LARCH: LATENT ATTRIBUTE REASONING THROUGH COMPOSITIONAL HIERARCHIES FOR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommenders predict items through multi-token identifiers. Beyond distinguishing individual items, these identifiers should expose shared relationships useful for prediction. This raises two connected questions: how to organize predictive information into shared components, and how to use those components to supervise intermediate computation and assign policy credit. We introduce LARCH, which derives latent supervision from target prefixes and policy feedback from sampled output words. Attribute-Compatible Set Supervision (ACS) trains continuous latent states through probability mass over entities compatible with the target prefix, without annotated reasoning traces. Structure-Aware Reinforcement Learning assigns token-specific credit according to output-component roles, with prediction-dependent selection of auxiliary latent supervision. For next-POI recommendation, where prediction depends on geographic, semantic, and behavioral relationships among locations, LARCH organizes these signals into shared Mobility Word components while preserving each POI's identity. Across three location-based social network datasets, LARCH improves Acc@1 over a SOTA baseline by an average of 2.52 percentage points. On Amazon Beauty, a Product Word instantiation surpasses the strongest reported baseline, demonstrating cross-domain use of structure-derived latent supervision.

1 INTRODUCTION

Generative recommendation recasts next-item prediction as the autoregressive generation of an item's tokenized identifier (Rajput et al., 2023; Wang et al., 2024). These identifiers can share components across related entities. Recent work adds natural-language reasoning (Lv et al., 2026; Zhuang et al., 2026), continuous latent computation (Hao et al., 2025; Guo et al., 2026; Hu et al., 2026), or reinforcement learning (Lv et al., 2026; Hu et al., 2026).

This raises two questions: which predictive relationships should identifiers share, and how can those components guide intermediate supervision and policy credit? A target prefix defines an Attribute-Compatible Entity Set: entities sharing its ordered components, which need not form a natural taxonomy. Sampled output words reveal component-level successes and errors.

We introduce LARCH—*Latent Attribute Reasoning through Compositional Hierarchies*—to connect compositional representations with intermediate supervision and policy refinement. ACS supervises latent states through compatible-entity probability mass. Structure-Aware Reinforcement Learning assigns policy credit according to output-component roles, supported by prediction-dependent latent supervision.

For next-POI recommendation, the representation must distinguish locations while capturing relationships relevant to the next visit. Nearby POIs may serve different functions; functionally similar places may differ in visitation times and transition patterns. Prior work incorporates semantic and collaborative signals (Wang et al., 2025), strengthens geographic structure (Liu et al., 2025b), or combines spatial hierarchy with functional semantics (Lv et al., 2026). Building on these foundations, our Mobility Word also organizes local context, temporal visitation regularities, and transition patterns into shared components, retaining a distinct identity field. Its components support prediction and supply the prefix relationships used in learning.

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.

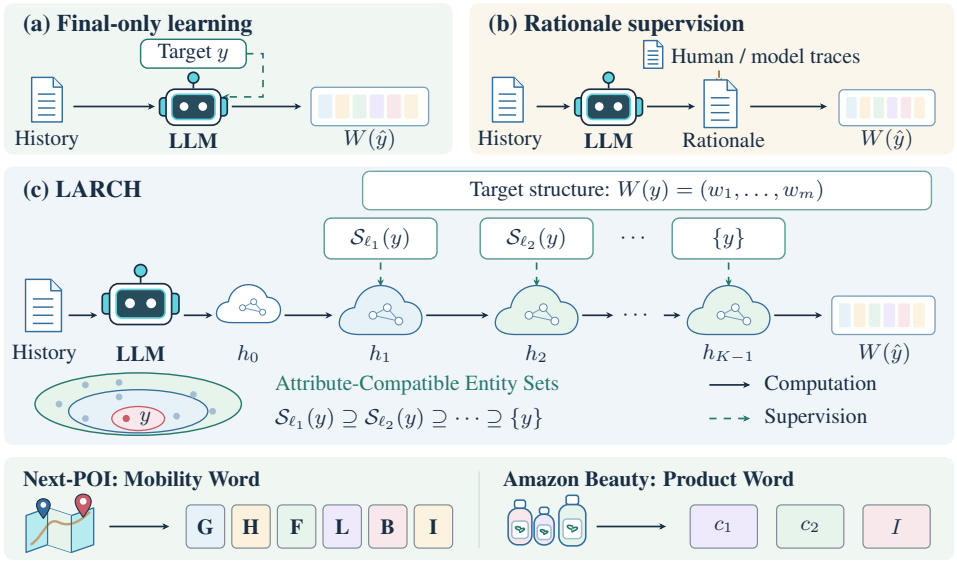

Figure 1: Compositional word prefixes define compatible entity sets for latent supervision across mobility and product domains.

Our contributions are threefold:

- **Structure-derived latent supervision.** We introduce ACS, which turns target-prefix compatibility into supervision for continuous latent states in a common entity space. The same objective covers singleton and multi-entity compatible sets without annotated reasoning traces, with cross-domain instantiations in mobility and Amazon Beauty.
- **Structure-Aware Reinforcement Learning.** We introduce Token-Specific Credit Assignment to align policy feedback with output-component roles. An auxiliary mechanism, Adaptive Latent Supervision, uses sampled predictions to determine when latent supervision is applied and which prefix levels it targets.
- **A Mobility Word for next-POI recommendation.** We construct a six-token representation with shared geographic, functional, local-context, and behavioral components, plus exact identity. Grounding the Mobility Word alone achieves leading next-POI performance; its compositional relationships also support latent supervision and policy feedback.

2 Related Work

Compositional identifiers in generative recommendation. Generative recommenders represent items as discrete token sequences. TIGER constructs semantic identifiers through residual quantization (Rajput et al., 2023); LETTER adds semantic, collaborative, and code-diversity objectives (Wang et al., 2024); and TokenRec quantizes collaborative representations (Qu et al., 2025). ETEGRec jointly trains the tokenizer and recommender (Liu et al., 2025a), while DIGER makes identifier assignment differentiable (Fu et al., 2026). ActionPiece learns reusable patterns over action sequences (Hou et al., 2025). LARCH uses the resulting type of compositional structure as an interface for intermediate supervision and policy feedback.

Structured representations for next-POI recommendation. LLM-based next-POI methods incorporate mobility information through contextual inputs and structured POI representations. LLM4POI serializes contextual check-in information for language-model-based prediction (Li et al., 2024). GNPR-SID quantizes semantic and collaborative POI features (Wang et al., 2025), while GA-LLM introduces hierarchical and Fourier-based geographic representations and aligns POI-transition information with the language model (Liu et al., 2025b). ROS constructs hierarchical spatial-semantic identifiers (Lv et al., 2026), and SPAR combines geographic coordinates with textual semantics in spatially intrinsic identifiers (Wang et al., 2026). These approaches capture com-

plementary aspects of mobility, but do not jointly organize geographic, functional, local, behavioral, transition, and identity information into an ordered representation whose prefix-induced entity groups supervise intermediate computation.

Reasoning and latent supervision. ROS uses mobility reasoning traces and spatially guided reinforcement learning (Lv et al., 2026); Think2Go optimizes reasoning and recommendation with uncertainty- and reward-aware updates (Zhuang et al., 2026). Refine-POI combines topology-aware identifiers with policy gradients (Li et al., 2025), and SIDReasoner aligns identifiers with language before outcome optimization (He et al., 2026). Coconut and ReaRec extend continuous latent computation (Hao et al., 2025; Tang et al., 2025). S^2GR aligns thinking tokens with codebook-cluster centroids (Guo et al., 2026); HiLaR aligns latent states with hierarchical user preferences and assigns process rewards from marginal target-likelihood gains (Hu et al., 2026). ACS instead supervises probability mass over entities compatible with the target prefix, and Stage 3 selects supervision levels based on sampled predictions.

Hierarchical supervision. Hierarchical classification aggregates fine-class probabilities to supervise coarser classes (Muller & Smith, 2020). ACS applies this probability-aggregation principle to entity sets induced by compositional target prefixes, using a common entity readout for intermediate latent states and reusing the objective for outcome-selected supervision during policy refinement.

3 METHOD

LARCH uses a Compositional Output Hierarchy to define sets of entities compatible with target prefixes and organize policy feedback. Entity-word grounding establishes the representation, ACS supervises latent entity distributions at selected prefix levels, and sampled output words guide token-specific credit and latent-supervision selection.

3.1 COMPOSITIONAL OUTPUT HIERARCHY

Compositional Entity Words. Existing generative recommenders commonly represent entities as multi-token semantic or structured identifiers (SIDs), turning next-entity prediction into structured sequence generation (Rajput et al., 2023; Wang et al., 2024). LARCH builds on this established paradigm and abstracts an ordered identifier as a Compositional Entity Word. Its contribution is to expose prefix-induced compatible sets as a shared interface for latent supervision and reinforcement learning.

Let $\mathcal{E}$ be an entity catalog, x a causally available interaction history, and $y \in \mathcal{E}$ the target next entity. LARCH represents each entity e by

$$W(e) = \big(w_1(e), \ldots, w_m(e)\big), \tag{1}$$

whose complete form uniquely identifies e and whose longer prefixes specify additional attributes. For a training Candidate Set $\mathcal{C} = \mathcal{C}(x, y) \subseteq \mathcal{E}$ containing y, prefix length ℓ induces the Attribute-Compatible Entity Set

$$\mathcal{S}_\ell(y; \mathcal{C}) = \{e \in \mathcal{C} : W(e)_{1:\ell} = W(y)_{1:\ell}\}. \tag{2}$$

For supervised levels $\ell_1 < \cdots < \ell_R = m$,

$$\mathcal{S}_{\ell_1}(y; \mathcal{C}) \supseteq \mathcal{S}_{\ell_2}(y; \mathcal{C}) \supseteq \cdots \supseteq \mathcal{S}_{\ell_R}(y; \mathcal{C}) = \{y\}. \tag{3}$$

These sets form the Attribute-Compatible Set Hierarchy. At prefix level ℓ, the event $e \in \mathcal{S}_\ell(y; \mathcal{C})$ means that entity e shares the target's first ℓ word components. A *singleton compatible set* contains only y; a *multi-entity compatible set* also contains other candidates. Complete words give $\mathcal{S}_m = \{y\}$, and a singleton at a shorter prefix remains singleton at longer prefixes. The same ACS objective covers both set cardinalities (Appendix D).

3.1.1 MOBILITY WORD CONSTRUCTION

For next-POI recommendation, LARCH represents each POI p as a six-token Mobility Word

$$W(p) = (G, H, F, L, B, I), \tag{4}$$

Table 1: The six fields of the Mobility Word.

Field	Encoded structure	Construction
G, H	Coarse/fine geography	Balanced hierarchy over occupied S2 cells and deterministic within-region cell indexing.
F	POI function	Train-fitted quantization of category semantics.
L	Local context	Quantized residual address semantics conditioned on G, H, F.
B	Mobility behavior	Quantized temporal, revisit, transition, and travel-distance statistics.
I	Exact identity	Deterministic collision-resolution index within a shared $GHFLB$ group.

which integrates complementary geographic, semantic, behavioral, and identity information. Table 1 summarizes the role and construction of each field; full construction details are in Appendix A.

POIs share component tokens for geographic, semantic, and behavioral attributes; the complete word identifies each POI through the collision-resolving field I. The local component L encodes address information not explained by G, H, F.

The six fields form the autoregressive output sequence, while latent supervision uses three selected structural levels:

$$\underbrace{(G, H)}_{\text{geographic compatibility}} \longrightarrow \underbrace{(G, H, F, L, B)}_{\text{mobility-attribute compatibility}} \longrightarrow \underbrace{(G, H, F, L, B, I)}_{\text{exact POI}}. \tag{5}$$

These levels correspond to geographic compatibility, complete non-identity mobility-attribute compatibility, and exact POI identity, giving $(\ell_1, \ell_2, \ell_3) = (2, 5, 6)$. All learned functional, local-context, and behavioral components are fitted using Train data only; frozen Train-fitted components are reused for catalog entities without Train interactions.

3.1.2 Product Word

For Amazon Beauty, the same abstraction yields a Product Word

$$W(i) = (c_1, c_2, I), \tag{6}$$

where c_1, c_2 form a Train-fitted coarse-to-fine semantic hierarchy and I identifies the product within its leaf. Supervised levels $(1, 2, 3)$ therefore provide coarse-semantic, fine-semantic, and exact-product targets under the same Attribute-Compatible Set formulation, demonstrating that the learning interface is not specific to geographic fields.

3.2 Stage 1: Compositional Entity Word Grounding

Stage 1 grounds the new entity-word vocabulary through bidirectional metadata–word mappings and causal next-entity prediction $x \to W(y)$. Training uses answer-token cross-entropy only. The resulting checkpoint initializes Stage 2 and supplies its frozen output-distribution teacher; no natural-language reasoning traces are required.

3.3 Stage 2: Latent Attribute Reasoning

Stage 2 supervises continuous latent states with sets of entities compatible with the target prefix, using the same ACS objective at each selected prefix level.

Continuous Latent Computation. The model performs K continuous steps $h_0, \ldots, h_{K-1}$, feeding each hidden representation back into the model instead of generating an intermediate text token. For selected states $1 \le s_1 < \cdots < s_R < K$ and hierarchy levels $\ell_1 < \cdots < \ell_R = m$, LARCH assigns

$$h_{s_r} \longleftrightarrow \mathcal{S}_{\ell_r}(y; \mathcal{C}). \tag{7}$$

The number of steps and selected states are instantiation choices rather than fixed properties of LARCH. Output cross-entropy over the complete entity word remains active throughout Stage 2.

Latent Entity Distribution. LARCH maps each supervised state into entity space without a separate classifier. With input embedding matrix E_θ, entity key k_e and the resulting Latent Entity Distribution are

$$k_e = \mathrm{sg}\left[\mathrm{norm}\left(\frac{1}{m}\sum_{j=1}^{m} E_\theta[w_j(e)]\right)\right], \tag{8}$$

$$q_r^{\mathcal{C}}(e \mid x) = \frac{\exp\big(\mathrm{norm}(h_{s_r})^\top k_e/\tau\big)}{\sum_{v\in\mathcal{C}(x,y)} \exp(\mathrm{norm}(h_{s_r})^\top k_v/\tau)}. \tag{9}$$

Here, sg stops gradients through the key branch and τ controls sharpness. Each state has its own auxiliary entity readout $q_r^{\mathcal{C}}$ over the same Candidate Set; gradients through h_{s_r} continue into the backbone.

The Candidate Set is used only to define latent supervision during training and does not constrain autoregressive inference. Its domain-specific construction is provided in the appendix.

Attribute-Compatible Set Supervision. For latent state h_{s_r} associated with hierarchy level ℓ_r, LARCH optimizes

$$\mathcal{L}_{\mathrm{ACS}}^{(r)} = -\log \sum_{e\in\mathcal{S}_{\ell_r}(y;\mathcal{C})} q_r^{\mathcal{C}}(e \mid x). \tag{10}$$

ACS maximizes the probability of the compatible-entity event without prescribing a target distribution within that set or constructing a group representative. Different latent states use the same entity space with events at their assigned prefix levels. Singleton sets recover exact-entity supervision; when all three sets equal $\{y\}$, that example's ACS terms coincide with repeated exact-entity supervision (Full/Full/Full), our control that targets the exact entity at all three supervised states. These properties characterize the supervision imposed at each selected state.

Stage-2 Objective. The complete objective is

$$\mathcal{L}_2 = \mathcal{L}_{\mathrm{CE}} + \sum_{r=1}^{R} \lambda_r \mathcal{L}_{\mathrm{ACS}}^{(r)} + \lambda_{\mathrm{KD}}\mathcal{L}_{\mathrm{KD}}. \tag{11}$$

Here, $\mathcal{L}_{\mathrm{CE}}$ trains complete entity-word generation, the ACS terms supervise selected latent states, and $\mathcal{L}_{\mathrm{KD}}$ preserves the grounded distribution of the frozen Stage 1 teacher. Candidate construction, distillation, and gradient-controller details are in Appendix A.

3.4 Stage 3: Structure-Aware Reinforcement Learning

Stage 3 uses Token-Specific Credit Assignment to align policy feedback with output-component roles. Adaptive Latent Supervision supports this refinement by selecting when and at which prefix levels to apply auxiliary supervision. The policy and frozen reference both initialize from the same Stage 2 checkpoint.

Token-Specific Credit Assignment. A single sequence-level reward assigns identical credit to all output tokens even though different fields encode different structural decisions. LARCH instead aligns policy feedback with the role of each output component. For the Mobility Word,

$$G \to A_G, \qquad H \to A_{H|G}, \qquad (F, L, B, I) \to A_{\mathrm{exact}}. \tag{12}$$

For G, group-relative feedback prioritizes exact matches and otherwise ranks valid geographic predictions by distance to the target region; H receives finer spatial feedback conditional on a correct G. The remaining suffix receives complete-word credit. More generally, Token-Specific Credit Assignment aligns feedback with the structural decision encoded by each component; the feedback function can vary with the Compositional Entity Word.

Adaptive Latent Supervision. Groups of sampled output words also determine the active latent target. All-correct groups receive no auxiliary latent loss; mixed groups supervise all selected levels; and all-incorrect groups activate the earliest level at which every sample fails, conditioned on a match at every preceding level. These comparisons select the supervised prefix levels, while the selected ACS objective remains differentiable. The exact branch logic is given in Appendix A.

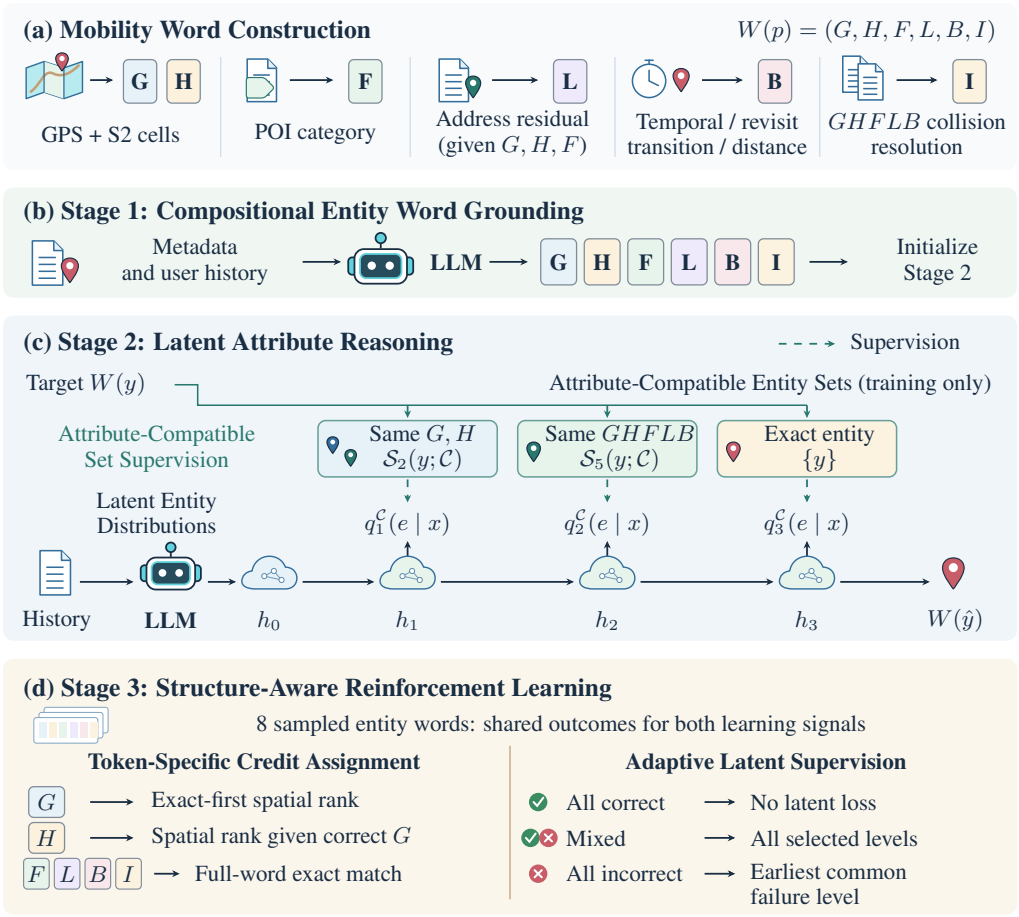

Figure 2: LARCH connects Mobility Word construction, compatible-set latent supervision, and structured policy feedback.

Stage-3 Objective. The complete Stage 3 objective combines token-specific policy optimization, adaptively selected latent supervision, and reference regularization:

$$\mathcal{L}_3 = \mathcal{L}_{\text{exact-suffix}} + \lambda_G \mathcal{L}_{\text{spatial-G}} + \lambda_{H|G} \mathcal{L}_{\text{spatial-H|G}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}} + \beta \mathcal{L}_{\text{rKL}}. \tag{13}$$

The first three terms implement the token-level credit signals in Equation 12. The auxiliary term contains the adaptively selected ACS loss and a lightweight Cross-State Consistency Regularization, while $\mathcal{L}_{\text{rKL}}$ regularizes against the frozen Stage 2 reference model. Appendix A gives the consistency term, spatial ranking, branch logic, and gradient-controller details.

4 Experiments

We evaluate performance across three mobility datasets, then use five research questions to examine representation, latent targets, policy credit, adaptive supervision, and transfer to product recommendation.

4.1 Experimental setup

Datasets and evaluation. For NYC, CA, and TKY, we follow the ROS preprocessing and split protocol, with 1,447, 2,864, and 7,079 evaluation trajectories. Raw Acc@1 requires an exact match of the freely generated six-token word; invalid and unknown words count as errors. Inference uses no catalog projection or candidate reranking. Stage comparisons fix the representation, targets, and decoding.

Amazon 2014 Beauty 5-core follows S^2GR's preprocessing, chronological splits, and Test filtering (Guo et al., 2026), yielding 5,123 users and 12,101 items with histories capped at 20. For three-token Product Words, we report HR@5/10 and NDCG@5/10 using catalog-trie beam search of width 20, alongside free-greedy raw Acc@1. Appendix B specifies the splits and scoring.

Implementation. LARCH uses Qwen3-4B-Base with full-parameter BF16 training. Stage 1 uses $K = 0$ and Stages 2–3 use $K = 4$, for one, two, and one epochs; Beauty uses Stages 1–2. Hardware, hyperparameters, and software versions appear in Appendix C.

Baselines. We reproduce ROS's original three-stage method with its official implementation and Qwen3-4B-Base. Other POI scores follow Lv et al. (2026); Zhuang et al. (2026); Beauty baselines follow Guo et al. (2026), including sequential recommenders, TIGER, and S^2GR. Appendix C gives attribution and reproduction details.

4.2 Overall performance and stage contributions

LARCH achieves 45.68%, 31.39%, and 39.30% raw Acc@1 on NYC, CA, and TKY, respectively (Table 2). With the same 4B backbone, it improves over reproduced ROS by 3.78, 2.16, and 1.63 percentage points, averaging 2.52 points.

Table 2: Next-POI recommendation performance (Acc@1, %).

Method	NYC	CA	TKY
GETNext	24.35	13.57	18.29
TPG	25.55	17.49	14.20
MTNext	26.20	14.53	25.75
STHGCN	27.34	17.30	29.50
ROTAN	31.06	21.99	24.58
ST-GR	29.20	16.59	26.10
LLM4POI	33.72	20.65	30.35
GNPR-SID	36.18	24.03	30.62
GA-LLM	39.19	25.66	34.82
CoAST	40.27	27.21	33.10
Think2Go	38.47	25.83	32.39
ROS†	41.90	29.22	37.67
LARCH	**45.68**	**31.39**	**39.30**

† ROS is reproduced using its official implementation and original method. LARCH follows its preprocessing and splits; both use Qwen3-4B-Base.

Table 3 shows that Mobility Word grounding alone surpasses all compared baselines on NYC and all except ROS on CA and TKY, establishing a strong foundation before latent computation or reinforcement learning. Stages 2 and 3 each improve all three datasets. The following controls examine the representation and supervision choices behind these gains.

Table 3: Raw Acc@1 (%) by training stage.

Training stage	NYC	CA	TKY
LARCH Stage 1 ($K = 0$)	42.16	28.84	35.02
LARCH Stage 2 ($K = 4$)	43.81	30.17	37.96
LARCH Stage 3 ($K = 4$)	**45.68**	**31.39**	**39.30**

4.3 Mechanism analysis

Table 4 organizes the matched NYC controls by training stage. Each comparison fixes the initialization and training protocol within that stage while changing the representation or supervision component under study. Appendix C provides the complete control definitions.

Table 4: Ablation study on NYC (1,447 targets). G and $G{+}H$ count exact prefix matches; Full counts exact whole-word matches.

Component setting	G	$G{+}H$	Full	Acc@1 (%)
Stage 1: Mobility Word construction				
Spatial + identity	907	597	592	40.91
w/o behavioral attributes	899	588	581	40.15
Full Mobility Word	914	614	**610**	**42.16**
Stage 2: Latent Attribute Reasoning				
CE + KD only	910	630	618	42.71
Full/Full/Full targets	908	632	626	43.26
Attribute-Compatible Set Supervision	904	638	**634**	**43.81**
Stage 3: Policy credit and latent supervision				
Shared advantage across all tokens	788	546	544	37.60
w/o latent auxiliary package	577	375	372	25.71
Static all-level supervision (Train-calibrated)	899	638	630	43.54
Full LARCH	923	666	**661**	**45.68**

Horizontal rules separate the three stages; bold marks the full LARCH setting. Stage 2 uses seed-42 two-epoch endpoints and Stage 3 uses one-epoch endpoints. Control definitions appear in Appendix C.

RQ1: Does the Mobility Word itself improve prediction? We compare representation variants after Stage 1 grounding, before latent computation or reinforcement learning. Spatial + identity retains geographic fields; the variant without behavior also retains F/L semantics but removes B. Both recompute identity to preserve unique POI words. The full Mobility Word reaches 42.16% Acc@1, versus 40.91% and 40.15%, respectively. These comparisons support the predictive value of the full attribute composition, including behavioral information beyond F/L semantics. Its shared components also provide the prefix relationships used in later supervision.

RQ2: Do prefix-compatible sets provide useful latent targets beyond repeated exact entities? We separate the value of supervising intermediate states from the choice of supervision targets. In the seed-42 comparison, with latent depth, initialization, teacher, and budget fixed, CE + KD yields 618/1,447 correct predictions, Full/Full/Full reaches 626, and default ACS reaches 634. The first comparison supports direct latent supervision. Since canonical mobility mainly induces singleton compatible sets, the matched $G/GH/\mathrm{Exact}$ control tests earlier-level multi-entity targets.

Changing only the supervised prefix lengths gives 64.952% of Train examples a discriminative multi-POI G set. ACS supervises its total probability mass, retaining an exact terminal target. Across three matched training seeds, $G/GH/\mathrm{Exact}$ improves mean Acc@1 from 43.446% to 43.723% over Full/Full/Full, with gains of 6, 1, and 5 correct predictions on the same 1,447 targets. These modest improvements across all three seeds support prefix-compatible groups as useful intermediate supervision targets (Appendix D.2).

RQ3: Does token-specific credit matter when the feedback signals are retained? To isolate credit allocation, the shared-composite control retains the exact, spatial-G, and conditional-H signals and Adaptive Latent Supervision, but shares one composite advantage across all six tokens. Acc@1 falls from 45.68% to 37.60%. With the feedback sources retained, this comparison supports assigning credit according to each output component's role, beyond simply including spatial feedback in the reward.

RQ4: Why select latent supervision from sampled predictions? We examine the latent auxiliary design within Stage 3. Removing the selected latent objective and consistency regularization reduces Acc@1 to 25.71% and known-word coverage to 71.53%, supporting the auxiliary package's role in accurate, valid generation. We then test whether applying this package at every level on every group suffices.

The Train-calibrated static control retains Token-Specific Credit Assignment and applies all latent levels on every group, with its gradient target calibrated from Train activation statistics. It achieves 100% format validity and 99.10% known-word coverage, versus 99.59% and 99.17% for

full LARCH, but reaches 43.54% Acc@1 versus 45.68%. Thus, even with almost entirely valid catalog outputs, outcome-conditioned latent selection improves accuracy over this static alternative. The comparison evaluates when and at which levels supervision is applied jointly; Appendix C.1 provides branch definitions and diagnostics.

Together, the controls support using compositional structure to define latent targets, allocate policy credit, and select supervision, building on the Mobility Word's predictive attributes.

4.4 Transfer beyond mobility

RQ5: Does prefix-compatible latent supervision transfer to product recommendation? Beauty tests the same ACS objective with semantic Product Words: earlier prefixes define multi-product targets and the full word identifies the exact product. We evaluate this Stage 1–2 instantiation under the product-domain protocol.

Table 5: Top-k recommendation performance on Amazon Beauty. Metrics are fractions; higher is better.

Method	HR@5 ↑	HR@10 ↑	NDCG@5 ↑	NDCG@10 ↑
Caser	0.0314	0.0514	0.0186	0.0251
GRU4Rec	0.0384	0.0634	0.0233	0.0314
SASRec	0.0434	0.0688	0.0259	0.0340
BERT4Rec	0.0394	0.0627	0.0255	0.0329
ReaRec	0.0460	0.0723	0.0284	0.0368
TIGER	0.0443	0.0728	0.0299	0.0390
S^2GR	<u>0.0495</u>	<u>0.0785</u>	<u>0.0340</u>	<u>0.0424</u>
LARCH Stage 1 ($K = 0$)	0.0441	0.0656	0.0324	0.0394
LARCH Stage 2 ($K = 4$)	**0.0607**	**0.0808**	**0.0439**	**0.0503**

LARCH uses the same 5,123 targets and catalog-trie beam evaluator at both stages.

Stage 2 improves HR@5 by 37.6% and NDCG@5 by 35.4% over Stage 1, with gains also at rank 10 (Table 5). Free-greedy raw Acc@1 rises from 1.366% to 2.050%; both stages generate valid catalog words for every target. Stage 2 also exceeds S^2GR, the strongest reported baseline, on all four ranking metrics.

With initialization, latent depth, representation, and budget fixed, ACS improves HR@10 from 0.0742 to 0.0808 and NDCG@10 from 0.0453 to 0.0503 over Full/Full/Full, supporting product-prefix groups as intermediate targets. Prefix-centroid compares group-representative classification with ACS's set-probability aggregation under the same hierarchy. ACS achieves the highest raw exact count and HR@5 among matched controls; centroid is slightly higher on HR@10 and NDCG@10, with nearly identical NDCG@5 (Appendix C.2). These results support transferring prefix-compatible latent supervision beyond mobility, with competitive performance against a hierarchy-aware alternative.

5 Conclusion

LARCH uses compositional target structure to connect intermediate supervision and policy refinement in generative recommendation. ACS supervises continuous latent states through probability mass over prefix-compatible entities, using the same objective for singleton and multi-entity targets. Structure-Aware Reinforcement Learning assigns policy credit according to output-component roles. For next-POI recommendation, the Mobility Word provides a shared-component representation that supports both prediction and these structure-aware learning mechanisms. Across three mobility datasets, the complete pipeline improves Acc@1 over the reproduced ROS baseline by 2.52 percentage points on average, while the Beauty instantiation demonstrates that structure-derived latent supervision can be applied beyond mobility.

AI USE STATEMENT

Generative AI tools were used to assist with software coding and the initial drafting and editing of the manuscript. They were not used to develop the research methodology, design experiments, or interpret experimental results. All AI-assisted code and text were reviewed and verified by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Appendix A specifies the word construction and training objectives, Appendix B defines the evaluation protocols, Appendix C reports hardware, software, hyperparameters, baseline reproduction, and ablation controls, and Appendix D analyzes compatible-set structure under the training candidate pools. Code will be made available after the anonymous review period.

REFERENCES

Shanshan Feng, Feiyu Meng, Lisi Chen, Shuo Shang, and Yew Soon Ong. ROTAN: A rotation-based temporal attention network for time-specific next POI recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 759–770, 2024. doi: 10.1145/3637528.3671809. URL https://doi.org/10.1145/3637528.3671809.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. Differentiable semantic ID for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 369–379, 2026. doi: 10.1145/3805712.3809641. URL https://doi.org/10.1145/3805712.3809641.

Zihao Guo, Jian Wang, Ruxin Zhou, Youhua Liu, Jiawei Guo, Jun Zhao, Xiaoxiao Xu, Yongqi Liu, and Kaiqiao Zhan. S^2GR: Stepwise semantic-guided reasoning in latent space for generative recommendation. *arXiv preprint arXiv:2601.18664*, 2026. URL https://arxiv.org/abs/2601.18664. Version 3; accepted to KDD 2026.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2412.06769.

Yingzhi He, Yan Sun, Junfei Tan, Yuxin Chen, Xiaoyu Kong, Chunxu Shen, Xiang Wang, An Zhang, and Tat-Seng Chua. Reasoning over semantic IDs enhances generative recommendation. *arXiv preprint arXiv:2603.23183*, 2026. URL https://arxiv.org/abs/2603.23183.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06939.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. ActionPiece: Contextually tokenizing action sequences for generative recommendation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24004–24024. PMLR, 2025. URL https://proceedings.mlr.press/v267/hou25f.html.

Guangneng Hu, Yu Zhang, and Qiang Yang. MTNet: A neural approach for cross-domain recommendation with unstructured text. In *KDD Deep Learning Day*, pp. 1–10, 2018. URL https://www.kdd.org/kdd2018/files/deep-learning-day/DLDay18_paper_5.pdf.

Peiyu Hu, Siying Gu, Weihai Lu, Zhuodong Liu, Yuntian Tang, Jiahao Liang, Yiying Xie, Jiang Rong, Zhaokai Luo, Zhiyong Wang, and Jia Wang. Hierarchical latent reasoning for LLM-based recommendation. *arXiv preprint arXiv:2607.27760*, 2026. URL https://arxiv.org/abs/2607.27760.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035. URL https://doi.org/10.1109/ICDM.2018.00035.

Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D. Salim. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024. doi: 10.1145/3626772.3657840. URL https://doi.org/10.1145/3626772.3657840.

Peibo Li, Shuang Ao, Hao Xue, Yang Song, Maarten de Rijke, Johan Barthélemy, Tomasz Bednarz, and Flora D. Salim. Refine-POI: Reinforcement fine-tuned large language models for next point-of-interest recommendation. *arXiv preprint arXiv:2506.21599*, 2025. URL https://arxiv.org/abs/2506.21599.

Haitao Lin, Zhen Yang, Jiawei Xue, Ziji Zhang, Luzhu Wang, Yikun Gu, Yao Xu, and Xin Li. SpaceTime-GR: A spacetime-aware generative model for large-scale online POI recommendation. *arXiv preprint arXiv:2508.16126*, 2025. URL https://arxiv.org/abs/2508.16126.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–739, 2025a. doi: 10.1145/3726302.3729989. URL https://doi.org/10.1145/3726302.3729989.

Zhao Liu, Wei Liu, Huajie Zhu, Jianxing Yu, Jian Yin, Wang-Chien Lee, and Shun Wang. Geography-aware large language models for next POI recommendation. *arXiv preprint arXiv:2505.13526*, 2025b. URL https://arxiv.org/abs/2505.13526.

Yan Luo, Haoyi Duan, Ye Liu, and Fu-Lai Chung. Timestamps as prompts for geography-aware location recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 1697–1706, 2023. doi: 10.1145/3583780.3615083. URL https://doi.org/10.1145/3583780.3615083.

Dongyi Lv, Qiuyu Ding, Heng-Da Xu, Zhaoxu Sun, Zhi Wang, Feng Xiong, and Mu Xu. Reasoning over space: Enabling geographic reasoning for LLM-based generative next POI recommendation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7322–7336, 2026. doi: 10.18653/v1/2026.acl-long.332. URL https://aclanthology.org/2026.acl-long.332/.

Bruce R. Muller and William A. P. Smith. A hierarchical loss for semantic segmentation. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, volume 4, pp. 260–267. SciTePress, 2020. doi: 10.5220/0008946002600267. URL https://www.scitepress.org/PublishedPapers/2020/89460/.

Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. TokenRec: Learning to tokenize ID for LLM-based generative recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 37 (10):6216–6231, 2025. doi: 10.1109/TKDE.2025.3599265. URL https://doi.org/10.1109/TKDE.2025.3599265.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.05065.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Jian Wu, and Yuning Jiang. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675*, 2025. URL https://arxiv.org/abs/2503.22675.

Jiaxi Tang and Ke Wang. Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, 2018. doi: 10.1145/3159652.3159656. URL https://doi.org/10.1145/3159652.3159656.

Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. Generative next POI recommendation with semantic ID. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2904–2914, 2025. URL https://arxiv.org/abs/2506.01375.

Fangye Wang, Yunjin Gu, Haowen Lin, Yifang Yuan, Song Yang, Xiaojiang Zhou, and Pengjie Wang. SPAR: Enhancing industrial-scale generative POI recommendation via real-world spatial perception. *arXiv preprint arXiv:2609.02062*, 2026. URL https://arxiv.org/abs/2609.02062.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024. doi: 10.1145/3627673.3679569. URL https://doi.org/10.1145/3627673.3679569.

Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. Spatio-temporal hypergraph learning for next POI recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 403–412, 2023. doi: 10.1145/3539618.3591770. URL https://doi.org/10.1145/3539618.3591770.

Song Yang, Jiamou Liu, and Kaiqi Zhao. GETNext: Trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1144–1153, 2022. doi: 10.1145/3477495.3531983. URL https://doi.org/10.1145/3477495.3531983.

Penglong Zhai, Jie Li, Fanyi Di, Yue Liu, Yifang Yuan, Jie Huang, Peng Wu, Sicong Wang, Mingyang Yin, Tingting Hu, Yao Xu, and Xin Li. Cognitive-aligned spatio-temporal large language models for next point-of-interest prediction. *arXiv preprint arXiv:2510.14702*, 2025. URL https://arxiv.org/abs/2510.14702.

Zhuang Zhuang, Shanshan Feng, Hangwei Qian, Mingqi Yang, Heng Qi, Yanming Shen, and Baocai Yin. Think2Go: Generative next POI recommendation with LLM reasoning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2026. doi: 10.1145/3770854.3780334. URL https://arxiv.org/abs/2607.28997.

A Representation and Training Details

A.1 Entity-word construction

Geographic attributes. Each POI is mapped to its occupied level-30 S2 cell. Instead of using two fixed S2 parent resolutions directly, LARCH recursively partitions occupied cells into approximately balanced geographic regions. $G(p)$ identifies the resulting coarse region, while $H(p)$ identifies the occupied cell within that region using deterministic S2 order. This construction balances the number of occupied cells assigned to coarse regions while retaining a deterministic coarse-to-fine geographic description.

Functional and local-context attributes. Let $z_F(p)$ be the frozen text-encoder embedding of the canonical POI category. Functional centroids are fitted on categories observed among Train POIs, and

$$F(p) = \arg\min_k \left\| z_F(p) - \mu_k^F \right\|_2^2 . \tag{14}$$

An unseen catalog category is assigned by the frozen Train-fitted quantizer without refitting. For local context, let $z_A(p)$ be the frozen address embedding and let $f_{\text{ridge}}(G, H, F)$ be a Train-fitted additive ridge predictor over one-hot representations of the preceding attributes. LARCH computes

$$r_A(p) = \text{norm}(z_A(p) - f_{\text{ridge}}(G, H, F)) , \qquad L(p) = \arg\min_k \left\| r_A(p) - \mu_k^L \right\|_2^2 . \tag{15}$$

The residual construction makes L complementary to the preceding geographic and functional fields.

Behavioral and identity attributes. For each Train-observed POI, the behavioral representation is

$$b_p = \left[b_p^{\text{time}}; b_p^{\text{visit}}; b_p^{\text{flow}}; b_p^{\text{distance}}; b_p^{\text{svd}} \right] , \tag{16}$$

covering temporal visitation patterns, check-in and revisit statistics, transition degree and entropy, travel-distance statistics, and low-rank representations of incoming and outgoing transitions. It is standardized with Train statistics and quantized to obtain $B(p)$; POIs without Train interactions receive a fixed cold-behavior code. Finally, I is a deterministic index among POIs sharing the same $GHFLB$ prefix, resolving residual collisions without adding another mobility attribute.

Product Word. For Beauty, a frozen text encoder maps product titles and available descriptions to semantic representations. A Train-fitted balanced semantic hierarchy supplies coarse and fine codes c_1, c_2, and I deterministically identifies a product within the resulting leaf. The same Candidate Set and Attribute-Compatible Set definitions then apply without geographic or mobility-specific losses.

A.2 Stage-specific details

Stage 1. For a grounding or sequential-prediction prompt u with answer a, only answer tokens contribute to the standard autoregressive objective

$$\mathcal{L}_1 = -\mathbb{E}_{(u,a)\sim\mathcal{D}_1} \sum_t \log p_\theta(a_t \mid u, a_{<t}). \tag{17}$$

Stage 2. The completed experiments use four continuous positions and supervise h_1, h_2, h_3, leaving h_0 without a direct ACS term. Mobility Candidate Sets contain the unique POIs in the visible Train history plus the gold target; Beauty uses all Train-observed catalog items. These sets define training supervision only and never constrain autoregressive inference. Entity keys are recomputed from the current input embeddings and detached on the key branch. The frozen Stage 1 teacher supplies cached output distributions for distillation; Table C.1 reports the candidate, temperature, and gradient-target settings.

Stage 2 distillation and loss-weight controller. The frozen Stage 1 teacher is evaluated with $K = 0$, and a Train example is cached only when its greedy complete-word prediction exactly matches the target. At each cached valid answer position, $\mathcal{L}_{\text{KD}}$ minimizes $\text{KL}(q_T^{\mathcal{B}} \| q_\theta^{\mathcal{B}})$ at temperature 1, where $\mathcal{B}$ partitions the vocabulary into the teacher's top-128 tokens and one aggregate tail bucket. The loss is averaged over valid answer positions; an uncached example contributes zero KD loss while its CE and ACS terms remain active.

All Stage 2 auxiliary coefficients are calibrated with sparse gradient probes at the shared initial continuous state h_0, not with parameter-gradient norms. Let $g_0 = \|\nabla_{h_0} \mathcal{L}_{\text{CE}}\|_{\text{RMS}}$ and $g_j = \|\nabla_{h_0} \mathcal{L}_j\|_{\text{RMS}}$, where the RMS is the square root of the distributed mean of per-row squared norms. For target ratio r_j, the proposed coefficient is

$$\widehat{\lambda}_j = r_j \frac{g_0}{g_j + 10^{-8}}. \tag{18}$$

The first valid calibration clips this value to the branch bound. Later calibrations first clip the proposal, apply an exponential moving average with coefficient 0.9, and limit the new coefficient to

$[0.5\lambda_j, 2\lambda_j]$ before a final bound clip. The bounds for region, context, exact, and KD are respectively $[0.001, 0.50]$, $[0.001, 0.75]$, $[0.001, 1.00]$, and $[0, 0.50]$. Calibration occurs at optimizer step 1, every 50 steps in epoch 1, and every 100 steps thereafter. The current coefficients score the current optimizer step, and an update applies from the next step. If a branch has no active examples or its probe is absent, non-finite, or at most 10^{-12}, its coefficient is unchanged.

Stage 3 policy credit. For each eight-sample group, the G token uses the ordered key $(0, 0)$ for an exact region, $(1, d_{\text{hav}})$ for a valid wrong region, and $(2, \infty)$ for an invalid region token. Unique keys are sorted lexicographically and converted to descending integer quality ranks; identical keys remain tied. The ranks are standardized within the group with the population standard deviation. If the standard deviation is below 10^{-8}, the branch is inactive. The resulting advantage is applied only to G. The H branch applies the same rule only to samples whose G is correct, using valid (G, H) cell centers and applying its advantage only to H; fewer than two eligible samples or no rank variation makes it inactive. Complete-word hit indicators are standardized separately and applied only to F, L, B, I. These disjoint policy terms are then combined as in Equation 13.

Adaptive latent branch. For an all-correct sampled group, the auxiliary latent objective is inactive. A mixed group activates all selected ACS levels. For an all-incorrect group, LARCH selects the earliest supervised level ℓ_r at which no sampled completion matches the target prefix, provided that every preceding supervised level has at least one matching completion. For the Mobility Word, the ordered tests are GH, $GHFLB$, and the complete $GHFLBI$ word.

Cross-State Consistency Regularization. Let $A_{\ell_r} q$ aggregate an entity distribution by prefixes of length ℓ_r. For adjacent supervised states,

$$J_r = \text{JS}\left(A_{\ell_r} q_r^{\mathcal{C}}, A_{\ell_r} q_{r+1}^{\mathcal{C}}\right), \qquad r = 1, \ldots, R-1. \tag{19}$$

This compares adjacent states at their shared coarse granularity while allowing the finer state to redistribute mass within each compatible group. The applicable consistency terms are included in $\mathcal{L}_{\text{aux}}$ together with the branch-selected ACS loss; the controller targets used in the completed runs appear in Table C.1.

Stage 3 reference regularization and loss-weight controller. At each of the six Mobility Word positions, $\mathcal{L}_{\text{rKL}}$ is the full-vocabulary $\text{KL}(p_\theta \| p_{\text{S2}})$ computed in FP32 and averaged over sampled tokens; EOS is excluded. Gradients flow only through the policy, while the Stage 2 reference remains frozen. Its coefficient is fixed at $\beta = 0.01$.

The adaptive coefficients again use probes with respect to h_0. Let E denote the exact-suffix policy branch and $j \in \{\text{aux}, G, H \mid G\}$. From each controller window, branch activity frequencies f_j and distributed RMS probe norms $\bar{g}_j$ give

$$\widehat{\lambda}_j = r_j \frac{f_E \bar{g}_E}{f_j \bar{g}_j + 10^{-8}}, \qquad (r_{\text{aux}}, r_G, r_{H|G}) = (0.15, 0.025, 0.025). \tag{20}$$

Valid probe norms use an exponential moving average with coefficient 0.9. The coefficient bounds are $[10^{-4}, 0.10]$ for the auxiliary branch and $[10^{-4}, 0.08]$ for each spatial branch. A decrease may move directly to the bounded estimate, whereas an increase is limited to twice the previous coefficient. A missing primary probe freezes all three coefficients; a missing or inactive auxiliary/spatial probe freezes only that branch. Updates occur every 50 optimizer steps in the one-epoch run and apply to the next step. The formal run starts from the Train-only smoke calibration $(\lambda_{\text{aux}}, \lambda_G, \lambda_{H|G}) \approx (0.01152, 0.003600, 0.01804)$.

B Evaluation definitions

For each target y_u and freely generated parsed entity $\hat{y}_u$, raw accuracy is $N^{-1} \sum_u \mathbf{1}[\hat{y}_u = y_u]$; invalid or unknown words are errors. For the distinct catalog-ranked list, let r_u be the rank of y_u, or infinity if absent. With one relevant next item per row,

$$\text{HR@}k = \frac{1}{N} \sum_u \mathbf{1}[r_u \le k], \qquad \text{NDCG@}k = \frac{1}{N} \sum_u \frac{\mathbf{1}[r_u \le k]}{\log_2(r_u + 1)}. \tag{21}$$

Beauty data processing. We follow the same preprocessing steps as S^2GR on Amazon 2014 Beauty 5-core. Interactions retain chronological order for all 22,363 users over 12,101 items. The last interaction of each user is the Test target and the penultimate interaction is the Validation target. The remaining interactions form Train sequences, yielding 131,413 next-product examples by predicting each item from its preceding interactions. Validation histories contain the Train sequence, and Test histories additionally include the Validation interaction. Histories are capped at the most recent 20 items. The Test-only filter retains 5,123 users whose complete, untruncated sequence contains at least ten interactions; shorter sequences remain in Train and Validation.

Beauty ranking. Catalog-trie beam search uses width 20. Candidate scores sum full-vocabulary log probabilities of the three word tokens and exclude EOS; beam search approximates catalog ranking rather than enumerating its exact global top-k. No sampled negative set, history-only ranking restriction, or target injection is used. The training candidate catalog excludes the 33 Train-unseen items, while the inference catalog includes all 12,101 items. Metadata grounding can include those items' static descriptions; this is distinguished from training on their withheld interactions.

C Implementation and Result Provenance

The backbone is Qwen3-4B-Base (https://huggingface.co/Qwen/Qwen3-4B-Base). The frozen text encoder is Qwen3-Embedding-0.6B (https://huggingface.co/Qwen/Qwen3-Embedding-0.6B); the Beauty cache contains 1,024-dimensional normalized vectors. These encoders are distinct from the candidate keys in Equation 8. Table C.1 lists shared and domain-specific settings.

Hardware and optimization. The reported LARCH training runs use eight PPU-ZW810E accelerators with 96 GB of memory each, full-parameter BF16 computation, and DeepSpeed ZeRO-2. Table C.2 gives the stage-specific hyperparameters. The software environment uses Python 3.12.3, PyTorch 2.9.0 with PPU runtime 2.0.0, Transformers 4.57.1, and DeepSpeed 0.14.4. The mobility Stage 3 recipe is shared across cities: only dataset-dependent epoch lengths and paths change. Beauty evaluates Stages 1–2 with the same learning rates and latent-supervision targets, using gradient accumulation to obtain a global batch of 32.

ROS reproduction. We reproduce ROS using its official implementation with Qwen3-4B-Base as the student and Qwen3-235B-A22B as the teacher, following the paper's three-stage training pipeline. Where the released code and paper differ, we follow the reward and clipping formulations reported in the paper. Because the original software environment cannot be reconstructed exactly from the released artifacts, our reproduction uses the disclosed compatible dependencies. We report the mean of three inference passes from one trained checkpoint per dataset, obtaining 41.90%, 29.22%, and 37.67% Acc@1 on NYC, CA, and TKY, respectively. The corresponding published ROS results are 44.78%, 31.49%, and 38.64%. The two systems use the same target splits and Qwen3-4B backbone scale, while retaining their respective representations and reasoning procedures. We therefore use our complete-pipeline reproduction for the controlled comparison in Table 2 and report the published ROS results separately for reference. **The remaining discrepancy may reflect software, hardware, or training stochasticity, and cannot be isolated from the publicly available artifacts alone.**

Reported baseline attribution. For the rows imported from the ROS comparison, the original references for GETNext, TPG, MTNet, STHGCN, ROTAN, SpaceTime-GR, and CoAST are Yang et al. (2022); Luo et al. (2023); Hu et al. (2018); Yan et al. (2023); Feng et al. (2024); Lin et al. (2025); Zhai et al. (2025); the remaining POI methods are cited in the main text. ROS prints "MT-Next" in its performance table but associates that row with the MTNet reference in its baseline description, so Table 2 preserves the reported label. For the Beauty comparison, the original references for Caser, GRU4Rec, SASRec, and BERT4Rec are Tang & Wang (2018); Hidasi et al. (2016); Kang & McAuley (2018); Sun et al. (2019); ReaRec, TIGER, and S^2GR are cited in the main text.

Checkpoint convention. Each LARCH result uses the checkpoint at the end of the prescribed stage schedule. The CA sequence has 826/864/899 correct targets out of 2,864, TKY has

Table C.1: Representation and supervision settings. Prefix lengths count word fields; history limits count interactions. Controller ratios target latent-state gradient-norm budgets.

Component	Mobility	Beauty
Backbone	Qwen3-4B-Base	Qwen3-4B-Base
Word length	6	3
Stage 2 continuous positions	4	4
Supervised states	h_1, h_2, h_3	h_1, h_2, h_3
Assigned prefix lengths	2, 5, 6	1, 2, 3
Stage 1 / 2 history limits	300 / 100	20 / 20
Test history limit	300	20
Set-loss candidates	Unique Train history plus gold	All 12,068 Train items
Keys	Detached current input-embedding means	Same rule, 3 fields
Set temperature τ	0.10	0.10
Prefix gradient targets	0.10 / 0.20 / 0.30	0.10 / 0.20 / 0.30
Distillation gradient target	0.15	0.15
Stage 2 epochs	2	2
Stage 3	One epoch; eight samples	—

Table C.2: Training hyperparameters. S1/S2/S3 denote the three stages. Global batches count trajectories, before Stage 3 sampling.

Setting	Mobility	Beauty
Accelerators	8 PPU-ZW810E	8 PPU-ZW810E
Memory per accelerator	96 GB	96 GB
Training / precision	Full parameter / BF16	Full parameter / BF16
Distributed strategy	DeepSpeed ZeRO-2	DeepSpeed ZeRO-2
Optimizer / seed	AdamW / 42	AdamW / 42
Epochs (S1 / S2 / S3)	1 / 2 / 1	1 / 2 / –
Global batch size	8	32
Per-device batch size	1	1
Gradient accumulation	1	4
S1 peak learning rate	10^{-5}	10^{-5}
S2 peak learning rate	2×10^{-6}	2×10^{-6}
S3 peak learning rate	1.05×10^{-6}	–
Learning-rate schedule	Cosine	Cosine
Warmup steps (S1 / S2)	100 / 50	100 / 50
Weight decay (S1 / S2 / S3)	0.1 / 0.1 / 0	0.1 / 0.1 / –
Gradient clipping norm	1.0	1.0
S3 samples per trajectory	8	–
S3 sampling temperature	1.0	–
S3 policy clip / KL coefficient	0.2 / 0.01	–
S3 latent-state gradient targets (latent / G / H)	0.15 / 0.025 / 0.025	–

2,479/2,687/2,782 out of 7,079, and NYC has 610/634/661 out of 1,447. Each Stage 3 run initializes from the corresponding Stage 2 checkpoint.

Matched NYC component controls. The Stage 1 reduced-word controls use the same base model, Train data, seed, decoding, and Test protocol as the full Mobility Word reference. Spatial + identity sets $F/L/B$ to null and recomputes I within GH; w/o behavioral attributes retains the $G/H/F/L$ assignments, sets B to null, and recomputes I within $GHFL$. Both mappings contain 5,099 unique six-token words and use no validation or Test interactions for fitting. They score 592 and 581, versus 610 for the full Mobility Word.

The Stage 2 control recipe starts from the Stage 1 checkpoint (610/1,447), using the same word mapping, 11,022 Train trajectories, teacher cache, four continuous states, persistent key–value com-

putation, and Test inputs. Training uses seed 42, global batch 8, peak learning rate 2×10^{-6}, and two epochs (2,756 updates); the distillation gradient target remains 0.15. The no-set control removes the three prefix objectives; Full/Full/Full replaces their positive sets while retaining the prefix gradient targets. CE + KD only, Full/Full/Full, and ACS yield 618, 626, and 634 correct predictions, respectively (Table 4). ACS yields eight more correct predictions than Full/Full/Full in this seed-42 comparison; Appendix D.2 reports the three-seed comparison of coarser-prefix ACS and Full/Full/Full.

C.1 Matched Stage-3 controls and diagnostics

All controls initialize policy and frozen reference from the same NYC Stage 2 checkpoint (634/1,447). They retain $K = 4$, seed 42, sampling, data, optimizer, reverse KL, the one-epoch cosine schedule (1,378 updates), and the canonical decoder and targets. Half-epoch checkpoints are diagnostics; the endpoint is the primary comparison.

Shared composite advantage. This control preserves the exact, spatial-G, and conditional-H feedback signals, their component standardization, adaptive latent supervision, and consistency. It changes where policy credit is assigned. For component $k \in \{\text{exact}, G, H \mid G\}$, let m_{ik} mark active rollout i, $n_k = \sum_i m_{ik}$, and U count rollouts active in at least one component. With $\lambda_{\text{exact}} = 1$, define

$$\bar{A}_i = \sum_{k:n_k>0} \lambda_k \frac{U}{n_k} m_{ik} A_{ik}, \qquad A_{it}^{\text{shared}} = \bar{A}_i, \quad t = 1, \ldots, 6. \tag{22}$$

The policy loss averages over the $6U$ active token positions. The exposure factors preserve component-wise active-rollout normalization while redistributing credit across all six positions. The controller rules and gradient targets remain fixed. The endpoint scores 544 versus 661 for full LARCH. An additional exact-only shared-advantage control, which also removes spatial feedback, scores 509. Thus the shared-composite comparison separates credit allocation from simply retaining spatial reward information.

Without latent auxiliary and consistency. This control preserves token-specific exact/G/$H \mid G$ policy optimization and latent computation, but removes all selected latent NLL and Cross-State Consistency Regularization contributions. The unused auxiliary weight and controller probe are zero; the G and $H \mid G$ controller branches remain active. Its endpoint scores 372. This control tests the contribution of the latent auxiliary package.

Train-calibrated static all-level supervision. This control retains Token-Specific Credit Assignment but replaces outcome-conditioned activation and granularity with the mean of all three latent target losses and their consistency terms on every sampled group. To adjust for its always-on exposure without consulting evaluation results, we predeclare its latent gradient target from Train statistics: full LARCH activates latent supervision on 7,588 of 11,024 Train groups, giving $0.15 \times 7{,}588/11{,}024 = 0.103$, rounded to 0.10. The control reaches 630 at the endpoint, compared with 661 for full LARCH, and directly tests adaptive selection against a static alternative at Train-calibrated strength.

Table C.3: Complete NYC Stage-3 controls: correct counts and endpoint accuracy.

Setting	Half epoch	One epoch	Acc@1 (%)
Exact-only shared advantage	489	509	35.18
Shared advantage across all tokens	518	544	37.60
Without latent auxiliary package	362	372	25.71
Static all-level (Train-calibrated)	600	630	43.54
Full LARCH	649	661	45.68

All counts use 1,447 targets. Half-epoch results are diagnostics, not checkpoint selection.

Table C.4: Structured generation on NYC (1,447 targets; all values in %).

Configuration	Acc@1	Format-valid	Known-word
Stage 2 initializer	43.81	99.24	99.10
Exact-only shared advantage	35.18	87.08	86.66
Shared advantage across all tokens	37.60	89.15	89.01
Without latent auxiliary package	25.71	71.80	71.53
Static all-level (Train-calibrated)	43.54	100.00	99.10
Full LARCH Stage 3	45.68	99.59	99.17

Format-valid outputs have the required six-field structure; known words map to catalog POIs. Stage 3 rows use one-epoch endpoints.

Interpretation of the Stage-3 controls. The controls test distinct parts of Structure-Aware Reinforcement Learning. The shared-composite control preserves the exact, spatial-G, and conditional-H feedback signals together with Adaptive Latent Supervision, while changing their allocation across output tokens. It therefore evaluates token-specific credit allocation with the spatial feedback retained. The no-auxiliary control evaluates the latent auxiliary package as a whole by removing both the selected latent objective and consistency regularization; its accuracy decline accompanies reductions in format validity and known-word coverage. The Train-calibrated static control retains Token-Specific Credit Assignment and achieves 100% format validity and 99.10% known-word coverage, compared with 99.59% and 99.17% for full LARCH. Its 43.54% Acc@1 versus 45.68% for full LARCH supports outcome-conditioned Adaptive Latent Supervision over the evaluated static alternative. This comparison assesses activation and granularity selection jointly under the respective adaptive and Train-calibrated static supervision schedules.

Evaluator and independent metric recomputation agree, and full-run audits verify that each intended treatment was applied.

C.2 Matched Beauty supervision controls

Stage 1 uses $K = 0$; all Stage 2 controls use $K = 4$, the same Stage 1 initialization, Product Word mapping, Train examples, frozen teacher cache, and 12,068-item Train candidate pool. They retain seed 42, global batch 32, peak learning rate 2×10^{-6}, optimizer, two-epoch schedule (8,214 updates), and endpoint evaluation on the same 5,123 users. The CE+KD-only control disables all three set losses and their control branches while retaining the original KD rule and gradient target 0.15. Full/Full/Full changes only the three positive sets to exact-entity targets, retaining the auxiliary controller rules. Ranking uses the same full-catalog trie beam search as the main comparison.

Table C.5: Matched Stage 2 supervision controls on Beauty.

Supervision	Raw	HR@5	HR@10	NDCG@5	NDCG@10
CE + KD only	90	0.0568	0.0726	0.0394	0.0444
Full/Full/Full targets	96	0.0584	0.0742	0.0403	0.0453
Prefix-centroid	101	0.0597	0.0820	0.0438	0.0510
LARCH Stage 2 (ACS)	105	0.0607	0.0808	0.0439	0.0503

Raw counts exact free-greedy predictions among 5,123 targets; ranking metrics use catalog-trie beam search.

Prefix-centroid supervision. This control compares two ways of using the same prefix hierarchy for latent supervision: entity-set probability aggregation and prefix-centroid classification. For each prefix group g in the unchanged Candidate Set, form a normalized centroid from the detached entity keys in Equation 8:

$$c_g = \mathrm{norm}\left(\frac{1}{|g|}\sum_{e\in g} k_e\right), \qquad \mathcal{L}^{(r)}_{\mathrm{centroid}} = -\log \frac{\exp(\mathrm{norm}(h_{s_r})^\top c_{g(y)}/\tau)}{\sum_{g'}\exp(\mathrm{norm}(h_{s_r})^\top c_{g'}/\tau)}. \tag{23}$$

Each present group appears once in the denominator. The control replaces only h_1/h_2 supervision, retains exact-entity supervision at h_3, and uses the same temperature $\tau = 0.10$. Beauty has 64 coarse and 1,024 fine groups; centroids are recomputed from detached entity keys each microbatch. Initialization, data, teacher cache, CE/KD, latent computation, controller targets, seed, and training budget match ACS. This implements prefix-centroid alignment within LARCH; the external S^2GR method additionally uses its own tokenizer and interleaved reasoning architecture.

ACS achieves 105 raw exact predictions and HR@5 of 0.0607, compared with 101 and 0.0597 for prefix-centroid supervision. Prefix-centroid reaches HR@10 of 0.0820 and NDCG@10 of 0.0510, compared with 0.0808 and 0.0503 for ACS; NDCG@5 is 0.0438 versus 0.0439. Both hierarchical objectives exceed the CE+KD-only and Full/Full/Full controls on all reported metrics in these runs.

Paired evaluation. Table C.6 reports 95% intervals from 10,000 paired bootstrap resamples of the 5,123 Test rows (seed 20260924), using one trained model per objective. Raw outcomes include 21 centroid-only and 25 ACS-only correct predictions; exact two-sided McNemar $p = 0.6587$. The intervals quantify paired Test-row uncertainty for these checkpoints.

Table C.6: Paired Beauty differences: prefix-centroid minus ACS. All metrics are fractions.

Metric	Difference	95% paired bootstrap interval
Raw Acc@1	-0.000781	[-0.003318, +0.001757]
HR@5	-0.000976	[-0.004685, +0.002733]
HR@10	+0.001171	[-0.002928, +0.005466]
NDCG@5	-0.000066	[-0.002443, +0.002325]
NDCG@10	+0.000690	[-0.001608, +0.003034]

External comparison. S^2GR scores are paper-reported: its recommender uses a four-layer encoder and four-layer decoder T5 architecture (Guo et al., 2026), whereas LARCH uses Qwen3-4B-Base. The external comparison evaluates overall recommendation performance; the matched controls compare supervision objectives within LARCH.

D Compatible-set structure

D.1 Statistics under actual training candidate pools

The statistics in Tables D.7–D.8 weight each original Train example equally, before distributed padding. They describe the actual positive sets used for supervision, not prefix prediction accuracy or full-catalog reranking. Write $S^{(r)} = \mathcal{S}_{\ell_r}(y; \mathcal{C})$ for the three supervised sets. Mobility uses $(\ell_1, \ell_2, \ell_3) = (2, 5, 6)$ and the deduplicated visible Train history, capped at 100 interactions, union the target. Beauty uses $(1, 2, 3)$ and all 12,068 Train-observed products. Because the complete word is unique, $S^{(3)} = \{y\}$ throughout. A set covering the entire candidate pool has zero set NLL and supplies no discriminative gradient.

Table D.7: Mean candidate-pool and positive-set sizes on Train.

Dataset	Examples	$\lvert\mathcal{C}\rvert$	$\lvert S^{(1)}\rvert$	$\lvert S^{(2)}\rvert$	$\lvert S^{(3)}\rvert$
NYC	11,022	15.6921	1.0328	1.0104	1.0000
CA	36,319	23.5194	1.0021	1.0002	1.0000
TKY	51,610	24.1034	1.0003	1.0000	1.0000
Beauty	131,413	12068.0000	188.5598	11.7928	1.0000

Effective supervision granularity. All 131,413 Beauty examples have $|S^{(1)}| \in \{188, 189\}$, $|S^{(2)}| \in \{11, 12\}$, and $|S^{(3)}| = 1$, so both inclusions are strict. In mobility, GH identifies a single occupied level-30 cell and usually an individual POI; restricting candidates to history can remove additional compatible entities. For NYC, 4,924 of 5,099 catalog POIs (96.57%) already

Table D.8: Singleton and equality rates on Train (%).

Dataset	$\lvert S^{(1)}\rvert = 1$	$\lvert S^{(2)}\rvert = 1$	$S^{(1)} = S^{(2)}$	$S^{(2)} = S^{(3)}$	All equal	Both strict
NYC	97.859	99.093	98.476	99.093	97.859	0.290
CA	99.785	99.981	99.805	99.981	99.785	0.000
TKY	99.973	100.000	99.973	100.000	99.973	0.000
Beauty	0.000	0.000	0.000	0.000	0.000	100.000

"Both strict" denotes two proper inclusions. $S^{(3)}$ is always a singleton compatible set. Rates include examples with a one-entity candidate pool.

have one-POI GH groups in the full catalog. Under the actual training pools, all three sets coincide on 10,786/11,022 NYC examples (97.859%), 36,241/36,319 CA examples (99.785%), and 51,596/51,610 TKY examples (99.973%). Two strict reductions occur on only 32 NYC examples and none in CA or TKY. These set structures reflect the joint choice of entity representation and training candidate pool.

Interpreting singleton and multi-entity supervision. ACS is defined over prefix-induced compatible events. Because the complete entity word uniquely identifies the target, the terminal supervision level is necessarily singleton; earlier levels may induce either singleton or multi-entity sets depending on the representation, selected prefix, and Candidate Set. When $S_\ell(y; \mathcal{C}) = \{y\}$, the corresponding ACS term reduces exactly to exact-entity supervision. The same objective thus covers both singleton and multi-entity compatible sets. Canonical mobility evaluates ACS predominantly in the singleton-compatible case, whereas the coarser NYC setting and Beauty include examples with multi-entity earlier-level compatible sets, on which ACS is mathematically distinct from repeated exact-entity supervision. These settings serve complementary roles: canonical mobility evaluates the complete pipeline under its canonical representation and Candidate Set, while the coarser NYC setting and Beauty directly evaluate set-valued compatible-event supervision.

The Mobility Word independently reaches 42.16% Acc@1 on NYC through grounding, with matched attribute-composition controls in Table 4. Stage 3 adds token-specific spatial credit and adaptive latent supervision. The coarser NYC control below retains the same Mobility Word.

D.2 Coarser-prefix NYC supervision control

The matched NYC control changes only the supervised prefix lengths at h_1, h_2, h_3 from $(2, 5, 6)$ to $(1, 2, 6)$, giving G, GH, and Exact targets. It retains the Mobility Word, candidate rule, Stage 1 initialization, 11,022 Train trajectories, teacher cache, detached entity keys, latent computation, gradient targets, and complete two-epoch recipe in Appendix C. Evaluation uses the same 1,447 targets and free-generation protocol; the predeclared epoch-2 endpoint is primary.

Under these Train candidate pools, 7,159/11,022 examples (64.952%) have a discriminative multi-POI G set, $1 < |S_G| < |\mathcal{C}|$; 7,445 (67.547%) have a strict G-to-GH reduction. All three targets are equal on 32.127% of examples, versus 97.859% under the original $(2, 5, 6)$ levels. Strict GH-to-Exact reduction occurs on 2.141%, so most multi-entity structure is at the first transition.

Table D.9: Matched NYC Stage 2 supervision at seed 42. All-equal rates use 11,022 Train examples; raw Acc@1 uses 1,447 NYC evaluation targets.

Supervised targets	All equal (%)	Exact correct	Acc@1 (%)
Full/Full/Full	100.000	626	43.26
Original $GH/GHFLB/\mathrm{Exact}$	97.859	634	43.81
Coarser $G/GH/\mathrm{Exact}$	32.127	632	43.68

Comparison across training seeds. We repeat the matched Full/Full/Full and $G/GH/\mathrm{Exact}$ treatments with Stage 2 seeds 42, 43, and 44, holding the Stage 1 initializer, representation, Train data, teacher cache, candidate construction, optimization recipe, and evaluation protocol fixed. Each result uses the prescribed epoch-2 endpoint. Table D.10 reports correct counts and mean Acc@1 with the sample standard deviation across training seeds.

Table D.10: Matched NYC Stage 2 results across three training seeds. Counts use the same 1,447 evaluation targets; Acc@1 is reported in percent as mean $\pm$ sample standard deviation across seeds.

Supervised targets	Seed 42	Seed 43	Seed 44	Mean correct	Acc@1 (%)
Full/Full/Full	626	636	624	628.67	43.446 ± 0.444
$G/GH/\mathrm{Exact}$	632	637	629	632.67	43.723 ± 0.279

Coarser-prefix ACS improves over Full/Full/Full in each evaluated seed, by 6, 1, and 5 correct predictions, respectively. The mean gain is four correct predictions, or 0.276 percentage points in Acc@1. These modest improvements support prefix-compatible groups as useful intermediate targets under the matched recipe, with the same entity representation and exact terminal supervision.