas the DSL-only/full-observation cell lies within the pre-specified practical-equivalence margin of symbolic filtering. In separately stratified boundary samples, method-blind executed replay records feasible-recovery false-negative rates of 2.4% and 2.9% for CAI, compared with 5.2% and 5.9% for Flat. The resulting regime map turns explicit constraint structure into a concrete choice of search interface rather than a universal default.

1 INTRODUCTION

Branch control in world-model planning has a regime structure. At one corner, an executable DSL and fully observed state make direct symbolic filtering sufficient. At the opposite corner, language and schema fields jointly define budgets, prerequisites, types, and precedence while the relevant state is only partially observed. The second regime can benefit from a learned search controller, but it also creates a semantic handoff: the seller, budget, or prerequisite named during continuation supervision must still denote the same object at a later imagined node. If stages encode those objects independently, a plausible world-model transition can be paired with a mask that removes its feasible recovery.

This paper makes specification form and state observability explicit experimental axes. The balanced phase diagram in Figure 1 shows a $+2.5$-point advantage for CAI in the mixed/partial cell and practical equivalence between CAI and Symbolic in the DSL-only/full cell. The same mechanism is tested across web navigation, enterprise workflows, API orchestration, and compositional control, where heterogeneous constraint types must remain aligned as search progresses (Sutton, 1991; Ha & Schmidhuber, 2018; Schrittwieser et al., 2020; Hamrick et al., 2021; Zhou et al., 2024b; Drouin et al., 2024). This organization asks both where an explicit interface earns its cost and which failures account for the difference.

We call a search procedure *constraint-carrying* when one canonical representation supplies entity, dependency, bound, precedence, and progress identities to label construction, node scoring, masking, and diagnosis. CAI implements that abstraction as an admissibility contract whose graph encoding

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.

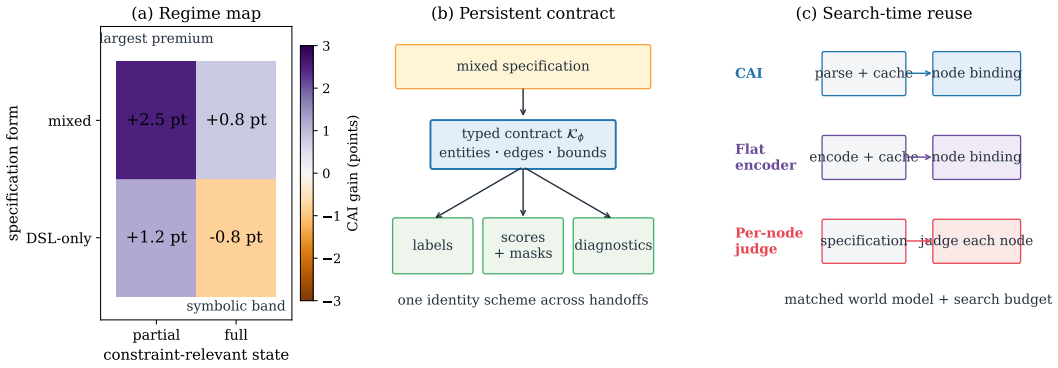


Figure 1: A regime view of constraint-carrying search. (a) The specification-by-observability phase diagram uses the same estimates as Table 2. (b) CAI canonicalizes a mixed specification once and reuses its typed identifiers in supervision, node scoring, masking, progress updates, and diagnosis. (c) The graph is cached at task level and bound to changing belief, progress, horizon, and grounded actions at each search node.

is cached once and fused at each tree node with grounded actions, simulated progress, remaining horizon, and belief. Its uncertainty rule retains actions when constraint-relevant successor features disagree. The set identity in Proposition 1 records the exact thresholded-recall effect of this retention rule; executed outcomes, rather than the identity, establish its planning value.

The primary comparison evaluates the complete interface package rather than generic pruning capacity. Flat receives the same task fields, progress and belief inputs, continuation labels, optimizer, horizon, threshold, search budget, and timing boundary, with only a 3.2% parameter-count difference. CAI gains $+1.7$ success points with paired 95% interval $[+0.8, +2.6]$; all four domain point estimates favor CAI, violation events fall by 0.06 per episode, and macro-average planning time falls by 2.4 seconds. Separately stratified executed replay, component removals, and parser-fidelity tests then locate how the package changes branch survival.

Claim and evidence hierarchy. The paper's central contribution is a regime-conditioned test of an explicit search interface: the matched Flat comparison estimates the value of the complete CAI representation, the 2×2 factorial suite descriptively localizes where that value concentrates, and branch-level and component diagnostics connect the operating-point gain to recovery retention. The technical object enabling this test is a reusable admissibility contract with node-specific horizon scoring and belief-sensitive boundary retention. Together, these elements yield a concrete design rule: carry explicit constraint state through search when semantic or state ambiguity is present; use direct symbolic filtering when both are already resolved.

2 Related Work

Formal control exposes specifications as costs, temporal logic, automata, or shields (Altman, 1999; Achiam et al., 2017; Tessler et al., 2019; Alshiekh et al., 2018; Könighofer et al., 2020; Toro Icarte et al., 2018; 2022; Jothimurugan et al., 2021). Automaton-guided MCTS uses formal progress to shape search (Velasquez et al., 2021); model-based shielding predicts future violations with learned dynamics (Goodall & Belardinelli, 2023); and POMDP shielding integrates reach–avoid rules with online belief-space planning (Sheng et al., 2024). These methods supply executable constraints before planning. CAI addresses the preceding interface: mixed language and schema fields must be canonicalized, bound to changing actions and progress, and reused by both learning and search. The DSL-only/full-observation cell measures the point at which that extra interface collapses to direct symbolic filtering.

Neuro-symbolic action masking learns symbolic grounding for constrained policies (Han et al., 2026), while SEM-CTRL combines Answer Set Grammars with token-level MCTS (Albinhassan et al., 2026). Constraints-of-Thought compiles intent–constraint pairs into executable predicates, and SCULPT injects domain constraints throughout mathematical-workflow MCTS (Alrashedy et al.,

2025; Fang et al., 2026). These close neighbors establish that structured constraints can guide learned search. CAI's object and test differ at the feature level: one task-level contract supplies stable identifiers to continuation labels, horizon scores, simulated progress, runtime masks, and recovery accounting under partial observability.

Language-agent search allocates branches with reasoning, reflection, values, or learned policies (Yao et al., 2023; Hao et al., 2023; Shinn et al., 2023; Zhou et al., 2024a; Li, 2025). AgentOccam changes the browser observation/action interface (Yang et al., 2025), and Prune4Web reduces DOM grounding through synthesized filters (Zhang et al., 2026); CAI operates after candidate grounding and controls imagined continuations. The matched Flat encoder is therefore the closest representation control: it receives the same raw specification, progress, belief, action, horizon, supervision, and search budget without the typed graph interface. Program synthesis supplies a compile-then-execute analogy (Gulwani et al., 2017; Ellis et al., 2021), but CAI's compiled object remains a belief-conditioned search state rather than a deterministic task program.

3 Problem Formulation

3.1 Belief-Space Planning with Constraint Progress

Let $x_t \in \mathcal{X}$ be the latent environment state, o_t the observation, and $b_t \in \Delta(\mathcal{X})$ the planner's belief after the executed history. A frozen world model provides transition and observation factors $f_\theta(x' \mid x, a)$ and $g_\theta(o \mid x')$ for candidate action $a \in \mathcal{A}(o_t)$. The task specification ϕ induces a constraint-progress state q_t containing satisfied prerequisites, consumed resources, and active bounds. During one root search at time t, node ν denotes an imagined history with progress $q_{\nu,t}$, decoded observation o_ν, and weighted particle belief $b_{\nu,t}$; the root has $(q_{\mathrm{root},t}, b_{\mathrm{root},t}) = (q_t, b_t)$. Executed transitions and benchmark evaluators remain distinct from these model states.

Definition 1 (Constraint graph)**.** *For specification ϕ, the graph $\mathcal{G}_\phi = (\mathcal{V}, \mathcal{E}, \mathcal{C})$ contains typed entity or subgoal nodes $\mathcal{V}$, directed prerequisite edges $\mathcal{E}$, and annotations $\mathcal{C}$ for resource bounds, type restrictions, ordering requirements, negation, and soft priorities.*

Definition 2 (Admissibility contract)**.** *For specification ϕ, the admissibility contract $\mathcal{K}_\phi = (\mathcal{G}_\phi, \delta_\phi, \Pi_\phi)$ combines the typed graph, a deterministic progress update $q_{t+1} = \delta_\phi(q_t, a_t, o_{t+1})$, and the canonicalization and precedence map Π_ϕ. A planner carries $\mathcal{K}_\phi$ when label construction, compiler inputs, runtime masks, and diagnostics address the same normalized entities, edges, units, precedence outcomes, and progress variables.*

The contract is explicit state shared by the parser, continuation labeler, compiler, and diagnostics. Every node, edge, annotation, and progress coordinate has a canonical identifier; runtime exclusions and audit outcomes are recorded against those identifiers rather than against newly generated text. Stable identifiers enforce type and precedence invariants and make every hard exclusion addressable to a bound, dependency, or progress variable.

3.2 Scores, Thresholding, and Feasible-Branch Recall

At search node ν, CAI outputs an *admissibility score* $M^h_{\nu,t}(a) \in [0, 1]$ for grounded action a at remaining horizon h. This score serves as a ranking surrogate rather than a calibrated probability. The thresholded mask retains action a when $M^h_{\nu,t}(a) \geq \varepsilon$. If that mask is empty, execution adds the maximum-score action, breaking ties by canonical action index, so search always receives a candidate. FBR excludes this deterministic liveness fallback and therefore isolates the compiler's threshold decisions; executed success and violations include the fallback.

For a node state and progress pair $(x, q_{\nu,t})$, let $\mathcal{F}^h_\nu(x)$ be the actions for which at least one constraint-satisfying continuation exists from x within horizon h.

Definition 3 (Feasible-branch recall)**.**

$$\mathrm{FBR}^h_\nu(M, \varepsilon; x) = \frac{\left|\{a \in \mathcal{F}^h_\nu(x) : M^h_{\nu,t}(a) \geq \varepsilon\}\right|}{|\mathcal{F}^h_\nu(x)|}. \tag{1}$$

For $\mathcal{F}^h_\nu(x) = \varnothing$, the node–state pair contributes neither numerator nor denominator to aggregate FBR.

True $\mathcal{F}^h_\nu(x)$ is unavailable online. For particle $x^{(i)}_{\nu,t}$, WM-FBR substitutes the set $\widehat{\mathcal{F}}^h_{\nu,i,\mathrm{wm}}$ returned by bounded continuation in the frozen world model. The reported diagnostic is the particle-weighted micro-average over feasible (ν, i, a) tuples, so the feasible set and score always refer to the same simulated node. Executed success and hard violations remain the primary evidence; WM-FBR diagnoses planner-internal pruning behavior. Relative expansions are the number of expanded nodes divided by No Pruning under the same N-iteration budget.

4 Constraint-Carrying Search

4.1 Typed Contract Construction and Bounded Continuation Labels

The parser maps structured fields deterministically into $\mathcal{K}_\phi$: endpoint signatures become entity types, preconditions become edges and progress transitions, and range or rate constraints become annotations. Language fields pass through a frozen encoder and schema-constrained span and relation decoders that emit entity mentions, typed relations, numeric bounds with units, negation, and resource tags; planner outcomes never update this encoder. A deterministic canonicalizer instantiates Π_ϕ by resolving aliases against the domain schema, converting numeric values to canonical units, checking action arity and argument types, merging duplicate mentions, and applying structured-over-language precedence. Each emitted object receives a canonical identifier used unchanged by the labeler, compiler, progress updater, and audit logger. Unattached spans, conflicting language-only parses, and schema-invalid tuples become soft priorities rather than hard exclusions.

A grounded action at node ν is represented as $e_\nu(a) = [e_{\mathrm{op}}; e_{\mathrm{target}}; e_{\mathrm{args}}; e_{\mathrm{obs}}]$: the canonical operation or endpoint, bound target identity, normalized argument values, and local predicted-observation features. This binding handles variable action sets without assigning graph-node positions to actions. Constraint nodes attend to matching action fields by canonical type and identity; unmatched fields receive a learned null binding.

The continuation labeler solves a finite planner-side problem. For each (x, q, a, h), it orders successors by model probability, retains the smallest prefix whose cumulative mass reaches validation-selected ρ, and removes transitions below validation-selected support floor $p_{\min}$. A label is positive when any retained successor path reaches a terminal state satisfying all hard constraints within h; a terminal state is positive exactly when its hard constraints are satisfied. Repeated subproblems are memoized by the latent state quantized at fixed resolution Δ_x, the complete constraint-progress state, remaining horizon, and retained-support signature. Successor and action ties use canonical indices. The values $(\rho, p_{\min}, \Delta_x)$ are selected without test outcomes and then shared by CAI and Flat. Thus, labels are exact for the finite enumerated support, horizon, and action abstraction actually solved, while the environment audit in Section 6 measures branch feasibility using executed replay. For a training node belief, particle i receives target $y^h_{\nu,i}(a) = \mathbf{1}[a \in \widehat{\mathcal{F}}^h_{\nu,i,\mathrm{wm}}]$. Weighting these state-level targets by the particle weights makes supervision and runtime scoring refer to the same node distribution without treating the score as a calibrated feasibility probability.

4.2 Horizon Compiler and Asymmetric Training

A four-layer graph neural network first produces cached contract embeddings $E_\phi = \mathrm{GNN}_\psi(\mathcal{K}_\phi)$. At node ν, the belief is a weighted particle set $b_{\nu,t} = \{(x^{(i)}_{\nu,t}, w^{(i)}_{\nu,t})\}_{i=1}^B$. Along an edge, particles are propagated through f_θ and reweighted by the decoded observation, $\widetilde{w}^{(i)}_{\nu,t} \propto w^{(i)}_{\mathrm{pa}(\nu),t} g_\theta(o_\nu \mid x^{(i)}_{\nu,t})$; normalized categorical resampling returns B equally weighted particles. The same update uses the executed observation when the next root is formed. Cross-attention fuses the cached contract, current progress, particle summary, grounded action embedding, and remaining horizon:

$$M^h_{\nu,t}(a) = \sigma(\mathcal{R}_\psi(E_\phi, q_{\nu,t}, b_{\nu,t}, e_\nu(a), h)), \quad a \in \mathcal{A}(o_\nu),\ h = 1, \ldots, H. \tag{2}$$

Simulated progress $q_{\nu,t}$ is updated along every model edge with the same δ_ϕ used after execution, and $\mathcal{A}(o_\nu)$ is grounded from that node's decoded observation. Thus, task-level parsing and graph encoding occur once, while the light readout $\mathcal{R}_\psi$ binds node-specific belief, progress, and actions without reinterpreting the specification. False negatives remove potentially successful branches,

CAI planning step at time t.
1. Parse ϕ once to obtain $\mathcal{K}_\phi$, cache E_ϕ, and initialize root progress and belief (q_t, b_t).
2. For each of N PUCT iterations, propagate $(q_{\nu,t}, b_{\nu,t})$ to the selected leaf, ground $\mathcal{A}(o_\nu)$, and evaluate Eq. 2 using cached E_ϕ.
3. Compute $u_{\nu,t}$, apply Eq. 5, retain threshold-passing or uncertainty-retained actions, and expand with the unchanged world model and backup rule.
4. Execute the selected root action; the new observation updates b_{t+1} and $q_{t+1} = \delta_\phi(q_t, a_t, o_{t+1})$, while E_ϕ remains cached for the task.

Figure 2: Constraint-carrying execution. The task-level graph representation is cached once; a node-local readout binds it to simulated progress, belief, horizon, and the current grounded action set. Executed observations update belief and progress without changing canonical constraint identities.

whereas false positives consume search. We therefore train with asymmetric binary cross-entropy,

$$\mathcal{L}_\psi = -\sum_{\nu,h}\sum_{i=1}^{B} w_{\nu,t}^{(i)} \sum_a \big[\lambda_+ y_{\nu,i}^h(a) \log M_{\nu,t}^h(a) + \lambda_-(1 - y_{\nu,i}^h(a)) \log(1 - M_{\nu,t}^h(a))\big], \qquad \lambda_+/\lambda_- = 5. \tag{3}$$

4.3 Belief-Conditioned Boundary Retention

Let $\bar{z}(x')$ collect constraint-relevant successor features: budget counters normalized by their contract bounds, order and type-validity indicators, and termination status. At node ν, disagreement is total predictive covariance over belief support and world-model successors,

$$u_{\nu,t}(a) = \mathrm{tr}\big[\mathrm{Cov}_{x\sim b_{\nu,t},\, x'\sim f_\theta(\cdot|x,a)}\big(\bar{z}(x')\big)\big]. \tag{4}$$

By total covariance, this quantity sums between-particle state ambiguity and within-model transition variability in the normalized features that can change admissibility. It is a node-local decision heuristic and therefore has no horizon superscript. The relaxed score is

$$M'^h_{\nu,t}(a) = \max\big(M_{\nu,t}^h(a), \mathbf{1}[u_{\nu,t}(a) > \tau]\alpha\big). \tag{5}$$

Proposition 1 (Threshold-retention identity)**.** *For any fixed nonempty $\mathcal{F}_\nu^h(x)$, if $\alpha \geq \varepsilon$, then*

$$\mathrm{FBR}_\nu^h(M', \varepsilon; x) - \mathrm{FBR}_\nu^h(M, \varepsilon; x) = \frac{\big|\{a \in \mathcal{F}_\nu^h(x) : M_{\nu,t}^h(a) < \varepsilon,\ u_{\nu,t}(a) > \tau\}\big|}{|\mathcal{F}_\nu^h(x)|} \geq 0. \tag{6}$$

This set identity requires no independence assumption and states only the thresholded-recall effect of adding uncertain actions. Expansion cost, executed success, and constraint behavior are measured empirically.

4.4 Compile Once, Reuse Across Search

For graph width d, L message-passing layers, B belief particles, and model-feature cost C_f, task-level encoding costs $O(L(|\mathcal{V}| + |\mathcal{E}|)d^2)$ time and $O(|\mathcal{V}|d)$ memory. A node with $|\mathcal{A}_\nu|$ grounded actions adds $O(|\mathcal{A}_\nu|(Hd + BC_f))$ readout and uncertainty work and stores $O(H|\mathcal{A}_\nu|)$ scores. The flat mask uses the same node-local action binding, particle summary, horizon outputs, and cache/readout schedule but replaces the canonical graph encoder with a task-serialization encoder; its parameter count differs by only 3.2%. A per-node judge instead incurs $O(N|\mathcal{A}|C_j)$ feasibility inference. On WebArena this distinction appears directly in end-to-end time: CAI uses 84.1 seconds, the flat mask 87.6 ± 2.8, and the LLM node filter 156.2 seconds per episode.

5 Experiments

5.1 Questions and Benchmarks

The evaluation follows the regime structure introduced in Figure 1. First, a balanced specification-by-observability factorial suite asks where the complete explicit interface improves over learned or

symbolic filtering. Second, a matched representation comparison tests whether CAI changes executed outcomes and search cost beyond a flat learned mask with the same inputs and training signal. Third, component removal and executed replay examine whether the operating-point difference is accompanied by fewer false-negative recovery prunes.

The public-environment evaluations are derived constrained suites rather than the unconstrained benchmark distributions. WebArena contributes 200 held-out navigation tasks augmented before execution with 2–5 applicable budget, ordering, or seller constraints (Zhou et al., 2024b). WorkArena contributes 180 held-out ServiceNow tasks spanning list, form, retrieval, and cross-application workflows, each augmented with 2–5 applicable budget/resource, prerequisite/order, or type constraints (Drouin et al., 2024). Environment state is reset from the benchmark snapshot after every episode. Official validators score the original task goal; a separate deterministic contract monitor records violations of the added constraints. The primary Typed API test contains 300 tasks over 50 typed endpoints. A separate factorial suite contains four disjoint 120-task cells crossing mixed versus DSL-only specifications with partial versus full constraint-relevant observability; it is not included in the 300-task aggregate. Compositional gridworld contains 150 tasks on a 16×16 map with resource, ordering, and zone constraints. WebArena and WorkArena provide distinct public interaction stacks; the separate Typed API factorial suite provides controlled descriptive regime contrasts under fixed action semantics.

Constraint augmentation first enumerates the domain-applicable inventory, removes exact duplicates and predicates already entailed by the official validator, and rejects combinations inconsistent with the initial state or official goal. A fixed pre-execution draw then selects 2–5 constraints from the remaining inventory. Base-task identity and canonical constraint composition define split groups, so neither a base task nor a paraphrase of its constraint graph can cross from compiler/parser training or threshold selection into test. World models are frozen before compiler training; label-support, threshold, and model choices are fixed on training/validation partitions before either the primary or factorial test suites are evaluated.

5.2 Baselines and Matched Protocol

No Pruning exposes every grounded action to PUCT. Symbolic evaluates observed type, bound, and prerequisite predicates directly and retains an action whenever unresolved language or hidden state prevents a formal exclusion. Penalized PUCT leaves the action set intact and subtracts a validation-tuned node-local predicted-violation penalty from the selection score. The Llama-3.1-8B-Instruct filter (Dubey et al., 2024) receives the specification, progress summary, candidate action, and predicted successor and returns a temperature-zero retain/prune decision independently at each expanded node. Oracle Mask is a privileged lookup into bounded-continuation labels.

Flat is the decisive representation control. It serializes the same task text and structured fields, uses the same grounded-action embeddings, progress vector, belief particles, horizon readout, and cache schedule, but replaces canonical graph nodes and identifiers with a learned task-sequence encoder. The progress values are supplied in a fixed positional serialization, so Flat observes the same state variables without receiving their entity or edge identities as intermediate outputs. Flat and CAI share the frozen world model, continuation labels, train/validation tasks, Adam optimizer, 256-dimensional width, $H = 8$, threshold-selection procedure, seeds, and end-to-end timing boundary; their trainable parameter counts differ by 3.2%.

All planners use $N = 256$ PUCT iterations, $c_{\mathrm{puct}} = 1.5$, $H = 8$, $\varepsilon = 0.15$, validation-calibrated τ, and $\alpha = 0.6$. CAI trains for 50 epochs with Adam (Kingma & Ba, 2015) at 3×10^{-4}. DOM-T5, a T5-large transition backbone (Raffel et al., 2020), is shared by WebArena and WorkArena; API-GPT2 uses a GPT-2 Medium transition backbone (Radford et al., 2019); Grid-MLP-6 is the gridworld transition ensemble. WebArena and WorkArena are used under Apache-2.0; the T5, GPT-2, and Llama 3.1 checkpoints are used under their Apache-2.0, MIT, and Llama 3.1 Community License terms, respectively. Each method/domain/seed run uses one NVIDIA A100 80GB GPU under CUDA 12.4; eight workers run in parallel on the same cluster. Seeds are indexed 0–4 and vary initialization, task order, and environment stochasticity.

Success is the benchmark evaluator's binary episode outcome. A violation event is each executed budget, order, type, or zone breach; Table 1 reports events per complete benchmark episode and therefore permits multiple events in one episode. A method's practical utility is read from the

ordered pair of official success and violation-event rate, preserving the public task outcome while exposing behavior under the added specification. A sampled-set recovery loss occurs when a mask removes every environment-feasible continuation among the six candidates audited at a sampled node. Per-episode wall-clock starts before specification parsing and ends at task termination after all root searches and executed actions, including model inference, filtering, and search. Operating metrics report mean $\pm$ seed-level standard error over five seeds. Success columns report point estimates from all completed task–seed units; uncertainty for the central success comparisons uses a paired cluster bootstrap that resamples base tasks with all five seed outcomes kept together, with 10,000 replicates. Macro-aggregates weight each benchmark family equally. Oracle Mask is a cached lookup into privileged bounded-continuation labels: offline label-generation cost is excluded, so its runtime is a ceiling and is not part of non-oracle fairness claims.

6 RESULTS

6.1 REGIME MAP AND CROSS-DOMAIN REPRESENTATION TEST

The phase diagram has a positive mixed/partial corner and a symbolic DSL/full corner (Table 2); the four-family test then asks whether CAI improves the practical operating point across heterogeneous constrained workloads. Table 1 jointly reports official success, added-constraint violations, search expansion, model-relative branch recall, and end-to-end time. Relative to the matched flat mask, the complete CAI interface gains $+1.7$ success points with paired 95% CI $[+0.8, +2.6]$, lowers violations by 0.06 events per episode, and reduces time by 2.4 seconds. The per-domain success intervals are WebArena $[+0.2, +3.6]$, Typed API $[-0.2, +2.3]$, gridworld $[0.0, +3.7]$, and WorkArena $[+0.1, +3.7]$. All four domain point estimates favor CAI, although the Typed API and gridworld intervals include zero under the stated task-cluster bootstrap. Compared with No Pruning, CAI improves macro-average success by $+7.3$ points while reducing expansions by 34%. Against the operational alternatives in Appendix Table 5, CAI also improves macro success by 2.5 points over Penalized PUCT and 3.3 over the per-node LLM filter while saving 39.2 and 50.1 seconds per episode, respectively.

The public benchmarks provide complementary evidence. On WebArena, CAI reaches 45.6% success while reducing violations from 0.24 ± 0.03 to 0.18 ± 0.02 and saving 3.5 seconds versus the flat mask. On WorkArena's 900 task–seed units, CAI obtains 39.3% success, 0.20 ± 0.02 violations, and 99.7 ± 3.1 seconds, improving the strongest learned comparator on all three axes. The smaller typed-API margin is consistent with formal schema fields already supporting strong direct filtering; the larger gridworld margin is consistent with composed resource and ordering state requiring alignment across horizon.

6.2 FACTORIAL CELL EVIDENCE

Table 2 reports the separate balanced factorial suite. CAI gains $+2.5$ points over the strongest non-oracle comparator in the mixed/partial cell, $+0.8$ in mixed/full, and $+1.2$ in DSL-only/partial. In DSL-only/full, Symbolic reaches 60.2 and CAI 59.3; the paired difference is -0.8 points with 95% CI $[-1.7, +0.1]$, contained within the pre-specified ± 2.0-point equivalence margin. The margin corresponds to 12 of the 600 task–seed outcomes in a cell and defines practical, rather than exact, equality. Using Flat as the common reference, the displayed raw counts give CAI premiums of $+2.5, +1.0, +1.2$, and -0.3 points across mixed/partial, mixed/full, DSL/partial, and DSL/full. Their descriptive main contrasts are $+1.3$ points for mixed specification form and $+1.5$ for partial observability; the cellwise paired intervals in Appendix Table 8 provide the inferential comparisons.

6.3 EXECUTED-RECOVERY TEST

The audit samples each method's boundary-heavy decisions under common constraint-type and boundary-distance quotas, replays six candidate continuations per sampled node in the corresponding executed environment, and applies benchmark validators without access to method identity, score, or mask. For learned masks, distance is $|M^h_{\nu,t}(a) - \varepsilon|$; for Symbolic, the boundary stratum contains actions whose status depends on one unresolved or threshold-equality predicate. Because the methods induce different boundary-node pools, these estimates characterize each method at its own identically

Table 1: Four-family matched-budget results. Success is the percentage over all task–seed units; paired 95% intervals for CAI–Flat are reported in text. Other nontrivial cells show mean ± seed SE. Violations are executed events per episode. Flat is the representation control; secondary per-node and privileged-ceiling results appear in Appendix Table 5.

Method	Success (%)	Viol./ep.	Rel. exp.	WM-FBR	Time (s)
WebArena constrained tasks					
No Pruning	38.4	0.42 ± 0.04	1.00	1.00	127.3 ± 4.1
Symbolic	41.7	0.31 ± 0.03	0.78 ± 0.02	0.94 ± 0.010	98.4 ± 3.1
Flat learned mask	43.7	0.24 ± 0.03	0.69 ± 0.02	0.953 ± 0.008	87.6 ± 2.8
CAI	45.6	0.18 ± 0.02	0.66 ± 0.02	0.98 ± 0.006	84.1 ± 2.6
Typed API sandbox					
No Pruning	52.1	0.35 ± 0.03	1.00	1.00	43.8 ± 1.5
Symbolic	58.3	0.19 ± 0.02	0.72 ± 0.02	0.97 ± 0.008	31.2 ± 1.1
Flat learned mask	58.7	0.16 ± 0.02	0.65 ± 0.02	0.975 ± 0.006	30.4 ± 1.0
CAI	59.8	0.12 ± 0.02	0.63 ± 0.02	0.99 ± 0.005	29.6 ± 0.9
Compositional gridworld					
No Pruning	44.7	0.51 ± 0.05	1.00	1.00	18.2 ± 0.7
Symbolic	48.9	0.34 ± 0.04	0.74 ± 0.02	0.96 ± 0.009	13.7 ± 0.5
Flat learned mask	49.6	0.25 ± 0.03	0.71 ± 0.02	0.952 ± 0.008	13.1 ± 0.5
CAI	51.5	0.19 ± 0.02	0.69 ± 0.02	0.98 ± 0.006	12.3 ± 0.4
WorkArena constrained tasks					
No Pruning	31.8	0.49 ± 0.04	1.00	1.00	149.2 ± 4.7
Symbolic	35.2	0.37 ± 0.03	0.80 ± 0.02	0.934 ± 0.010	116.8 ± 3.8
Flat learned mask	37.4	0.27 ± 0.03	0.70 ± 0.02	0.952 ± 0.008	104.3 ± 3.5
CAI	39.3	0.20 ± 0.02	0.67 ± 0.02	0.982 ± 0.006	99.7 ± 3.1

Table 2: Separate Typed API 2×2 suite (success percentage over 600 task–seed units per cell). Each cell has 120 disjoint held-out tasks evaluated with five paired seeds; it is not part of the 300-task primary Typed API aggregate in Table 1. Paired intervals appear in Appendix Table 8.

Specification / observation	Symbolic	Flat mask	CAI
Mixed / partial	57.8	59.3	61.8
Mixed / full	60.0	59.8	60.8
DSL-only / partial	55.2	56.0	57.2
DSL-only / full	60.2	59.7	59.3

stratified operating boundary; the environment-feasible denominators are determined after replay. Table 3 shows that CAI's environment-grounded false-negative branch rate is $2.4\%[1.5, 3.7]$ on WebArena and $2.9\%[1.8, 4.5]$ on WorkArena, compared with $5.2\%[3.8, 7.0]$ and $5.9\%[4.3, 7.9]$ for the matched flat mask. Under identical stratum quotas, CAI also has the fewest sampled-set recovery losses and sampled nodes with at least one hard violation among the six candidate replays in both environments. These method-blind, within-method diagnostics reproduce the WM-FBR ordering with executed transitions; the paired end-to-end comparison remains Table 1.

Table 3: Environment-grounded recovery audit. FNR is the fraction of environment-feasible candidate continuations removed within method-specific, identically stratified boundary samples, with Wilson 95% intervals. Sampled-set losses—removing every feasible continuation among the six audited candidates—and replay violations—at least one hard violation among those six replays—are incidences per 100 sampled nodes (mean $\pm$ seed SE), distinct from full-evaluation violation events per episode in Table 1.

Domain	Method	Env. FNR [95% CI]	Set loss/100	Replay viol./100	Success (%)
WebArena	Symbolic	$8.6\%[6.8, 10.8]$	4.7 ± 0.7	5.8 ± 0.6	41.7
	Flat mask	$5.2\%[3.8, 7.0]$	2.6 ± 0.5	4.1 ± 0.5	43.7
	CAI	$2.4\%[1.5, 3.7]$	1.2 ± 0.4	3.1 ± 0.5	45.6
WorkArena	Symbolic	$9.2\%[7.2, 11.7]$	5.1 ± 0.7	6.2 ± 0.7	35.2
	Flat mask	$5.9\%[4.3, 7.9]$	3.0 ± 0.6	4.6 ± 0.6	37.4
	CAI	$2.9\%[1.8, 4.5]$	1.6 ± 0.4	3.5 ± 0.5	39.3

6.4 Mechanism Decomposition

The component results separate the complete interface from two recovery-protection mechanisms. Removing typed graph structure while retaining only a pooled contract summary costs 4.8 success points and lowers WM-FBR from 0.98 to 0.91; this is not the Flat control, which instead learns a capacity-matched encoder directly from the full raw serialization and trails CAI by 1.7 points. Removing belief relaxation costs 2.3 points and lowers WM-FBR to 0.89, while replacing the $5{:}1$ asymmetric loss with ordinary binary cross-entropy costs 3.1 points and lowers WM-FBR to 0.86 (Appendix Table 6). The independent parser audit further records atomic F1 of 0.930 under paraphrase and 0.900 under compositional holdout (Appendix Table 9). Together, the matched Flat test compares the complete graph-and-canonicalization package with a sequence encoder, while the component removals locate additional sensitivity to typed topology, uncertainty retention, and false-negative weighting.

7 Discussion

The phase diagram identifies a practical boundary: CAI improves the strongest observed operating point when either the specification or relevant state is ambiguous, while direct symbolic filtering reaches the same practical band when both are explicit. Across the four-family test, CAI outperforms the capacity-matched Flat control in success, violation-event rate, and runtime.

The matched comparison estimates the complete CAI package relative to Flat; ablations, parser fidelity, and executed replay localize its components and branch-survival behavior. The evaluated regime uses fixed constraint vocabularies, five paired seeds, derived constrained public-environment suites, and candidate sets of at most 100 actions.

8 Conclusion

Constraint-carrying search treats specification form and state observability as design variables for world-model branch control. CAI turns mixed language and schema constraints into a task-level graph whose identifiers remain available as belief, progress, horizon, and grounded actions change. Across four workload families, this complete interface gains $+1.7$ success points over the matched Flat encoder while lowering violation events and runtime, and its boundary samples contain fewer feasible-recovery false negatives in both public environments. The 2×2 factorial suite places the largest premium in the mixed/partial regime and direct symbolic filtering in the same practical band when specifications and relevant state are fully explicit. Explicit contracts therefore earn their role where planning must preserve constraint structure through semantic or observational ambiguity.

REPRODUCIBILITY STATEMENT

Sections 4–6 and Appendices A–H define the planner operations, label semantics, timing boundary, evaluation populations, raw paired outcomes, and environment-replay procedure used for the reported comparisons.

AI USE STATEMENT

Generative AI tools assisted with language editing, organization, presentation, bibliographic consistency checks, and formatting during manuscript preparation. This preparation pass did not generate data, conduct experiments, or introduce new methods or results.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Mohammad Albinhassan, Pranava Madhyastha, and Alessandra Russo. SEM-CTRL: Semantically controlled decoding. *Transactions on Machine Learning Research*, 2026. URL https://mlanthology.org/tmlr/2026/albinhassan2026tmlr-semctrl/.

Kamel Alrashedy, Vriksha Srihari, Zulfiqar Zaidi, Ridam Srivastava, Pradyumna Tambwekar, and Matthew Gombolay. Constraints-of-thought: A framework for constrained reasoning in language-model-guided search. *arXiv preprint arXiv:2510.08992*, 2025. URL https://arxiv.org/abs/2510.08992.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 2669–2678, 2018. doi: 10.1609/aaai.v32i1.11797.

Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, 1999. doi: 10.1201/9781315140223. URL https://doi.org/10.1201/9781315140223.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662. PMLR, 2024. URL https://proceedings.mlr.press/v235/drouin24a.html.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B Tenenbaum. DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pp. 835–850, 2021. doi: 10.1145/3453483.3454080. URL https://doi.org/10.1145/3453483.3454080.

Qitong Fang, Haotian Li, and Xu Wang. SCULPT: Constraint-guided pruned MCTS that carves efficient paths for mathematical reasoning. *arXiv preprint arXiv:2601.12842*, 2026. URL https://arxiv.org/abs/2601.12842.

Alexander W. Goodall and Francesco Belardinelli. Approximate model-based shielding for safe reinforcement learning. In *ECAI 2023*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pp. 883–890. IOS Press, 2023. doi: 10.3233/FAIA230357.

Sumit Gulwani, Oleksandr Polozov, and Rishabh Singh. Program synthesis. *Foundations and Trends in Programming Languages*, 4(1–2):1–119, 2017. doi: 10.1561/2500000010. URL https://doi.org/10.1561/2500000010.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2451–2463, 2018. URL https://papers.neurips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.

Jessica B. Hamrick, Abram L. Friesen, Feryal Behbahani, Arthur Guez, Fabio Viola, Sims Witherspoon, Thomas Anthony, Lars Holger Buesing, Petar Veličković, and Théophane Weber. On the role of planning in model-based deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.04021.

Shuai Han, Mehdi Dastani, and Shihan Wang. Neuro-symbolic action masking for deep reinforcement learning. *arXiv preprint arXiv:2602.10598*, 2026. URL https://arxiv.org/abs/2602.10598.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. Compositional reinforcement learning from logical specifications. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10026–10039, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/531db99cb00833bcd414459069dc7387-Abstract.html.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Bettina Könighofer, Florian Lorber, Nils Jansen, and Roderick Bloem. Shield synthesis for reinforcement learning. In *Leveraging Applications of Formal Methods, Verification and Validation: Verification Principles*, volume 12476 of *Lecture Notes in Computer Science*, pp. 290–306. Springer, 2020. doi: 10.1007/978-3-030-61362-4_16.

Yang Li. Policy guided tree search for enhanced LLM reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35710–35730. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25bv.html.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yuanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020. doi: 10.1038/s41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

Shili Sheng, David Parker, and Lu Feng. Safe POMDP online planning via shielding. In *IEEE International Conference on Robotics and Automation*, pp. 126–132. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610195.

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. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract.html.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377. URL https://doi.org/10.1145/122344.122377.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2107–2116. PMLR, 2018. URL https://proceedings.mlr.press/v80/icarte18a.html.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022. doi: 10.1613/jair.1.12440. URL https://doi.org/10.1613/jair.1.12440.

Alvaro Velasquez, Brett Bissey, Lior Barak, Andre Beckus, Ismail Alkhouri, Daniel Melcer, and George K. Atia. Dynamic automaton-guided reward shaping for Monte Carlo tree search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12015–12023, 2021. doi: 10.1609/aaai.v35i13.17427.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. AgentOccam: A simple yet strong baseline for LLM-based web agents. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oWdzUpOlkX.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Jiayuan Zhang, Kaiquan Chen, Zhihao Lu, Enshen Zhou, Qian Yu, and Jing Zhang. Prune4Web: DOM tree pruning programming for web agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34710–34718, 2026. doi: 10.1609/aaai.v40i41.40772. Issue 41.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62138–62160. PMLR, 2024a. URL https://proceedings.mlr.press/v235/zhou24r.html.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=oKn9c6ytLx.

A Proof and Label Construction

A.1 Proof of Proposition 1

Proof. Equation 5 is elementwise, so $M'^{h}_{\nu,t}(a) \geq M^{h}_{\nu,t}(a)$ for every action. When $\alpha \geq \varepsilon$, every action satisfying $u_{\nu,t}(a) > \tau$ is retained after relaxation. The feasible actions retained by M' but not by M are therefore exactly $\{a \in \mathcal{F}^{h}_{\nu}(x) : M^{h}_{\nu,t}(a) < \varepsilon,\ u_{\nu,t}(a) > \tau\}$. Dividing the cardinality of this rescued set by $|\mathcal{F}^{h}_{\nu}(x)|$ gives Eq. 6; nonnegativity follows immediately. □

A.2 Stochastic Successor Enumeration

At each continuation node, the labeler sorts model successors by probability and retains the shortest prefix reaching the validation-selected cumulative mass. It then removes successors below the support floor and renormalizes the retained probabilities. A branch stops on the first hard violation, on a satisfying terminal state, or when its remaining horizon reaches zero. Positive labels require at least one satisfying path in retained support; transition probabilities affect support membership but do

not convert existence into an expectation. Memoization keys contain the latent-state quantization, complete constraint-progress state, remaining horizon, and support signature, preventing accidental reuse across distinct bounds or stochastic truncations.

B Configurations and Contract Mechanisms

The configuration fixes a common compiler width, training schedule, search budget, and validation protocol while changing only domain-specific dynamics and action abstractions.

Table 4: Configuration summary. All non-oracle methods share the listed world model and action abstraction within a domain.

Component	WebArena / WorkArena	Typed API	Gridworld
World model	DOM-T5 (T5-large)	API-GPT2 (GPT-2 Medium)	Grid-MLP-6 ensemble
Contract parser	schema rules + frozen language span encoder	same	schema rules
Contract state	graph, progress, precedence	same	same
Belief input	DOM/state particles	schema-state particles	discrete-state particles
Candidate actions	grounded browser actions	50 typed endpoints	moves/interactions
CAI compiler	cached 4-layer typed GNN + cross-attention readout, width 256	same	same
Flat control	256-dimensional serialized-task encoder + same node-local readout	same	same
Search	PUCT, $N = 256$, $H = 8$	same	same
Thresholds	$\varepsilon = .15$, $\alpha = .6$; $\tau, \rho, p_{\min}, \Delta_x$ on validation	same	same
Training	Adam, 50 epochs, 3×10^{-4}	same	same
Primary test tasks	200 / 180	300	150
Seeds	0, 1, 2, 3, 4	same	same

Table 5 shows that CAI's advantage is not specific to Flat: node-local penalties and repeated language judging consume more time while attaining lower success. Oracle Mask provides a privileged ceiling rather than a fairness comparator.

Table 5: Secondary matched-budget comparators and privileged ceilings (mean $\pm$ SE, five seeds). Oracle Mask is a cached lookup into bounded-continuation labels; its offline label-generation cost is excluded, so it provides a runtime ceiling outside the non-oracle fairness comparison.

Method	Success	Viol.	Rel. exp.	WM-FBR	Time (s)
WebArena constrained tasks					
Penalized PUCT	43.1 ± 1.4	0.28 ± 0.03	0.82 ± 0.02	0.96 ± 0.009	142.7 ± 4.6
LLM node filter	42.5 ± 1.6	0.25 ± 0.04	0.71 ± 0.03	0.93 ± 0.012	156.2 ± 5.7
Oracle Mask	48.9 ± 0.9	0.09 ± 0.01	0.52 ± 0.02	1.00	71.3 ± 2.2
Typed API sandbox					
Penalized PUCT	56.7 ± 1.4	0.22 ± 0.03	0.79 ± 0.02	0.95 ± 0.010	51.4 ± 1.8
LLM node filter	55.4 ± 1.8	0.21 ± 0.03	0.68 ± 0.03	0.91 ± 0.014	62.7 ± 2.3
Oracle Mask	63.5 ± 0.8	0.04 ± 0.01	0.48 ± 0.01	1.00	24.1 ± 0.8
Compositional gridworld					
Penalized PUCT	50.3 ± 1.5	0.29 ± 0.03	0.81 ± 0.02	0.97 ± 0.008	22.6 ± 0.8
LLM node filter	49.1 ± 1.7	0.32 ± 0.04	0.73 ± 0.03	0.92 ± 0.013	28.4 ± 1.1
Oracle Mask	56.2 ± 0.8	0.07 ± 0.01	0.51 ± 0.02	1.00	9.8 ± 0.3
WorkArena constrained tasks					
Penalized PUCT	36.1 ± 1.5	0.34 ± 0.03	0.83 ± 0.02	0.951 ± 0.008	165.6 ± 5.2
LLM node filter	36.0 ± 1.7	0.31 ± 0.04	0.72 ± 0.03	0.918 ± 0.012	178.9 ± 6.4

The timing decomposition shows that task-level encoding is amortized over search; CAI's lower search component more than offsets its compiler cost.

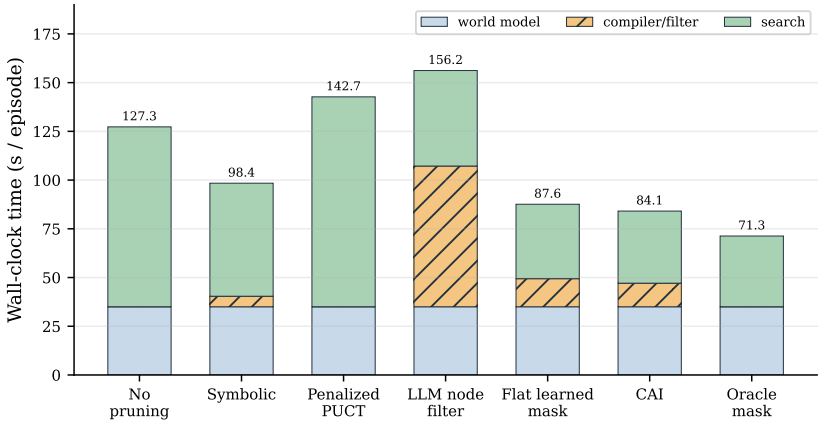


Figure 3: WebArena end-to-end wall-clock decomposition. Task-level encoding and all node readouts are included inside the timing boundary. Oracle Mask is a cached privileged lookup whose offline label-generation cost is excluded.

The ablations distinguish graph structure, uncertainty retention, and false-negative weighting without changing the shared search protocol.

Table 6: Mechanism ablations, unweighted macro-average across the pre-specified WebArena, Typed API, and gridworld isolation suite. WorkArena serves as an additional end-to-end transfer workload in Table 1. All variants retain the data, width, optimizer, search budget, and validation procedure.

Variant	Δ success	WM-FBR	Rel. expansions
Full CAI	0.0	0.98	0.66
Pooled contract, no graph	-4.8	0.91	0.73
No uncertainty retention	-2.3	0.89	0.61
Symmetric loss	-3.1	0.86	0.62

The no-graph variant pools the canonical contract fields and removes typed nodes, edges, and message passing; unlike Flat, it does not restore the full raw task serialization. The retention ablation applies M without Eq. 5, and the symmetric-loss variant sets $\lambda_+ = \lambda_-$. Each variant is trained from initialization and uses the same validation selection rule as full CAI.

C Raw Counts and Paired Uncertainty

All primary-suite episodes completed: WebArena $1000/1000$, Typed API $1500/1500$, gridworld $750/750$, and WorkArena $900/900$. The Typed API count here is the 300-task primary test and excludes the separate 480-task factorial suite. Across these units, Flat/CAI violation-event counts were $240/180$ on WebArena, $240/180$ on Typed API, $188/143$ on gridworld, and $243/180$ on WorkArena.

The raw ledger summarizes the paired success discordances underlying the task-cluster bootstrap.

Table 7: Per-seed raw success and paired discordance counts. Each cell is Flat successes / CAI successes / CAI-only successes / Flat-only successes. Tasks are paired within seed.

Domain	Seed	Flat / CAI / CAI-only / Flat-only
WebArena	0	$85/89/9/5$
	1	$88/92/10/6$
	2	$86/91/9/4$
	3	$91/93/8/6$
	4	$87/91/9/5$
Typed API	0	$174/177/11/8$
	1	$177/180/11/8$
	2	$176/179/10/7$
	3	$179/181/11/9$
	4	$175/180/11/6$
Gridworld	0	$73/76/6/3$
	1	$74/78/7/3$
	2	$76/77/6/5$
	3	$75/78/6/3$
	4	$74/77/6/3$
WorkArena	0	$65/69/8/4$
	1	$67/71/8/4$
	2	$69/72/8/5$
	3	$68/70/7/5$
	4	$68/72/8/4$

The paired task-cluster-bootstrap success intervals for CAI minus Flat are $[+0.2, +3.6]$, $[-0.2, +2.3]$, $[0.0, +3.7]$, and $[+0.1, +3.7]$, respectively. The four-family macro interval is $[+0.8, +2.6]$.

D Factorial Cell Diagnostics

Each disjoint Typed API factorial cell contains 120 held-out tasks and 600 paired task–seed units, for 480 tasks and 2,400 units in this separate suite. Table 8 reports raw success counts and the remaining operating metrics in Symbolic / Flat / CAI order.

The cellwise counts support both the common-Flat factorial contrasts and the strongest-comparator intervals reported in the main text.

Table 8: Detailed results for the separate factorial suite. Δ is CAI minus the strongest non-oracle comparator with paired 95% CI, computed from unrounded paired outcomes. Other triplets are Symbolic / Flat / CAI.

Cell	Success counts	Δ success [CI]	Violations	WM-FBR / time (s)
Mixed/partial	$347/356/371$	$+2.5[+1.2, +3.8]$	$.26/.22/.14$	$.95/.96/.99$ / $34.1/32.7/31.3$
Mixed/full	$360/359/365$	$+0.8[+0.1, +1.5]$	$.18/.17/.13$	$.98/.98/.99$ / $30.8/30.1/29.8$
DSL/partial	$331/336/343$	$+1.2[+0.3, +2.1]$	$.23/.21/.16$	$.96/.97/.98$ / $33.4/31.9/30.7$
DSL/full	$361/358/356$	$-0.8[-1.7, +0.1]$	$.11/.12/.12$	$.993/.984/.991$ / $27.2/28.4/27.9$

E Environment-Audit Protocol

For each method, the audit samples 420 WebArena and 390 WorkArena nodes with common quotas over constraint type and distance-to-boundary strata. It replays six candidates per node, totaling 2520 and 2340 branches per method, respectively. Learned methods use score distance to ε; Symbolic uses the corresponding unresolved-or-equality predicate stratum. Benchmark validators identify environment-feasible continuations, and the evaluator is blinded to method and mask. Samples are

not candidate-paired because each method induces its own boundary pool; the unequal feasible-branch denominators below are the numbers deemed feasible after replay within the identical quota design. For WebArena, false negatives/feasible branches are Symbolic $65/756$, Flat $42/808$, and CAI $19/792$. For WorkArena they are $65/707$, $45/763$, and $22/759$. All labels come from deterministic environment validators, so manual adjudication and inter-annotator agreement are not applicable.

F ADMISSIBILITY-CONTRACT FIDELITY

The fidelity audit contains 450 WebArena and 450 WorkArena specifications, balanced across in-distribution, paraphrase, and compositional holdouts. Two evaluators compare canonical graphs against task-validator constraints before compiler training, reaching $\kappa = 0.92$ $[0.89, 0.95]$ agreement on atomic constraint decisions. Table 9 pools the benchmarks while retaining equal domain weight. Exact-match and accuracy cells include raw correct/total counts; atomic cells include precision/recall/F1 and bootstrap 95% intervals.

Parser accuracy declines smoothly from in-distribution to paraphrase and composition, while conservative abstention prevents uncertain spans from becoming unsupported hard exclusions.

Table 9: Independent admissibility-contract fidelity. Abstention means an uncertain span was excluded from the hard-constraint subgraph; it therefore measures conservative contract construction rather than an incorrect hard constraint.

Split	Metric	Estimate [95% CI] and raw count
In-distribution	Exact graph	90.7% [87.0,93.5] (272/300)
	Atomic P/R/F1	.970/.951/.960 [.947,.972] (890/28/46)
	Numeric bounds	97.0% [94.4,98.5] (291/300)
	Precedence	95.7% [92.7,97.5] (287/300)
	Conservative abstention	4.0% [2.3,6.9] (12/300)
Paraphrase	Exact graph	83.7% [79.1,87.5] (251/300)
	Atomic P/R/F1	.952/.910/.930 [.911,.948] (850/43/84)
	Numeric bounds	92.7% [89.1,95.2] (278/300)
	Precedence	90.0% [86.1,93.0] (270/300)
	Conservative abstention	9.3% [6.6,13.2] (28/300)
Composition	Exact graph	78.0% [73.0,82.3] (234/300)
	Atomic P/R/F1	.929/.873/.900 [.879,.920] (816/62/119)
	Numeric bounds	87.7% [83.5,91.0] (263/300)
	Precedence	85.0% [80.5,88.6] (255/300)
	Conservative abstention	12.0% [8.9,16.3] (36/300)

G ACTION-SPACE SCALING

The scaling study pools WebArena and WorkArena, stratifies tasks by observed candidate-action count, and preserves five paired seeds per task. Figure 4 and Table 10 show that CAI's expansion reduction increases from 24% to 42% while success remains within 1.5 points of its public-benchmark average. The evidence covers the observed range through 100 actions.

As candidate sets grow, CAI converts a larger fraction of proposed expansions into savings while preserving the best success and runtime profile.

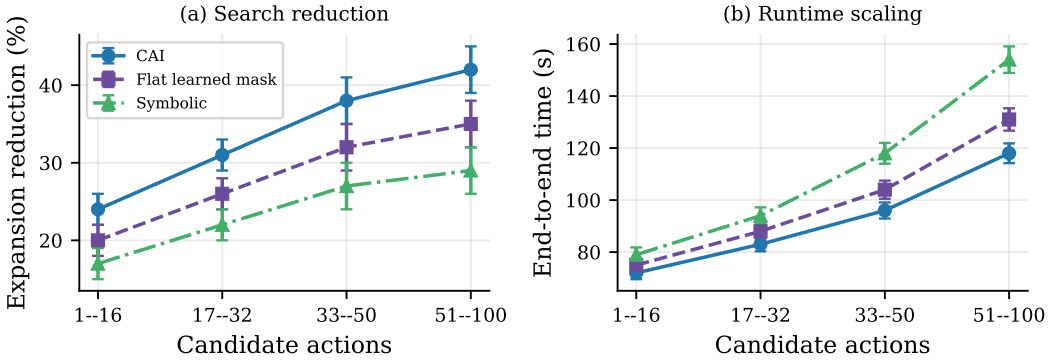


Figure 4: Action-space scaling across the two public web benchmarks. CAI converts larger candidate sets into increasing expansion savings while retaining lower end-to-end time than matched Flat and Symbolic alternatives.

The numerical bins underlying Figure 4 show that CAI's advantage persists throughout the observed action-count range.

Table 10: Scaling values corresponding to Figure 4. Each method cell reports success / expansion reduction / time, with mean ± SE over five seeds.

Actions	Tasks	Method	Success / expansion reduction / time
1–16	112	CAI	$42.8 \pm 1.1/24\% \pm 2\%/72 \pm 2.4$
		Flat learned mask	$41.2 \pm 1.2/20\% \pm 2\%/75 \pm 2.6$
		Symbolic	$39.4 \pm 1.3/17\% \pm 2\%/79 \pm 2.8$
17–32	104	CAI	$42.7 \pm 1.2/31\% \pm 2\%/83 \pm 2.7$
		Flat learned mask	$40.9 \pm 1.3/26\% \pm 2\%/88 \pm 3.0$
		Symbolic	$38.8 \pm 1.4/22\% \pm 2\%/94 \pm 3.2$
33–50	91	CAI	$42.1 \pm 1.3/38\% \pm 3\%/96 \pm 3.1$
		Flat learned mask	$40.3 \pm 1.4/32\% \pm 3\%/104 \pm 3.5$
		Symbolic	$37.9 \pm 1.5/27\% \pm 3\%/118 \pm 4.0$
51–100	73	CAI	$41.2 \pm 1.4/42\% \pm 3\%/118 \pm 3.8$
		Flat learned mask	$39.4 \pm 1.5/35\% \pm 3\%/131 \pm 4.3$
		Symbolic	$36.8 \pm 1.6/29\% \pm 3\%/154 \pm 5.1$

Across all four action-count bins, CAI retains the highest success while its expansion and time advantage over Flat increases with the candidate set. This pattern is consistent with amortizing a fixed task representation while binding only the actions available at each node.

H CONTRACT EVIDENCE MATRIX

The matrix below summarizes which observation supports each part of the mechanism claim. It separates executed outcomes from model-relative diagnostics and from ablations and scaling tests on representation, uncertainty, and action-set size.

Each row connects one component-level claim to a controlled observation and its planning consequence.

Table 11: Evidence matrix linking each contract object to its controlled test and observed planning consequence.

Contract object and stressor	Controlled evidence and implication
Feasible recoveries near a score boundary	Method-blind environment validators estimate CAI, Flat, and Symbolic recovery errors under identical boundary-stratum quotas on WebArena and WorkArena. Lower environment FNR, sampled-set recovery loss, and replay-violation incidence connect representation decisions to executed recovery (Table 3; Appendix E).
Mixed budget, order, and type constraints	Schema types and executed validators support the balanced 2×2 factorial suite. Success, violations, and compute expose both the mixed/partial premium and the DSL/full symbolic boundary (Tables 1–2).
Parser grounding under paraphrase and composition	Canonical annotations measure graph and atomic fidelity on both public benchmarks. Rising conservative abstention localizes uncertainty without introducing unsupported hard exclusions (Table 9).
End-to-end utility as candidate sets grow	A shared world model and timing boundary isolate search control over the observed 1–100-action range. Expansion savings and runtime measure compiler amortization without sacrificing success (Figure 4; Table 10).