TSCraft: An Experience-Guided Agent for Evolving Traffic Signal Control Policies

Abstract

Traffic signal policies must adapt to demand while remaining fast and auditable. Rule-based policies need manual revision, deep RL encodes control in learned parameters, and online language-model controllers generate and validate actions inside the control loop. We present TSCraft, an experience-guided agent that evolves compact code policies offline and freezes one for deployment. It maintains multiple policy lineages and revision experience, evaluates anonymous candidates on matched simulator scenes, and maps lane-group demand to legal phases through a topology-normalized interface. The frozen policy runs locally at each intersection without language-model calls, parameter updates, or inter-intersection communication. In one 20-round campaign, TSCraft evaluated 40 candidates and promoted eight distinct non-seed policies, producing an explicit stepwise trajectory. Across matched flows and horizons, each controller is evaluated through its released deployment interface. Under this end-to-end protocol, TSCraft achieves 15.30 s mean waiting on a 23-scenario test constructed after policy selection, versus 25.08 s for MaxPressure, with lower waiting in 22 scenarios. When the same source is deployed at every intersection, it reduces average waiting time relative to MaxPressure by 25.49%, 20.96%, and 22.50% on the evaluated Hangzhou, Jinan, and New York flows. Across 23 sealed episodes, a replay record reconstructs all 5,295 phase choices and execution-aware durations. TSCraft thus places LLM reasoning between releases while fast, auditable code controls the evaluated intersections.

1 Introduction

Traffic signal timing allocates limited green time to traffic that changes by movement, junction, and hour. The deployed policy must do more than reduce delay. It must also run within the control cycle, expose why a phase was selected, and remain maintainable as demand changes. Retuning every junction by hand is expensive. Replacing that work with an opaque or slow controller creates a different deployment problem.

Representative controller families address different parts of this problem. FixedTime and manually engineered rules are fast and inspectable, but they require manual redesign or retuning when demand or junction configurations change. MaxPressure adapts through an explicit pressure objective, but is constrained by a predefined state, phase, and timing interface (Varaiya, 2013). Deep reinforcement learning (RL) learns simulator-based control mappings (Wei et al., 2019a; Chen et al., 2020; Wei et al., 2019b), but the resulting parameterized policies are harder to inspect and typically require compatible observation–action representations for reuse. LLM-based controllers can reason over structured traffic descriptions and provide natural-language rationales (Lai et al., 2025), but online deployment incurs generation, parsing, and validation at each decision. In our controller-call study, LLMLight requires 2.87 s per decision, compared with 17.7 ms for the frozen TSCraft policy.

TSCraft treats traffic signal control as a maintained executable artifact. An LLM agent revises the policy between releases using simulator evidence and accumulated experience. The process preserves alternative policy lineages and uses matched anonymous evaluation to select each release. The selected source is then frozen and reused across intersections without online model calls.

Our technical contributions are threefold.

- **A multi-lineage policy-evolution process.** TSCraft preserves a bounded population of code policies, records the outcomes of prior revisions, and authors two isolated alternatives per round. Candidates are compared with the current reference on matched simulator scenes under hidden identities. This process produced eight accepted non-seed policies in a preserved 20-round campaign.

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.

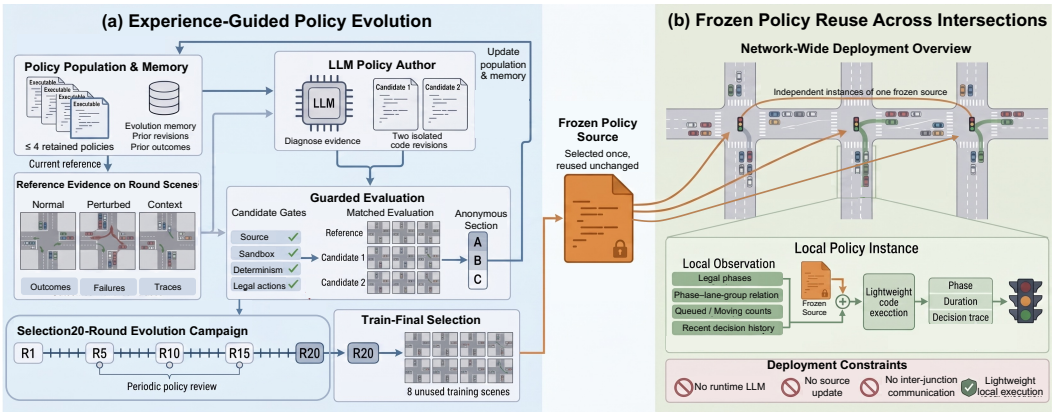


Figure 1: Overview of TSCraft. (a) In each round of the offline campaign, a bounded policy population and its accumulated experience provide reference evidence to an LLM policy author. Two isolated revisions pass source, sandbox, determinism, and legal-action gates before matched evaluation and anonymous selection. Periodic reviews retain the policy population, and a final comparison on eight unused training scenes freezes one source. (b) The same frozen source is instantiated independently at every junction. Each local instance maps a topology-normalized observation to a phase, duration, and decision trace through lightweight code, without runtime LLM calls, source updates, or inter-junction communication.

- **A fixed phase–demand code interface.** The interface maps anonymous lane-group demand to the legal phases that serve it, while fixing observation fields, action bounds, and safety checks across revisions. A deployed decision contains a phase, a duration, and a source-level trace. Replay reconstructed all 5,295 decisions in the sealed test episodes.
- **Topology-normalized multi-intersection evolution and reuse.** Policy code uses phase–lane-group relations rather than junction IDs, road IDs, or fixed phase numbers. Evolution exposes the same interface to several intersections, after which one frozen source is instantiated unchanged at unseen junctions and across full networks. It records lower waiting than MaxPressure in 22 of 23 sealed scenarios and the lowest full-network AWT among the compared methods in all three evaluated cities.

2 TSCraft

2.1 Overview: Offline Policy Evolution and Online Control

TSCraft moves language-model reasoning out of the signal-control loop. The LLM agent revises complete controller programs between releases; it does not select phases for deployed intersections. This choice separates a slow, evidence-seeking maintenance process from the latency-sensitive controller that acts on every decision step.

Figure 1 follows this separation. In the offline plane, the bounded policy population and evolution memory identify a current reference and summarize its outcomes, failures, and traces on the round scenes. The LLM policy author diagnoses this evidence and produces two isolated code revisions. Each revision must pass source, sandbox, determinism, and legal-action gates. The reference and surviving candidates are then evaluated on the same scenes and horizon, after which anonymous selection updates the population and memory. TSCraft repeats this cycle for 20 rounds with periodic policy reviews. A final comparison of at most four survivors on eight unused training scenes selects and freezes one source before validation or test results are opened.

In the online plane, each intersection runs an independent instance of the frozen source. The local instance reads legal phases, phase–lane-group relations, lane-group demand, and recent local history through the topology-normalized interface. Lightweight code execution returns a legal phase, its duration, and a decision trace. Deployment makes no LLM call, changes no parameter or source code, and exchanges no state between intersections.

2.2 Fixed Code-Policy Interface

Policy revisions are comparable only when they solve the same control problem. TSCraft fixes the observation and action interface before evolution. For junction j, let $\mathcal{G}_j$ denote its anonymous lane-group tokens, $\mathcal{P}_j$ its legal phase tokens, and $\Phi_j : \mathcal{P}_j \to 2^{\mathcal{G}_j}$ the relation that maps a phase to the lane groups it serves. At decision step t, the policy observes

$$\mathbf{o}_{j,t} = \left(s_t, p^{\mathrm{cur}}_{j,t}, e^{\mathrm{cur}}_{j,t}, \mathcal{P}_j, \Phi_j, \mathbf{q}_{j,t}, \mathbf{n}_{j,t}, \mathbf{w}_{j,t}, \kappa_j, \mathbf{h}^{(6)}_{j,t}\right). \tag{1}$$

Here, $\mathbf{o}_{j,t}$ is the local observation and s_t is simulator time in seconds. The symbols $p^{\mathrm{cur}}_{j,t}$ and $e^{\mathrm{cur}}_{j,t}$ denote the current phase and its elapsed time. For every lane group in $\mathcal{G}_j$, the maps $\mathbf{q}_{j,t}$, $\mathbf{n}_{j,t}$, and $\mathbf{w}_{j,t}$ provide queue length, vehicle count, and waiting-vehicle count. The map κ_j lists the compatible movements of each lane group, and $\mathbf{h}^{(6)}_{j,t}$ contains the six most recent anonymous decisions.

A code policy c maps this observation to

$$\begin{aligned} c(\mathbf{o}_{j,t}) &= (p_{j,t}, \tau_{j,t}, \mathbf{z}_{j,t}, \mathbf{d}_{j,t}), \\ p_{j,t} &\in \mathcal{P}_j, \qquad \tau_{\min} \le \tau_{j,t} \le \tau_{\max}. \end{aligned} \tag{2}$$

The output $p_{j,t}$ is the selected legal phase, and $\tau_{j,t}$ is its requested duration in seconds. The non-empty sequence $\mathbf{z}_{j,t}$ records the executed decision path, while the optional map $\mathbf{d}_{j,t}$ stores diagnostics. The interface-wide safety bounds are $\tau_{\min} = 5$ s and $\tau_{\max} = 90$ s; the frozen policy evaluated in Section 3 requests 5–45 s. A controller instance may retain private state between calls.

The agent may change the program's phase-ranking logic, duration rule, and private-state update. It may not change the fields in Equation 1, emit an illegal phase, exceed the duration envelope, access files or networks, or invoke another model. These fixed boundaries let the harness validate every candidate under the same contract. They also expose how lane-group demand enters each legal phase without revealing junction IDs, road IDs, or raw phase numbers.

2.3 Experience-Guided Policy Evolution

Editing one incumbent in place can discard a useful mechanism when a revision regresses. TSCraft separates preservation from revision through a bounded policy population. Its state after round r is

$$\Omega_r = \left(\Pi_r, c^{\mathrm{ref}}_r, \mathcal{M}_r\right), \quad c^{\mathrm{ref}}_r \in \Pi_r, \quad |\Pi_r| \le K. \tag{3}$$

The index $r \in \{0, \dots, R\}$ denotes the evolution round. Ω_r is the complete evolution state, Π_r is the set of retained code policies, c^{ref}_r is the active reference policy, and $\mathcal{M}_r$ is the evolution memory. We use a round budget $R = 20$ and population capacity $K = 4$.

Policy population and evolution memory. Each member of Π_r stores its source hash, parent lineage, admission round, mechanism summary, and behavior signature. Mutation and rewrite use one parent; recombination uses two. The memory $\mathcal{M}_r$ contains the latest eight committed rounds and the latest training-checkpoint summary. A round record links a code change to its expected behavior, falsification condition, observed behavior, selection outcome, and next search direction. The memory is limited to the same campaign and excludes validation and test evidence.

Matched evidence and anonymous selection. Each round evaluates the reference on three training scenes:

$$\mathcal{C}_r = \left\{\xi^{\mathrm{nor}}_r, \xi^{\mathrm{per}}_r, \xi^{\mathrm{ctx}}_r\right\}. \tag{4}$$

The elements ξ^{nor}_r, ξ^{per}_r, and ξ^{ctx}_r are the normal, perturbed, and context scenes assigned to round r. From their evidence and $\mathcal{M}_{r-1}$, the agent specifies two distinct revision hypotheses, their parents, and falsification conditions. Two isolated authoring sessions produce candidates $c^{(1)}_r$ and $c^{(2)}_r$; neither session can inspect the other candidate.

Candidates first pass source, sandbox, determinism, and legal-action checks. The valid candidates and c^{ref}_{r-1} then run on the same $\mathcal{C}_r$. Let $\mathcal{Q}_r = \{c^{\mathrm{ref}}_{r-1}, c^{(1)}_r, c^{(2)}_r\}$ be these three policies. Their execution evidence and the selection outcome are

$$\begin{aligned} \mathbf{e}_r(c) &= \left((\mathbf{m}(c,\xi), \mathbf{b}(c,\xi))_{\xi \in \mathcal{C}_r}, \mathbf{g}(c)\right), \\ y_r &= \mathcal{S}\left(\{(\ell, \mathbf{e}_r(\beta_r(\ell))) : \ell \in \mathcal{L}\}, \bar{\Pi}_{r-1}\right). \end{aligned} \tag{5}$$

In Equation 5, $c \in \mathcal{Q}_r$ denotes one policy, $\mathbf{m}(c, \xi)$ its traffic-metric vector on scene ξ, $\mathbf{b}(c, \xi)$ its execution-behavior summary, and $\mathbf{g}(c)$ its gate and diagnostic-probe record. The vector $\mathbf{e}_r(c)$ collects this evidence across the three scenes. The neutral label set is $\mathcal{L} = \{A, B, C\}$, and $\beta_r : \mathcal{L} \to \mathcal{Q}_r$ is a secret one-to-one mapping from labels to policies. The selector $\mathcal{S}$ receives the labeled evidence and $\bar{\Pi}_{r-1}$, the anonymous summaries of the retained population. Its outcome y_r is a selected label, a request for bounded additional evidence, or a no-clear-winner decision. The selector cannot read policy source, authoring intent, evolution memory, or the secret mapping.

Round update and policy freeze. An eligible selected candidate enters the population; a full population also requires an anonymous eviction choice. Token-permutation, phase-cardinality, demand-response, and parent-distance probes inform selection but do not impose a traffic objective. At rounds 5, 10, and 15, a training-owned checkpoint compares the five-round block start, block end, and an optional nominated intermediate. After round R, at most four survivors enter an anonymous tournament on eight unused training scenes. The tournament freezes $c^\star$. Validation and test evidence remain unavailable throughout authoring, round selection, checkpointing, and the final tournament.

2.4 Multi-Intersection Evolution and Frozen Reuse

The code interface removes site-specific identifiers, while the evolution curriculum exposes the agent to structural variation. Each round assigns a primary junction for normal and perturbed evidence and a context junction that uses the same interface. Multi-intersection campaigns balance primary and context exposure over the round budget. Changing the number or geographic scope of curriculum junctions does not change the policy contract, candidate gates, or selection process.

Deployment reuses source code rather than authoring history. Let $\mathcal{J}$ be the set of deployment junctions and $c^\star$ the policy selected by the training-final tournament. Junction $j \in \mathcal{J}$ receives an independent instance $c_j^\star$:

$$\begin{aligned} c_j^\star(\mathbf{o}_{j,t}) &= (p_{j,t}, \tau_{j,t}, \mathbf{z}_{j,t}, \mathbf{d}_{j,t}), \\ \mathrm{source}(c_j^\star) &= \mathrm{source}(c^\star), \qquad \forall j \in \mathcal{J}. \end{aligned} \tag{6}$$

Here, $\mathrm{source}(\cdot)$ returns a controller's source program, and the superscript $\star$ marks the frozen selection. The symbols $p_{j,t}$, $\tau_{j,t}$, $\mathbf{z}_{j,t}$, and $\mathbf{d}_{j,t}$ have the meanings defined in Equation 2. Junctions differ only in their local observations and private instance states. No node-specific code, model retraining, runtime LLM call, or inter-junction communication is added. Sections 3.2 and 3.3 evaluate this unchanged-source reuse on held-out intersections and full road networks.

3 Experiments

3.1 Experimental Setup

We evaluate TSCraft at one controlled junction and across an entire road network. We use CityFlow with the road networks released by LLMLight (Lai et al., 2025; Zhang et al., 2019). The main comparison contains a 19-scenario public single-intersection panel and a 26-flow, 1,800-second full-network panel. A separate 23-scenario bank was generated after policy freeze on the same three network families and is never pooled with the public panel. We use this sealed 23-scenario test bank for the final comparison. The *single-intersection* panel contains five seen Hangzhou, five held-out Hangzhou, five Jinan, and four New York scenarios. The evaluated method controls one target while all other signals use FixedTime. The *full-network* panel deploys one method at every node on common flows: 20 Hangzhou, four Jinan, and two New York flows. "HZ seen" denotes new scenarios at an evolution junction; "HZ unseen" denotes scenarios at a held-out Hangzhou junction.

We compare FixedTime, MaxPressure (MP), released MPLight, PressLight, and CoLight checkpoints, LLMLight with its released LightGPT backbone, and one frozen TSCraft policy. Released RL and LLMLight baselines keep their original weights and four-action adapters. FixedTime and MP choose among eight legal phases every 30 s. TSCraft chooses among eight legal phases and requests a 5–45 s duration. No method is updated on a held-out target.

At one intersection, we report mean target waiting time. At network scale, we report average travel time (ATT), average waiting time (AWT), and average queue length per lane (AQL). Tables 1

Method	Mean target-intersection waiting time (s) ↓ HZ seen	HZ unseen	Jinan	NY	Software time (ms)
FixedTime	34.182	26.453	33.040	88.227	**0.015**
MaxPressure	12.390	10.147	18.048	42.631	0.166
MPLight	21.155	19.522	42.837	**31.651**	18.043
PressLight	32.835	24.488	35.891	58.667	4.542
CoLight	17.675	14.154	25.959	52.619	7.7–11.1[†]
LLMLight	54.538	33.630	226.012	72.960	2870.777
TSCraft	**2.612**	**1.822**	**5.876**	33.588	17.664

Table 1: Single-intersection control on public LLMLight flows. One target uses the listed method; all other intersections use FixedTime. Groups contain 5/5/5/4 scenarios. RL methods use released weights, and no method is updated on a held-out target. Call time is measured in the controller-call study. [†]CoLight reports the range of native batched graph-forward means over 12-, 16-, and 196-node networks; it is excluded from isolated-call timing ranks.

and 2 compare complete systems under their released interfaces on matched flows and horizons. A controller-call study measures software time per controller call. CoLight has no isolated-node path, so its native batched graph-forward time is shown but not ranked. Bold, single-underlined, and double-underlined entries denote the best, second-best, and third-best displayed means. Paired bootstrap intervals are reported for the 20 Hangzhou network flows. Complete adapter, fallback, and timing protocols appear in the supplement. We keep these native-interface end-to-end tables separate from the common-interface selector diagnostic in Appendix S1, Table S5: the former preserves each released controller's action space and decision clock, whereas the latter fixes the phase set and 30-second clock to isolate phase-selection behavior.

3.2 Control Performance at Single Intersections and Full Networks

TSCraft records the lowest displayed mean in three of four single-intersection groups and eight of nine full-network city–metric cells. Tables 1 and 2 report end-to-end system performance on matched flows and horizons while preserving each controller's released action space and decision clock. They report absolute traffic values and software time per controller call. The supplement visualizes full-network metrics after normalization by MP.

Single-Intersection Control and Decision Time

TSCraft achieves the lowest mean waiting time on seen Hangzhou, held-out Hangzhou, and Jinan. Table 1 compares rules, released RL checkpoints, online LLMLight, and the frozen TSCraft policy on the same public-flow panel. On New York, MPLight records 31.65 s and TSCraft 33.59 s. Because the four New York observations are strongly skewed, we treat these values as descriptive means rather than a stable rank. The frozen code policy remains within 1.94 s of the best target-compatible released RL checkpoint while retaining a direct source-level decision path.

TSCraft also removes online generation from the timing path. It averages 17.7 ms per call, including observation construction, isolated execution, and output validation. MPLight averages 18.0 ms and LLMLight 2.87 s. A six-run TSCraft repetition records 17.4 ms. CoLight's 7.7–11.1 ms range contains native batched graph-forward means rather than isolated controller calls. The supplement reports cold-start and tail latency.

Sealed-Test Performance

TSCraft improves all four aggregate traffic outcomes against MP on the sealed 23-scenario test. It records 15.299 s mean waiting versus 25.077 s for MP and wins 22 of 23 paired scenarios. The paired mean difference is -9.778 s with a 95% bootstrap interval of $[-13.209, -5.430]$ from 10,000 scenario-level paired bootstrap resamples. Mean queue falls from 4.666 to 2.653 vehicles, residual traffic falls from 9.217 to 4.087 vehicles, and throughput rises from 277.087 to 280.130. FixedTime and LLMLight record 54.210 and 57.081 s mean waiting. Each scenario has equal weight; the four

Method	Hangzhou (20 flows)			Jinan (4 flows)			New York (2 flows)			Software time (ms)
	ATT	AWT	AQL	ATT	AWT	AQL	ATT	AWT	AQL	
FixedTime	496.13	204.47	1.943	406.51	185.14	3.431	737.19	560.54	0.811	**0.015**
MaxPressure	408.62	84.25	1.254	337.76	108.37	2.717	619.60	300.21	0.473	0.166
MPLight	444.75	156.10	2.019	390.94	182.20	3.264	667.99	462.40	0.578	18.043
PressLight	468.27	194.56	2.326	383.21	171.28	3.149	770.76	624.49	0.688	4.542
CoLight	434.36	144.58	1.980	371.44	166.64	3.232	651.56	375.15	0.487	7.7–11.1†
LLMLight	421.93	116.73	1.670	340.73	110.55	**2.251**	628.15	314.16	0.418	2870.777
TSCraft	**388.39**	**62.77**	**1.108**	**315.95**	**85.65**	2.499	**567.42**	**232.66**	**0.373**	17.664

Table 2: Full-network control on common 1,800-second flows and horizons. Every intersection uses the listed method under its declared interface. ATT and AWT are seconds; AQL is vehicles per lane; lower is better. LLMLight includes documented invalid-output fallback, and native action interfaces differ as stated in Sec. 3.1. Timing comes from the controller-call study; CoLight's graph-level range is contextual and unranked. Hangzhou contains 20 paired flows and supports the reported intervals; Jinan and New York report means over four and two flows. Bold, single-underlined, and double-underlined values are first, second, and third. Software time is measured per controller call in that study and is not full-network step latency. †CoLight reports the range of native batched graph-forward means over 12-, 16-, and 196-node networks.

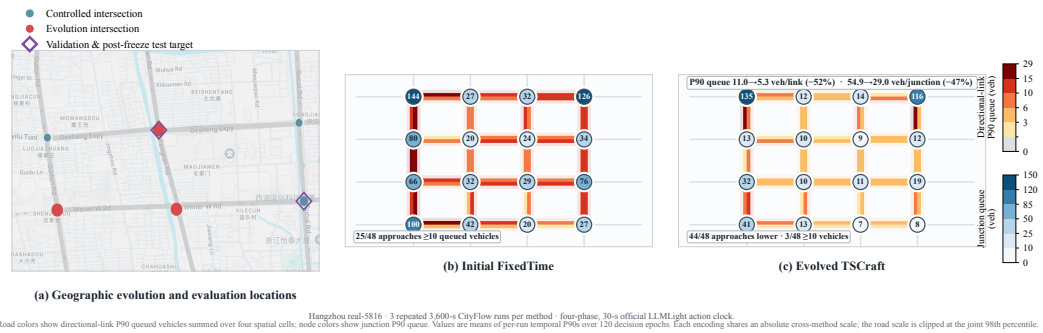


Figure 2: The queue reduction extends across the Hangzhou network. Panel (a) marks the four intersections used during evolution and held-out evaluation intersections. Panels (b)–(c) show the mean of per-run temporal P90 queues on each directed road for Initial FixedTime and the frozen TSCraft phase selector. This diagnostic uses one public Hangzhou flow, four legal phases, a fixed 30-second clock, and three 3,600-second runs per method; TSCraft's duration output is disabled. Node color gives total junction queue; edge color gives road queue.

geographic groups contribute 5/5/5/8 scenarios. Released RL coverage is reported on the complete public panel in Table 1.

Full-Network Control

TSCraft records the lowest AWT among compared methods in all three cities and the lowest value in eight of nine city–metric cells. Table 2 reports complete full-network means under each method's released interface. Relative to MP, TSCraft lowers AWT by 25.49% in Hangzhou, 20.96% in Jinan, and 22.50% in New York. LLMLight is best on the remaining cell, Jinan AQL, with 2.251 versus 2.499 vehicles per lane. Across the 20 paired Hangzhou flows, bootstrap intervals for TSCraft minus MP lie below zero for ATT, AWT, and AQL. Jinan and New York comparisons report means over four and two flows.

In the selector-only spatial diagnostic, queue reductions extend across the network. With four phases and the 30-second clock fixed for both methods, TSCraft lowers the directional-road P90 queue on 44 of 48 directed roads and at all 16 junctions. The mean directional-road P90 queue falls from 11.0 to 5.3 vehicles, and the number of roads averaging at least 10 queued vehicles falls from 25 to 3. This

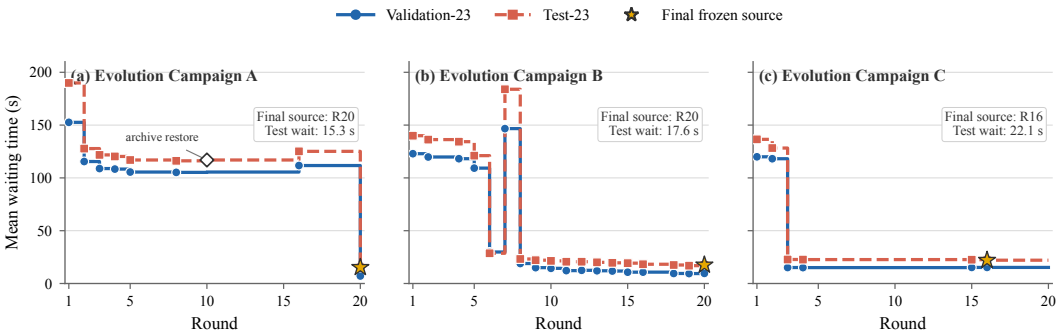


Figure 3: Three complete campaigns expose stepwise and non-monotone policy evolution. Curves are post-campaign replays of the active reference on common Validation-23 and Test-23 banks; these scores were hidden during authoring and selection. Circles and squares show the active reference replayed at every round. Vertical steps indicate admitted source changes, while flat segments carry the incumbent forward. The diamond marks an archived-source restoration, and each star identifies the source round of the frozen policy.

controlled comparison isolates the evolved phase selector against FixedTime. Table 2 evaluates the complete policy against all baselines.

LLMLight ranks more strongly in the network-scale protocol. It ranks outside the leading methods in the single-target panel, where surrounding signals remain FixedTime, but reaches the top three in all nine full-network cells when it controls every node. The two scales describe distinct traffic contexts and control scopes.

3.3 Policy Evolution, Decision Traceability, and Cross-Network Reuse

We analyze the evolved policy from three complementary perspectives. Fixed-bank replay shows how admitted revisions change performance. An instrumented decision reconstructs the discovered control mechanism. Unchanged-source deployment evaluates reuse across held-out targets and networks.

Accepted Revisions Form Stepwise Evolution Trajectories

All three campaigns produce explicit, stepwise evolution trajectories. Figure 3 replays every saved reference policy on common retrospective Validation-23 and Test-23 banks. These banks were hidden during evolution. In the primary campaign, 40 candidates are evaluated and eight distinct non-seed policies are promoted. Final Test-23 waiting is 15.30, 17.57, and 22.06 s. The frozen sources originate from rounds 20, 20, and 16, respectively. The third campaign freezes its round-16 source, showing that the archive can retain an earlier policy when later edits do not improve fixed-bank replay.

The trajectories show both large improvements and deliberate retention. Campaign A moves from 189.91 s after its first admission to 116.17 s at round 8, restores its round-5 source at round 10, and reaches 15.30 s at round 20. Campaign B makes its largest valid step from 183.93 s at round 7 to 23.21 s at round 8, then finishes at 17.57 s. Campaign C moves from 128.27 s at round 2 to 22.73 s at round 3 and freezes the source admitted at round 16. The curves are not forced to improve monotonically because the agent selects from current training evidence rather than hidden test scores. Supplement Sec. S5.3 reports the complete campaign records and authoring configurations.

Frozen-Policy Traces Reconstruct Phase and Duration Decisions

We analyze the final frozen policy as a case study through its replayable decision traces. For junction j and legal phase p, its compatible-demand score $D_{j,p}$ and stopped-demand score $U_{j,p}$ are

$$D_{j,p} = \sum_{g \in \Phi_j(p)} (q_{j,g} + w_{j,g} + 0.25 n_{j,g}),$$
$$U_{j,p} = \sum_{g \in \Phi_j(p)} (q_{j,g} + w_{j,g}). \tag{7}$$

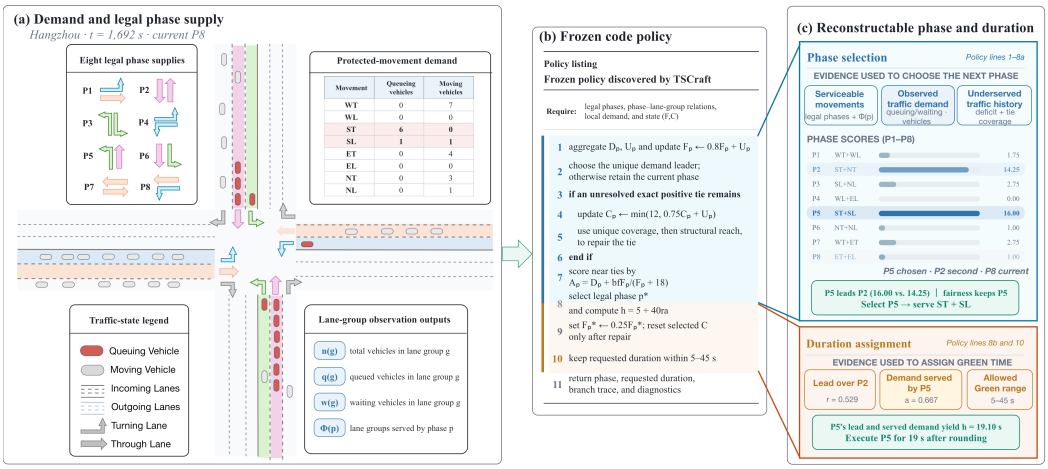


Figure 4: One recorded decision is reconstructed from traffic state and source-code branches. Panel (a) maps lane-group demand to eight legal phase supplies. Panel (b) summarizes the policy mechanism discovered by TSCraft. Panel (c) follows one execution: P5 has demand 16.00 and P2 has 14.25, so the policy selects P5, requests 19.10 s, and executes 19 s after rounding.

Here, g ranges over the lane groups served by p through $\Phi_j(p)$. The values $q_{j,g}$, $w_{j,g}$, and $n_{j,g}$ are the current queue, waiting-vehicle, and total-vehicle counts for lane group g. The controller first ranks phases by compatible demand. It retains the current phase when it belongs to an exact positive tie. If the current phase is excluded, a bounded coverage state and then a deterministic structural rule break the tie. Among phases within a $0.12 \max_p D_{j,p}$ band, a stopped-demand deficit may change the winner. Let $D_1 = D_{j,p^\star}$ and let D_2 be the largest competing demand. The duration branch uses relative dominance $r = D_1/(D_1 + D_2)$ and absolute load $a = D_1/(D_1 + 8)$, then requests $\tau = 5 + 40ra$ within the frozen policy's 5–45 s range.

The case reconstructs both outputs from observed quantities. At $t = 1{,}692$ s, the current phase is P8. P5 has demand 16.00 and P2 has 14.25, so the recorded branch selects P5 to serve the southbound through and left movements. No tie repair is active. With $r = 0.529$ and $a = 0.667$, the duration branch requests 19.10 s and the runner executes 19 s. The explanation is the program's execution trace rather than a post-hoc language-model rationale.

Replaying the frozen program on the recorded observations reconstructs all 5,295 phase choices and execution-aware durations, including 21 episode-horizon clips. The traces contain 3,032 current-phase tie retentions, 558 exact-tie repairs, 1,696 unique-leader decisions, and nine deficit-induced changes. Controlled interventions also recover the expected response in 210 of 210 phase tests and 210 of 210 duration tests. These results verify trace fidelity, while Figure 4 shows how one decision can be inspected at the level of observed quantities and code branches.

One Topology-Normalized Policy Transfers across Networks without Retraining

Tables 1 and 2 compare the same frozen TSCraft source with FixedTime, MP, released RL checkpoints, and LLMLight at single-intersection and full-network scales. TSCraft evolves on four Hangzhou intersections and receives no source edit, parameter update, or target feedback on held-out Hangzhou, Jinan, or New York. It ranks first on held-out Hangzhou and Jinan and second on New York. Its New York mean is within 1.94 s of the best released RL checkpoint.

Full-network deployment extends unchanged-source reuse to 16 Hangzhou, 12 Jinan, and 196 New York nodes. Hangzhou tests network-scale reuse in the source city. Jinan and New York test cross-network zero-shot reuse. TSCraft obtains the lowest AWT among compared methods in all three

cities and the lowest value in eight of nine city–metric cells. The same source controls every evaluated junction without node-specific code.

3.4 Framework Components and Evolution Scope

Mechanism Ablations

At the same 20-round budget, the system records 15.299 s mean waiting on the test set; removing phase–demand alignment, rolling experience, or lineage breadth raises this to 24.261, 119.225, or 28.172 s, respectively. These controlled removals show that phase–demand alignment, rolling experience, and lineage breadth each contribute to the evolved controller. The LLM is used only for offline authoring; the frozen runtime phase selector is isolated by the common-interface diagnostic in Appendix S1, Table S5. Appendix S5.1 gives the complete geographic and queue breakdown.

Evolution Curricula

Under the same 20-round budget and 19-scenario target panel, the four-Hangzhou-intersection curriculum records 1.822 s on held-out Hangzhou and 33.588 s on New York. A single-Hangzhou-intersection curriculum records 165.362 and 515.018 s on the same groups, while the Hangzhou–Jinan curriculum records 5.977 s on Jinan and 328.804 s on New York. Appendix S5.2 reports the complete curriculum records; Appendix S5.3 and S5.4 provide the three-campaign and authoring-model analyses.

4 Related Work

Traffic signal control. FixedTime follows a predetermined cycle, whereas MP selects phases from explicit pressure (Varaiya, 2013). PressLight, MPLight, and CoLight learn pressure-based or graph-aware policies (Wei et al., 2019a; Chen et al., 2020; Wei et al., 2019b). Programmatic alternatives include π-Light, GPTraffic, GPLight+, and SymLight (Gu et al., 2024; Liao et al., 2024; 2025a;b). π-Light also studies cross-region policy reuse. TSCraft instead evolves stateful phase–duration code over anonymous phase–lane-group relations and deploys one source across an entire network.

Language models for traffic control. LLMLight generates signal actions online, CoLLMLight adds cooperative reasoning, and VLMLight uses an LLM meta-controller (Lai et al., 2025; Yuan et al., 2026; Wang et al., 2025b). TSCraft limits LLM use to offline policy evolution and deploys a non-LLM program.

Language-model program and agent search. FunSearch, Evolution of Heuristics, and EUREKA combine language-model proposals with program execution (Romera-Paredes et al., 2024; Liu et al., 2024; Ma et al., 2024). ADAS searches complete agent programs, while AFlow optimizes code-represented workflows (Hu et al., 2025; Zhang et al., 2025). Experiential agents condition future runs on reusable insights or workflows (Zhao et al., 2024; Wang et al., 2025d). TSCraft applies this search pattern to policy maintenance, but the selected traffic controller invokes no LLM at runtime.

LLM-guided traffic-policy evolution. EvolveSignal improves fixed-time timing plans, TPET evolves lightweight heuristics, and SignalClaw evolves skills for runtime composition (Wang et al., 2025a;c; Lei et al., 2026). TSCraft searches stateful phase–duration policies with cross-round experience, matched three-policy selection, and frozen local or full-network deployment. Supplement Sec. S7 compares these design boundaries.

5 Conclusion

TSCraft uses an experience-guided agent to evolve traffic-signal programs offline and deploys one frozen, topology-normalized source without runtime model calls. The selected policy exposes phase and duration traces, transfers unchanged across the evaluated networks, and records 15.30 s mean waiting on the sealed 23-scenario test bank versus 25.08 s for MaxPressure. This separates policy maintenance from real-time control: agent reasoning operates between releases, while compact code runs at each intersection.

Ethics Statement

We evaluate simulated traffic-control policies only; no personal data are collected and no live infrastructure is actuated. Any field deployment would require independent safety, fairness, and operational review.

Reproducibility Statement

The main text fixes the policy contract, evolution protocol, data splits, baselines, and audit metrics; the appendix supplies complete tables, configurations, traces, timing protocols, and frozen-policy evidence.

AI Use Statement

Language models were used for offline policy authoring and manuscript editing. All values, citations, source files, and claims were checked against archived artifacts; the authors remain responsible for the submission.

References

Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3414–3421, 2020. doi: 10.1609/aaai.v34i04.5744. URL https://ojs.aaai.org/index.php/AAAI/article/view/5744.

Yin Gu, Kai Zhang, Qi Liu, Weibo Gao, Longfei Li, and Jun Zhou. π-Light: Programmatic interpretable reinforcement learning for resource-limited traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21107–21115, 2024. doi: 10.1609/aaai.v38i19.30103. URL https://ojs.aaai.org/index.php/AAAI/article/view/30103.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/36b7acf6f6010652b3f2a433774a66fe-Abstract-Conference.html.

Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. LLMLight: Large language models as traffic signal control agents. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pages 2335–2346. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709379.

Da Lei, Feng Xiao, Lu Li, and Yuzhan Liu. SignalClaw: Llm-guided evolutionary synthesis of interpretable traffic signal control skills, 2026. URL https://arxiv.org/abs/2604.05535.

Xiao-Cheng Liao, Yi Mei, and Mengjie Zhang. Learning traffic signal control via genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 924–932. Association for Computing Machinery, 2024. doi: 10.1145/3638529.3654037.

Xiao-Cheng Liao, Yi Mei, and Mengjie Zhang. GPLight+: A genetic programming method for learning symmetric traffic signal control policy. *IEEE Transactions on Evolutionary Computation*, 2025a. doi: 10.1109/TEVC.2025.3578575.

Xiao-Cheng Liao, Yi Mei, and Mengjie Zhang. SymLight: Exploring interpretable and deployable symbolic policies for traffic signal control, 2025b. URL https://arxiv.org/abs/2511.05790.

Fei Liu, Tong Xialiang, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. Evolution of heuristics: Towards efficient automatic algorithm design using large language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32201–32223. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bs.html.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, and Anima Anandkumar. EUREKA: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70c26937fbf3d4600b69a129031b66ec-Abstract-Conference.html.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6. URL https://www.nature.com/articles/s41586-023-06924-6.

Pravin Varaiya. The max-pressure controller for arbitrary networks of signalized intersections. In Satish V. Ukkusuri and Kaan Ozbay, editors, *Advances in Dynamic Network Modeling in Complex Transportation Systems*, Complex Networks and Dynamic Systems, pages 27–66. Springer, 2013. doi: 10.1007/978-1-4614-6243-9_2.

Leizhen Wang, Peibo Duan, Hao Wang, Yue Wang, Jian Xu, Nan Zheng, and Zhenliang Ma. EvolveSignal: A large language model powered coding agent for discovering traffic signal control strategies, 2025a. URL https://arxiv.org/abs/2509.03335. Version 3.

Maonan Wang, Yirong Chen, Aoyu Pang, Yuxin Cai, Chung Shue Chen, Yuheng Kan, and Man On Pun. VLMLight: Safety-critical traffic signal control via vision-language meta-control and dual-branch reasoning architecture. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3849b5861dcaeaf4758eef0979a98cc6-Abstract-Conference.html.

Ruibing Wang, Shuhan Guo, Zeen Li, Zhen Wang, and Quanming Yao. Evolutionary discovery of heuristic policies for traffic signal control, 2025c. URL https://arxiv.org/abs/2511.23122.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 63897–63911. PMLR, 2025d. URL https://proceedings.mlr.press/v267/wang25bx.html.

Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. PressLight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1290–1298. Association for Computing Machinery, 2019a. doi: 10.1145/3292500.3330949.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. CoLight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922. Association for Computing Machinery, 2019b. doi: 10.1145/3357384.3357902.

Zirui Yuan, Siqi Lai, and Hao Liu. CoLLMLight: Cooperative large language model agents for network-wide traffic signal control. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KeJqoEVOeY.

Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference*, pages 3620–3624, 2019. doi: 10.1145/3308558.3314139. URL https://doi.org/10.1145/3308558.3314139.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5492ecbce4439401798dcd2c90be94cd-Abstract-Conference.html.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (17):19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

APPENDIX: SUPPLEMENTARY MATERIAL

This supplement reports complete results behind the main-paper claims. We keep single-intersection, common-flow full-network, original 3,600-second, policy-evolution, and latency protocols separate because they differ in control scope, horizon, action interface, decision clock, and independent unit.

S0. EVALUATION INTERFACES AND EVOLUTION CONFIGURATION

S0.1 NATIVE AND COMMON-INTERFACE REPORTING

The native-interface tables report complete systems as deployed: released baselines retain their published action heads and decision clocks, while the frozen TSCraft source retains its eight legal semantic phases and adaptive 5–45 s duration. These tables answer the end-to-end deployment question. The common-interface table in Sec. S1 fixes the admitted phase set and a 30-second clock for both FixedTime and the TSCraft selector, disables TSCraft's duration branch, and therefore answers the narrower selector-control question. The two views are not pooled into one rank.

The CityFlow harness selects directly among the legal green phases and advances the simulator for the requested number of steps; it does not insert a separate yellow/all-red interval. Roadnet phase entries with nominal duration below 10 s are excluded from the legal set, and the executed duration is clipped at the simulation horizon.

For target waiting metrics, vehicles observed after warm-up contribute zero when they incur no stopped waiting; P95 uses the nearest-rank definition and includes zero-wait vehicles. Vehicles already present at warm-up are excluded from arrival counts. Network AWT is the vehicle-level mean waiting time from the speed threshold, while AQL is the mean decision-interval waiting count per incoming lane. Residual queue is measured at the effective simulation end, after horizon clipping. These definitions keep the native and matched panels auditable without changing the controller.

Scale	Method family	Training–evaluation relation	Controller interface
Single intersection	TSCraft / FT / MP	Frozen program or rule; no target update	Eight legal semantic phases; TSCraft returns a 5–45 s duration
	RL	Released checkpoint; no target update	Four-action released head; fixed 30 s polling in the released adapter
	LLMLight	Pretrained online model; no target fine-tuning	Four-action parsed output; fixed 30 s polling
Full network	TSCraft	Held-out Hangzhou flows; zero-shot Jinan and New York	Unchanged source over legal local phases; policy-returned duration
	MPLight / PressLight	Held-out Hangzhou flows; zero-shot Jinan and New York	Released checkpoint/output head; fixed 30 s polling
	CoLight	Same-network held-out Hangzhou; cross-network zero-shot in Jinan and New York	Target graph rebuilt; released weights shape-checked and loaded without optimizer updates; fixed 30 s polling
	LLMLight	Pretrained; no target-scenario fine-tuning	Online output plus fallback; fixed 30 s polling

Table S1: Declared interfaces for the main comparisons. The tables measure end-to-end performance on common flows and horizons under these interfaces; they are neither action-capacity- nor decision-clock-matched policy-family experiments.

Component	Configuration
Authoring curriculum	20 rounds; two candidates per round; four Hangzhou intersections with balanced primary-junction exposure and a round-level context schedule
Authoring and selection model	GPT-5.6-Sol in isolated, schema-constrained diagnosis, candidate, selection, and checkpoint stages; no runtime LLM calls
Population and experience	At most four content-distinct programs; rolling summaries of the eight most recent committed rounds
Round execution	Parent and each of two candidates execute one matched normal, perturbed, and context scene, giving nine matched executions per round
Hard candidate gates	JSON/schema, source, compilation, policy-contract, determinism, duplicate, legal-phase, and bounded-duration checks
Diagnostic probes	Token permutation, phase cardinality, demand response, and parent distance; these descriptors inform selection but do not reject a candidate before simulation
Selection and population update	The reference and two candidates are randomly mapped to neutral labels A, B, and C. The selector observes matched execution profiles but not the role-to-label map. An eligible selected candidate enters the population; the selector chooses an eviction when capacity four is reached
Final selection	Surviving programs are compared on disjoint Train-Final scenes after round 20; retrospective validation/test panels remain hidden from authoring
Frozen policy release	One Python controller and its evolution configuration, prompts, schemas, and reported result records
Reproduction environment	Python 3.9.25 and CityFlow 0.1

Table S2: Evolution configuration for the policy reported in the main paper.

The method uses schema-constrained language-model judgment. The selector receives matched traffic summaries, hard-constraint findings, behavior signatures, and population context under neutral labels. Its output must match one of the declared decisions: `select_A`, `select_B`, `select_C`, `no_clear_winner`, or `need_more_evidence`. An ineligible candidate cannot be selected; the selector cites supplied evidence and states scope and uncertainty. There is no hidden weighted score. When the population is full, the selector may also return one anonymous archive-eviction preference; this is resolved by the sealed role map after selection. Thus the selector protocol is explicit while the deployed controller remains unchanged.

The anonymous release contains the frozen policy, evolution configuration, prompts and schemas, and the result records used in this paper. Together they specify the protocol for independent replication; policy authoring remains stochastic.

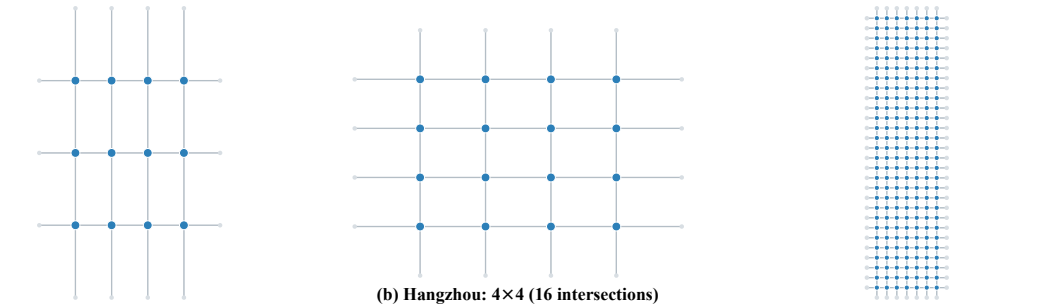
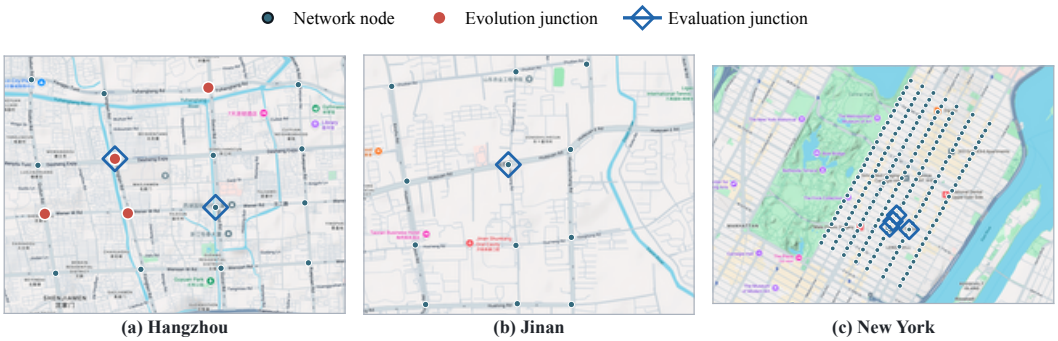


Figure S1: CityFlow road-network topologies used by the LLMLight benchmark. The figure is rendered directly from the public LLMLight roadnet files rather than from map screenshots. Blue nodes are controlled intersections; pale endpoints are virtual boundary intersections. Jinan, Hangzhou, and New York contain 12, 16, and 196 controlled intersections, respectively.



(a) Hangzhou (b) Jinan (c) New York

Figure S2: Geographic context for policy evolution and evaluation. Red markers identify the four Hangzhou junctions used during evolution; blue diamonds identify evaluation junctions. Validation and frozen-policy test flows reuse these junction locations but contain disjoint traffic instances, so the map shows one evaluation marker rather than two overlapping symbols. Markers count unique junctions, not traffic scenarios: the single Jinan junction supports five single-intersection flows, whereas the public-flow New York panel covers four target junctions. The maps provide geographic context only and are not used to compute the results.

S1. Complete Single-Intersection Results

	Mean target-intersection waiting time (s) ↓			
Method	HZ seen	HZ unseen	Jinan unseen	NY unseen
FixedTime	34.182	26.453	33.040	88.227
MaxPressure	12.390	10.147	18.048	42.631
MPLight	21.155	19.522	42.837	**31.651**
PressLight	32.835	24.488	35.891	58.667
CoLight	17.675	14.154	25.959	52.619
LLMLight	54.538	33.630	226.012	72.960
TSCraft	**2.612**	**1.822**	**5.876**	33.588

Table S3: Complete public-flow panel. One target intersection uses the listed method and all other signals use FixedTime. The four groups contain 5/5/5/4 scenarios. RL rows use released checkpoints without target-time updates.

The four New York observations are strongly skewed. TSCraft's mean/median/sample-SD are 33.588/0.244/66.816 s; MPLight's are 31.651/15.202/37.407 s. We treat the displayed New York means as descriptive outcomes rather than stable population ranks.

Method	Wait (s) ↓	Queue ↓	Throughput ↑	Residual ↓
TSCraft	**15.299**	**2.653**	**280.130**	**4.087**
MaxPressure	25.077	4.666	277.087	9.217
FixedTime	54.210	9.371	275.304	14.000
LLMLight	57.081	14.372	263.522	25.000

Table S4: Complete aggregate metrics on the 23-scenario test constructed after policy selection. RL methods appear in the public-flow panel, which provides released-checkpoint coverage for every evaluated topology. Each scenario has equal weight; group sizes are 5/5/5/8.

For TSCraft minus MaxPressure, the paired mean waiting-time difference is -9.778 s with a 95% bootstrap interval of $[-13.209, -5.430]$. TSCraft has lower waiting in 22 of 23 paired scenarios. The test assigns 5/5/5/8 scenarios to reference Hangzhou, unseen Hangzhou, unseen Jinan, and unseen

New York. It was generated after policy selection and remained outside candidate authoring and selection.

Matched-interface diagnostic	FixedTime	TSCraft selector
Mean directional-road P90 queue (vehicles)	11.0	5.3
Roads with lower queue	–	44/48
Roads tied	–	1/48
Roads with higher queue	–	3/48
Junction totals with lower queue	–	16/16
Roads averaging at least 10 queued vehicles	25	3

Table S5: Common-interface selector diagnostic on the official Hangzhou-5816 flow. Both methods use four legal phases, a fixed 30-second decision clock, and three independent 3,600-second CityFlow runs; TSCraft's evolved duration branch is disabled. This table is intentionally separate from the native end-to-end tables: it isolates phase-selection behavior against FixedTime and does not claim that all released baselines share this interface.

S2. Full-Network Results and Uncertainty

Table S6: Complete common-flow full-network means under declared native interfaces. ATT and AWT are seconds; AQL is vehicles per lane. No aggregate is formed across cities. Bold, single-underlined, and double-underlined entries are first, second, and third within each city–metric column.

City	Method	Flows	ATT ↓	AWT ↓	AQL ↓
Hangzhou	FixedTime	20	496.13	204.47	1.943
	MaxPressure	20	408.62	84.25	1.254
	MPLight	20	444.75	156.10	2.019
	PressLight	20	468.27	194.56	2.326
	CoLight	20	434.36	144.58	1.980
	LLMLight	20	421.93	116.73	1.670
	TSCraft	20	**388.39**	**62.77**	**1.108**
Jinan	FixedTime	4	406.51	185.14	3.431
	MaxPressure	4	337.76	108.37	2.717
	MPLight	4	390.94	182.20	3.264
	PressLight	4	383.21	171.28	3.149
	CoLight	4	371.44	166.64	3.232
	LLMLight	4	340.73	110.55	**2.251**
	TSCraft	4	**315.95**	**85.65**	2.499
New York	FixedTime	2	737.19	560.54	0.811
	MaxPressure	2	619.60	300.21	0.473
	MPLight	2	667.99	462.40	0.578
	PressLight	2	770.76	624.49	0.688
	CoLight	2	651.56	375.15	0.487
	LLMLight	2	628.15	314.16	0.418
	TSCraft	2	**567.42**	**232.66**	**0.373**



Figure S3: Full-network performance normalized by MaxPressure. Each method controls every node on the common 1,800-second flows. Cells show ATT, AWT, or AQL divided by the MaxPressure mean for the same city; lower ratios are better.

Comparator	Metric	TSCraft difference	Paired 95% interval
FixedTime	ATT (s)	-107.742	[-122.919, -93.830]
	AWT (s)	-141.692	[-152.315, -131.648]
	AQL (veh./lane)	-0.835	[-0.973, -0.707]
MaxPressure	ATT (s)	-20.228	[-24.469, -15.890]
	AWT (s)	-21.478	[-25.199, -17.315]
	AQL (veh./lane)	-0.146	[-0.175, -0.120]
MPLight	ATT (s)	-56.359	[-61.035, -51.857]
	AWT (s)	-93.325	[-112.177, -75.923]
	AQL (veh./lane)	-0.911	[-1.313, -0.537]
PressLight	ATT (s)	-79.876	[-90.755, -70.926]
	AWT (s)	-131.782	[-149.287, -115.045]
	AQL (veh./lane)	-1.218	[-1.680, -0.795]
CoLight	ATT (s)	-45.964	[-51.807, -40.479]
	AWT (s)	-81.809	[-106.566, -59.505]
	AQL (veh./lane)	-0.872	[-1.322, -0.458]
LLMLight	ATT (s)	-33.540	[-35.716, -31.259]
	AWT (s)	-53.956	[-66.064, -43.125]
	AQL (veh./lane)	-0.562	[-0.829, -0.315]

Table S7: Paired Hangzhou effects over the same 20 flows. Difference is TSCraft minus comparator, so negative favors TSCraft. Intervals use 10,000 scenario-level paired bootstrap draws.

The analysis contains 182 method–flow observations over 26 flows and the 20 paired Hangzhou differences for each comparator. City means in the main paper are rounded only for display.

Jinan and New York contain only four and two flows. Their sample standard deviations for TSCraft are 131.04/122.10/4.545 and 65.09/103.95/0.246 for ATT/AWT/AQL, respectively. We report them as descriptive transfer panels rather than significance tests. The New York LLMLight system falls back to its default action on 23.223% and 20.986% of decisions in the two flows. Its entries describe the executed LLMLight-plus-fallback system.

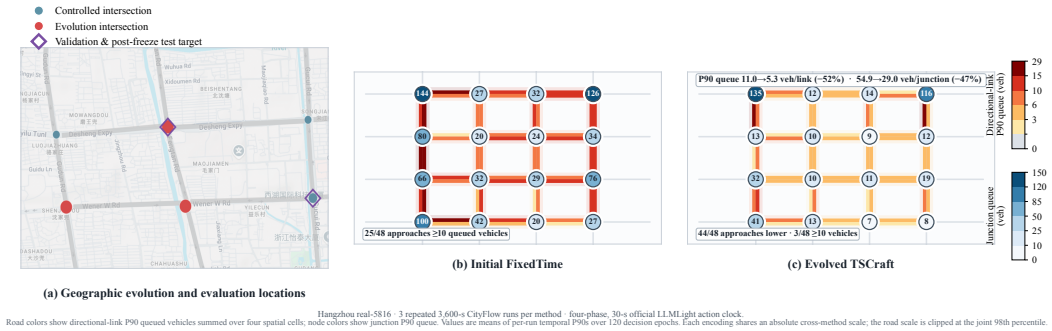


Figure S4: Action-matched spatial queue case in Hangzhou. Initial FixedTime and the frozen TSCraft phase selector are evaluated on the official Hangzhou-5816 flow with four phases, a fixed 30-second clock, and three 3,600-second CityFlow runs per method. TSCraft's evolved duration rule is disabled. Panel (a) marks evolution and held-out intersections. Panels (b)–(c) show the mean of per-run temporal P90 queues by directed road and total junction P90 queue. Scales are shared across methods. TSCraft is lower on 44 of 48 roads, ties on one, and is higher on three; all 16 junction totals are lower.

S3. Original 3,600-Second Protocol

Table S8: Results under the original 3,600-second simulation-horizon protocol. Each method–flow entry is one aggregate run. TSCraft's phase selector is unchanged, while the fixed 30-second clock disables its evolved duration rule. The table reports this robustness panel without rank highlighting.

Scenario	Method	ATT ↓	AWT ↓	Queue (veh.) ↓
Jinan-2000	FixedTime	441.190	66.720	294.142
	MaxPressure	273.199	38.249	106.575
	AdvancedMPLight	270.429	45.190	103.525
	LLMLight	268.731	44.545	102.600
	TSCraft	268.439	42.637	102.600
Jinan-2500	FixedTime	450.113	69.191	394.342
	MaxPressure	265.753	40.198	133.900
	AdvancedMPLight	264.281	47.701	133.233
	LLMLight	263.770	47.143	132.517
	TSCraft	261.702	42.427	129.225
Hangzhou-5816	FixedTime	486.693	72.797	425.117
	MaxPressure	347.742	70.577	215.533
	MPLight	435.790	65.294	348.792
	PressLight	466.539	293.058	479.692
	CoLight	430.255	68.146	336.800
	LLMLight	335.853	65.562	195.183
	TSCraft	334.020	60.882	189.992
New York-double	FixedTime	1535.651	290.125	3173.625
	MaxPressure	1223.270	153.279	2410.158
	LLMLight	1117.285	120.676	2153.733
	TSCraft	1165.472	148.131	2279.783
	MPLight (trained on double)	1496.860	275.891	3038.167
New York-triple	FixedTime	1771.744	428.711	4523.067
	MaxPressure	1566.771	255.468	3936.533
	LLMLight	1497.138	309.197	3813.392
	TSCraft	1525.811	257.122	3854.358
	MPLight (trained on double)	1746.116	407.915	4502.617

Here Queue is the native LLMLight protocol's mean network-total queued-vehicle count, not the vehicles-per-lane AQL used in the common 1,800-second protocol.

The two New York names describe demand: "double" uses twice the base traffic volume and "triple" uses three times the base volume. They are not two model variants. The MPLight row uses a checkpoint trained on New York-double and is labeled accordingly.

S4. Complete Evolution and Replay Evidence

Admitted round	Candidate evaluations so far	Retrospective validation WT	Retrospective test WT
1	2	152.610	189.908
2	4	115.508	127.768
3	6	108.848	121.771
4	8	108.361	120.310
5	10	105.543	116.953
8	16	105.221	116.166
16	32	111.704	125.137
20	40	7.110	15.299

Table S9: All eight admitted policies in the 20-round campaign. The saved policies are evaluated retrospectively on fixed validation and test panels. These values were not revealed to the selector.

Round 10 restores the source associated with the round-5 checkpoint. Round 16 is retained although its retrospective Test-23 value is higher. Both events show that current training evidence, rather than the retrospective banks, governs selection. The release includes the admitted-source sequence and the complete frozen Python controller. The source specifies floating-point comparisons, the near-tie score, structural-reach tuple, state-update order, and requested duration. The simulator runner specifies rounding and episode-horizon clipping.

Replay or intervention check	Result
Recorded phase decisions reproduced	5,295 / 5,295
Execution-aware durations reproduced	5,295 / 5,295
Episode-horizon clips reconstructed	21
Current-phase tie retention branch	3,032
Exact-tie coverage repair branch	558
Unique/base demand branch	1,696
Decisions changed by deficit bonus	9
Phase counterfactual probes	210 / 210 expected
Duration counterfactual probes	210 / 210 expected

Table S10: Frozen-source replay and directional intervention results. Counts sum to all recorded decisions.

S5. Framework Stresses, Curricula, and Authoring Robustness

S5.1 Mechanism Ablations

Variant	HZ seen	HZ unseen	Jinan	New York	All-23 WT	Queue
Full TSCraft	**1.037**	**1.477**	5.057	**39.253**	**15.299**	**2.653**
Without phase–demand alignment	7.849	11.873	10.961	50.574	24.261	4.113
Without rolling experience memory	6.022	2.877	**4.241**	334.560	119.225	9.608
Single policy lineage	16.785	15.337	25.815	44.784	28.172	5.672

Table S11: Complete 20-round mechanism ablations. Each row is a separately evolved 20-round configuration evaluated on the same post-selection Test-23 bank; the remaining evolution and evaluation settings are held fixed. Values are waiting time in seconds except the final queue column.

S5.2 Three Evolution Curricula

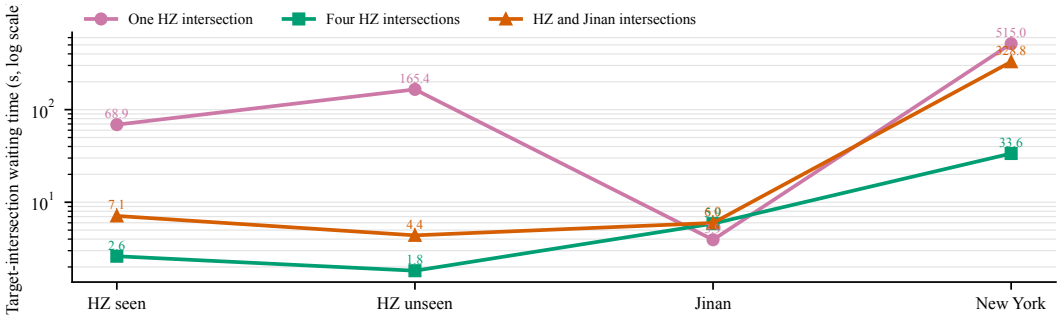


Figure S5: Transfer outcomes for three evolution curricula under the same 20-round budget. Four-intersection Hangzhou evolution gives the lowest cross-panel means. The one-intersection and Hangzhou–Jinan runs retain different transfer profiles at the same final round.

Authoring curriculum	HZ seen	HZ unseen	Jinan	New York
One Hangzhou intersection	68.851	165.362	3.933	515.018
Four Hangzhou intersections	2.612	1.822	5.876	33.588
Hangzhou and Jinan intersections	7.115	4.390	5.977	328.804

Table S12: Curriculum results on the 19-scenario target-intersection panel. The rows respectively implement single-city single-intersection, single-city multi-intersection, and multi-city multi-intersection evolution. The four-intersection Hangzhou curriculum is the policy reported as TSCraft in the main paper. The multi-city schedule has the largest context space but the same 20-round budget; all rows use the same evaluation panel.

S5.3 Three Evolution Campaigns

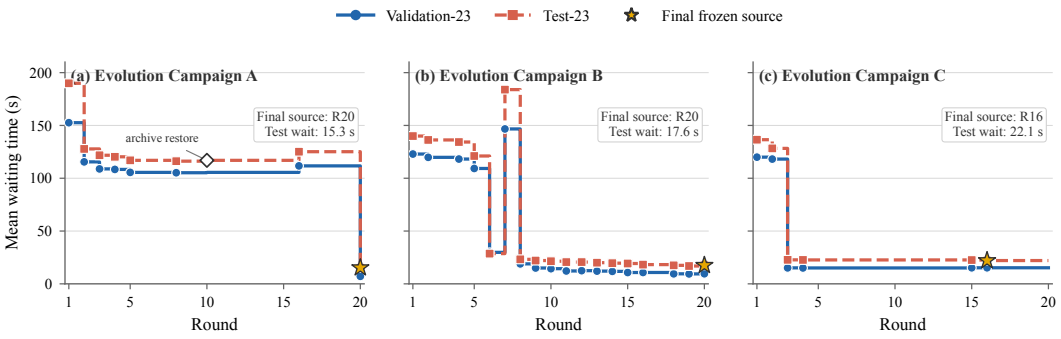


Figure S6: Retrospective trajectories for three independent policy-evolution campaigns. Every admitted frozen policy is replayed after its campaign on the same Validation-23 and sealed Test-23 banks. Unmarked rounds carry the active incumbent forward; markers denote admitted source versions, and the hollow diamond denotes Campaign A's round-10 restoration of its round-5 source. Final policies originate at rounds 20, 20, and 16. Neither fixed bank was exposed to candidate authoring or anonymous selection.

Campaign	Mean waiting time (s)	Macro gain vs. FixedTime	Better scenes
Campaign A	11.188	86.66%	80/80
Campaign B	13.183	82.71%	79/80
Campaign C	18.787	70.45%	80/80

Table S13: Final policies from Campaigns A–C on 80 held-out single-target scenes. “Macro gain” is the unweighted mean of the 80 scene-level relative reductions versus paired FixedTime.

The three campaigns use separate authoring realizations and demonstrate distinct admitted-policy trajectories.

Figure S6 and Table S13 use different held-out panels. The figure isolates the full incumbent trajectory on the common 23+23 fixed banks; the table compares the three final policies on the 80-scene panel.

S5.4 Authoring-API Sensitivity

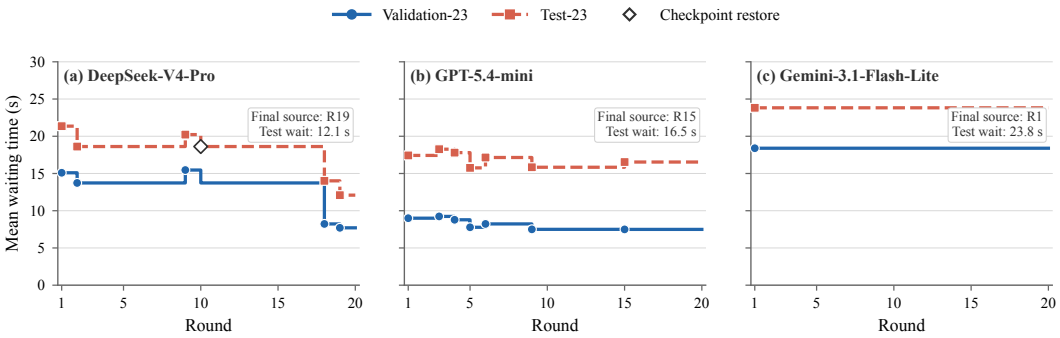


Figure S7: Retrospective fixed-bank trajectories under three authoring models. The plotting contract matches Figure S6: each step is the active frozen incumbent evaluated on the same Validation-23 and sealed Test-23 banks, rather than the round-specific selection score. DeepSeek-V4-Pro, GPT-5.4-mini, and Gemini-3.1-Flash-Lite produce their final admitted policies at rounds 19, 15, and 1, respectively. The round-10 diamond in the DeepSeek panel denotes a checkpoint return to its round-2 controller.

Figure S7 uses the three-city fixed banks to show the complete incumbent trajectories. Table S14 reports the held-out Hangzhou authoring-model panel. The two panels provide complementary sensitivity checks, and their absolute means are not cross-compared.

Authoring model	Mean WT (s) ↓	Mean queue ↓	P95 WT (s) ↓
DeepSeek V4-Pro	1.590	0.288	6.2
GPT-5.4-mini	2.132	0.388	8.4
Gemini 3.1 Flash Lite	9.509	1.737	40.8
GPT-5.6-Sol	1.806	–	9.3

Table S14: Authoring-model sensitivity under the same 20-round workflow and held-out Hangzhou panel. GPT-5.6-Sol used the primary authoring environment; the other three models used a common sensitivity-study environment. Each row reports one 20-round run.

S6. Controller-Call Timing

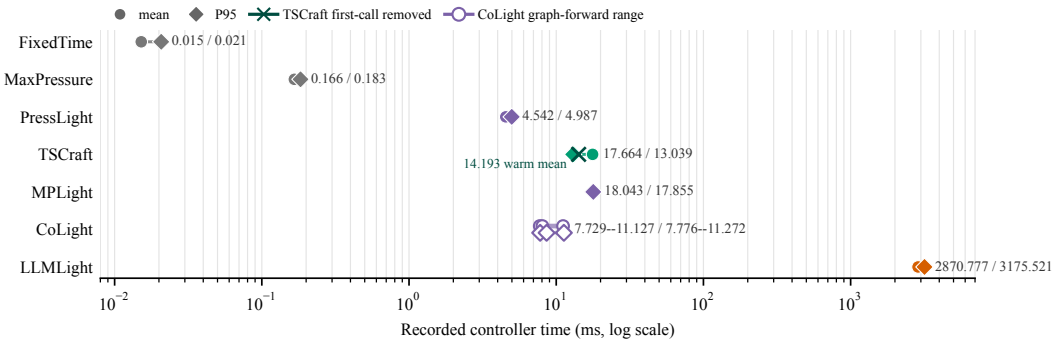


Figure S8: Isolated controller-call timing under one common measurement path. The log axis keeps rule, RL, TSCraft, and LLMLight costs visible together. TSCraft's pooled mean exceeds its P95 because six first calls and five later outliers form a sparse right tail. CoLight is shown with a hollow three-city range because it reports native batched graph-forward time.

Method	Recorded mean (ms)	Recorded P95 (ms)
FixedTime	**0.015**	**0.021**
MaxPressure	0.166	0.183
PressLight	4.542	4.987
TSCraft	17.664	13.039
MPLight	18.043	17.855
CoLight	7.729–11.127	7.776–11.272
LLMLight	2870.777	3175.521

Table S15: Controller-call timing under one common measurement path. TSCraft's mean after removing the first call from each process is 14.193 ms. Six process runs contain 1,539 decisions; cold-start outliers raise the pooled mean above the pooled P95. Bold, single-underlined, and double-underlined entries are first, second, and third among comparable isolated-call methods. CoLight reports its three-city full-network graph-forward range and is unranked.

The reported endpoint is observed software-call time on the measurement host. LLMLight timing includes prompt construction, local model inference, and action parsing. TSCraft timing includes semantic observation construction, serialization, sandbox mailbox I/O, frozen-code execution, and output validation. The mailbox polls at 10 ms intervals, which accounts for most of its roughly 13 ms P95; the arithmetic rule was not timed in isolation. An independent six-run repetition records 17.446 ms mean and 12.901 ms P95 over the same 1,539-call set, closely reproducing the 17.664/13.039 ms result.

CoLight graph	Nodes	Runs	Mean (ms)	P95 (ms)
Jinan	12	3	7.729	7.776
Hangzhou	16	3	8.051	8.599
New York	196	3	11.127	11.272

Table S16: CoLight full-network model-forward measurement. Each run contains 60 graph-forward epochs. Feature construction and model loading are excluded. These batched network values are not inserted into the isolated single-controller column: nodes within one graph pass are neither separate calls nor independent samples.

S7. Position Relative to Traffic-Policy Program Search

Symbolic traffic controllers already provide an important non-LLM precedent. Genetic programming evolves readable phase-urgency trees, GPLight+ imposes a symmetric movement representation, and SymLight uses MCTS to search symbolic phase-priority functions (Liao et al., 2024; 2025a;b). TSCraft instead searches a stateful source program that jointly returns phase and duration, and studies an experience-bearing multi-lineage evolution process. The comparison below positions the four systems by search object, feedback, selection, and deployment scope.

Property	EvolveSignal	TPET	SignalClaw	TSCraft
Search object	Fixed-time Python plan	Lightweight heuristic	Phase-scoring skill	Stateful phase–duration source
Search structure	Program database / MAP-Elites	Top-three retention	Elite plus LLM variants	Capacity-four lineages, two candidates/round
Feedback	SUMO delay/stops	Structured defect traces	Fitness/trend signals	Matched normal/perturbed/ context behavior cards
Selection guard	Executable evaluation	Fitness plus critique	AST/sandbox and fitness	Source/structure gates plus role-masked selection
Runtime LLM	No	No	No	No
Reported deployment scope	One intersection	Three CityFlow datasets	One 4x4 SUMO network	Single intersections and full networks in three cities

Table S17: Design-level positioning from the cited primary papers. TSCraft combines stateful phase–duration policies, matched behavioral evidence, role-masked selection, and single-intersection-to-network deployment (Wang et al., 2025a;c; Lei et al., 2026).