TARO v11 Research Architecture Specification

Status: Proposed
Date: 2026-02-07
Supersedes: `ResearchData/taro_v10_1.md` for offline learning scope only

1. Research Intent

Use sequence-based temporal graph structure learning ideas to improve TARO's offline model compilation quality, while preserving deterministic and contract-safe runtime behavior.

Key distinction from the TGSL paper: - TGSL optimizes temporal link prediction. - TARO must optimize routing outcomes (ETA quality, route quality, safety constraints, deterministic serving).

Therefore v11 is a constrained adaptation, not a direct import.

2. Existing TARO Baseline Strength

Current implementation provides a stable deterministic serving plane: - time contract normalization (`TimeUtils`) - validated spatial runtime (`SpatialRuntime`) - validated profile lookup (`ProfileStore`) - bounded live overlay semantics (`LiveOverlay`)

This is a strong foundation for offline learning because v11 can improve artifacts without destabilizing runtime.

3. Core Hypothesis

If we learn time-aware context over observed edge interaction sequences and use that context to refine profile parameters (and optionally constrained structure), then routing quality improves under sparse/noisy temporal observations while runtime correctness remains unchanged.

4. v11 Learning Architecture

4.1 Inputs

- base topology (`nodes.csv`, `edges.csv`, turn metadata)
- temporal profiles (`profiles.csv`)
- historical telemetry (trip fragments / observed travel times)
- optional incident streams

4.2 Feature Construction

Per edge and transition: - time-of-day bucket - day-of-week bit - historical speed and travel-time stats - transition context (`from_edge -> to_edge`) when available - optional weather/event tags

4.3 Encoder (TGSL-inspired, TARO-constrained)

TGSL idea retained: - edge-centric temporal representation - sequence context prediction

TARO adaptation: - direction-aware edge encoding - turn-feasible transition encoding - fixed-time embedding option for stability and reproducibility

4.4 Candidate Generation

Three candidate families: 1. Profile candidate updates for existing edges (default). 2. Connector-edge proposals within legal topology constraints (opt-in). 3. Research-only drop/suppress candidates (disabled in production by default).

Sampling policy mirrors TGSL spirit but adds routing legality filters: - one-hop and transition-local sampling - constrained multi-hop corridor sampling - bounded random exploration with geometry caps

4.5 Selection Strategy

Training: - can use stochastic exploration (including gumbel-style relaxation).

Export: - deterministic ranking by calibrated score. - thresholding by confidence and safety rules.

No stochastic selection can pass into compiled model output.

5. Objective Design (Routing-Centric)

Combined objective (offline): - supervised travel-time regression loss - route-level ranking/regret loss - profile smoothness and regularization loss - optional contrastive consistency term (original vs refined graph view) - hard penalty for FIFO-unsafe proposals

The objective prioritizes routing error reduction, not link prediction accuracy alone.

6. Constraint Layer (Hard Gates)

Any accepted refinement must satisfy: - schema integrity - topology integrity (if structure changed) - temporal validity - FIFO compliance after profile synthesis - deterministic reproducibility under pinned seed - runtime non-regression guardrails

Rejected candidates are retained with reason code for audit and ablations.

7. Compile Output Contract

Output remains `.taro` model with optional lineage additions: - learning module version - training window - seed - policy hash - validation report hash

Runtime reads deterministic arrays only; no model-serving neural dependency.

8. Evaluation Framework

Primary routing quality metrics: - ETA MAE - ETA MAPE - route-time regret vs held-out observed trips

Reliability metrics: - calibration error - confidence-coverage curves - fallback frequency to base profiles

Safety/system metrics: - FIFO violation count (must be zero post-gate) - A*/Dijkstra parity mismatch count (must be zero) - runtime p95 latency and memory deltas

9. Ablation Grid (Paper-Ready)

1. No learning baseline (v10.1 only)
2. Profile-only learning
3. Profile + structure learning
4. Profile + structure + calibration
5. Candidate strategies: - local-only - local + corridor - local + corridor + random
6. Selection budgets `K`
7. Confidence thresholds
8. FIFO repair enabled vs strict reject

10. Research Contributions Enabled by v11

Potential claims: 1. Learning-augmented routing architecture that preserves deterministic runtime contracts. 2. Constraint-aware adaptation of temporal graph structure learning to travel-time optimization. 3. Empirical evidence that profile-first refinement yields most gains per risk unit. 4. A reproducible build-to-runtime lineage strategy for temporal routing systems.

11. Publication Positioning

Candidate paper theme: - "Constraint-Aware Time-Aware Graph Refinement for Deterministic Routing Engines"

Core novelty should be framed as: - integration of sequence-based temporal structure learning with strict route correctness constraints and production-safe serving contracts.

12. Immediate v11 Research Priorities

1. Implement profile-only learning baseline.

2. Establish deterministic export and audit trail.
3. Run ablations on sparsity/noise stress conditions.
4. Quantify quality gains vs safety/system overhead.
5. Promote structure learning only after stable profile gains.