

Schema-Hierarchical Soft Prompts for Structured Information Extraction

Soumyadeep Dey* Srikanth Tenneti Bamdev Mishra
Shobhit Niranjana Mayur Datar

Microsoft India

{soumyadeep.dey, stenneti, bamdevm, shniranjana, mayurdatar}@microsoft.com

Abstract

Structured information extraction at small-language-model scale is a routine production workload, yet each new task added to a serving stack can demand new task-specific model state. We propose a schema-hierarchical soft prompt: a small learned prefix, trained on a frozen backbone, that gives every JSON-schema node its own slot, orders the slots by the schema, and marks tree depth with learned embeddings. Used alone on the open USPTO-ORD patent task, it recovers most of a LoRA reference’s extraction quality while training more than an order of magnitude fewer parameters, and ablations attribute the gain to depth and schema order rather than token count. Used with a LoRA adapter on an internal booking task, the same slots stabilize one of two held-out categories, so we report this as a bounded mechanism study rather than a general gain. Inference-time slot replacement indicates that position supplies the schema scaffold while learned content stabilizes generation. Measured serving gives a far smaller task artifact and higher throughput than multi-LoRA as interleaved tasks grow. The soft-prompt module is available at <https://github.com/microsoft/Hierarchical-Soft-Prompt>.

1 Introduction

Extracting structured JSON from unstructured text is a routine production workload at small-language-model (SLM) scale. Booking emails become booking records. Patent procedures become reaction graphs. Organizations often serve these workloads with 3-to-8B-parameter models that run close to the data. The practical question is simple: when an SLM is already in production, how much model-specific machinery is needed to add the next structured-extraction task?

The usual choices each have a cost. Prompt engineering ships a schema description in the prompt.

On small models, this can plateau below the accuracy needed for operational extraction, and the prompt grows with the schema. Full fine-tuning creates a new task-specific model copy. LoRA (Hu et al., 2022) makes the task artifact much smaller, but a multi-tenant serving stack must still either keep task-specific model copies resident or load and apply adapters at request time. Soft-prompt tuning (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2022) is attractive because it leaves the backbone frozen, but standard soft prompts are usually flat prefixes with no explicit relationship to the output schema.

We propose a soft prompt whose topology mirrors the target JSON schema, making three structural properties explicit in the trainable parameters. Each schema-node key gets its own learnable slot initialized from the embedding of the node’s key name. The slots follow a depth-first traversal (DFS) of the schema and each carries a learned depth embedding. A small set of shared global tokens is prepended to the per-key slots. The base model weights remain frozen throughout. A joint configuration also trains a small LoRA adapter on attention projections.

We evaluate this architecture on two structured-extraction tasks (USPTO-ORD patent reactions and an internal booking-email corpus) and three backbones in the main results (Qwen3-4B-Instruct, Qwen3-8B base, Phi-3.5-mini-Instruct). The schema-hierarchical topology serves two operating points on a frozen backbone, with a single mechanism connecting them.

On the patent task, soft-prompt-only adaptation is strong enough to be useful on its own, and the schema-shaped prefix remains a much smaller task artifact than the LoRA reference (§5). At serving time, adding a new task can be a prefix-embedding write rather than a model or adapter reload.

On the booking task, where the test distribution shifts in both booking category and personally iden-

* Corresponding author.

tifiable information (PII) surface forms, pairing the same positional slots with LoRA makes the transport-held-out, PII-shifted condition more stable than LoRA alone (§6). The deployed artifact in this regime is the LoRA adapter plus the positional prefix rather than the prefix alone.

Slot-drift diagnostics and an inference-time intervention that replaces held-out transport slots while preserving their prompt positions (§7) point to one explanation common to both operating points: the per-key slot acts as a positional scaffold. Its location in the prompt carries the main benefit, while the trained content at that position stabilizes the joint model across seeds.

Contributions. We first give a parameterization that places stable JSON-schema topology directly in the trainable prompt. Controlled ablations then show that depth and DFS order drive the open-task gain, rather than token count or semantic initialization alone. A paired positive and negative leave-one-category-out study characterizes when positional transfer holds. Finally, we report a measured multi-tenant serving profile against a modern multi-LoRA runtime. Appendix sections contain additional details.

2 Related Work

Schema-aware and hierarchical prompting for information extraction. Prior schema-aware extraction methods usually expose the schema to the model as text rather than as parameter structure. UIE (Lu et al., 2022) and InstructUIE (Wang et al., 2023) inject the target schema as hard text under full or instruction fine-tuning. PromptNER (Ashok and Lipton, 2023) and SSGPF (Yuan et al., 2025) similarly rely on hard schema or entity-type prompts, with SSGPF combining them with LoRA. TKDP (Liu et al., 2024) moves label knowledge into deep soft prompts, but it targets few-shot NER over flat entity-type label sets. Hierarchical prompting has also been explored outside fixed JSON schemas: TreePrompt (Zhang et al., 2023) composes soft prompts bottom-up from a parsed dependency tree for visual grounding, and Tree of Attributes Prompt (Ding et al., 2025) aligns hierarchical prompt tokens to LLM-generated concept trees. These methods either encode the schema as hard input text, assume a flat label set, or derive the hierarchy from the query rather than from a stable output schema. We instead make the schema a structural property of the *soft-prompt parameters*

themselves. On a frozen backbone, a nested JSON schema becomes K per-key slots (K = number of schema nodes) in DFS pre-order, with depth embeddings indexed by each node’s tree depth.

Soft prompt tuning and LoRA. Soft-prompt methods reduce task-specific state, but most treat the prompt as a flat sequence. Lester et al. (Lester et al., 2021) introduced flat-token prompt tuning, Prefix-Tuning (Li and Liang, 2021) prepends learnable activations at every transformer layer, and P-Tuning v2 (Liu et al., 2022) shows that deep layer-wise prompts can match full fine-tuning. SK-Tuning (Prottasha et al., 2024) initializes soft prompts from semantically meaningful word embeddings, and we adopt this principle at the per-key slot level. LoRA (Hu et al., 2022) provides our reference adapter baseline. The closest joint-adapter precedent is LoPA (Jain et al., 2024), which combines soft prompts with low-rank decomposition at the prompt level on GLUE NLU and code understanding tasks. We differ in schema-aware topology, the structured-extraction task family, and the multi-seed cross-backbone evaluation against a LoRA reference.

Decoding-time structured-output methods (orthogonal). A separate line of work (Zheng et al., 2024; Park et al., 2024; Schmidt and Cimiano, 2024; Hemmer et al., 2023; Lu et al., 2025; Wang et al., 2025) enforces format compliance at decode time. These methods improve format compliance, not value-extraction quality, and are orthogonal: a trained soft prompt can be stacked with a grammar-constrained decoder.

3 Method: Schema-Hierarchical Soft Prompt

The name *schema-hierarchical* reflects three structural properties of the target JSON schema that this prompt makes explicit in its trainable embeddings: the identity of each schema-node key, the order of nodes in the schema, and the depth of each node in the tree. The prompt has two parts. The first is k_g shared global tokens that carry task-identity information. The second is a sequence of K per-key slots, each holding t learnable tokens that carry schema content. Identity is preserved by allocating one slot per schema node and initializing it from the embedding of the node’s key name, the last component of its dotted path. Order is preserved by laying out the per-key slots in a depth-first traversal

Algorithm 1 Schema-hierarchical soft-prompt construction

Require: JSON schema S , embedding table E , k_g , tokens/slot t

- 1: $V \leftarrow \text{DFS_PREORDER}(S)$
- 2: **for all** $v \in V$ **do**
- 3: $q_v \leftarrow$ final component of the dotted path of v
- 4: $\mathbf{z}_v \leftarrow E[\text{TOKENIZE}((q_v))]$
- 5: $\mathbf{p}_v \leftarrow \text{REPEAT}(\text{MEAN}(\mathbf{z}_v), t)$
- 6: $\mathbf{p}_v \leftarrow \mathbf{p}_v + \mathbf{e}_{\text{depth}(v)}$
- 7: **end for**
- 8: $\mathbf{P} \leftarrow [\mathbf{P}_{\text{global}}; \mathbf{p}_{v_1}; \dots; \mathbf{p}_{v_K}]$
- 9: prepend $\mathbf{P}$, or insert each $\mathbf{p}_v$ beside its schema key
- 10: optimize $\mathbf{P}$ (and optional LoRA), keeping all backbone weights frozen

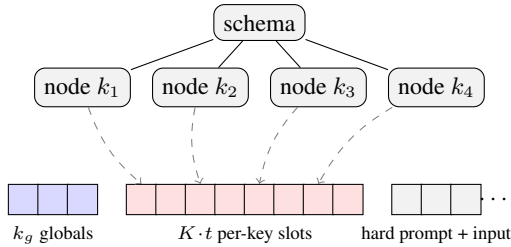


Figure 1: Schema-hierarchical soft-prompt architecture. The schema tree (top) defines K schema-node keys ($K=4$ shown, while real schemas have $K=94$ on Patent, of which 52 are leaves, and $K=62$ on Booking, of which 59 are leaves). The soft prompt (bottom) places k_g shared global tokens (blue) followed by $K \cdot t$ per-key tokens (red, $t=2$) in DFS pre-order over the schema, then the hard prompt and input text. Each per-key slot is initialized from the embedding of its key’s dotted-path name, and the backbone is frozen.

of the schema, so that sibling nodes of the same parent are adjacent in the prompt and parents precede children in token position. Depth is preserved by adding a small learned embedding, indexed by the node’s tree depth, to each per-key slot. The slots are then prepended to the input embeddings together with the globals. Training updates the prompt parameters while the base model weights remain frozen (Figure 1). When this prompt is paired with LoRA in the joint configurations of §4, the LoRA adapter is also trained.

Parameters. Let the target schema have K keys (all schema nodes in DFS pre-order), let L be the depth-embedding table size (one row per schema depth level, $L=5$ on Patent and $L=2$ on Booking),

and let D denote the backbone’s hidden dimension. Three tensors are trainable. The per-key slots $\mathbf{P}_{\text{key}} \in \mathbb{R}^{K \times t \times D}$ hold $t=2$ tokens per key. The global tokens $\mathbf{P}_{\text{glob}} \in \mathbb{R}^{k_g \times D}$ number $k_g=20$. The depth embeddings $\mathbf{E}_{\text{depth}} \in \mathbb{R}^{L \times D}$ are added to each per-key slot according to the node’s tree depth. The totals are ~ 545 K on Patent ($K=94$ schema nodes, 52 leaves, prefix length 208) and ~ 374 K on Booking ($K=62$ schema nodes, 59 leaves, prefix length 144).

Placement. We evaluate two placements. *Prepend* places all per-key slots together at position 0, followed by the hard prompt (the textual schema instructions) and the input. *Positional* inserts each per-key slot next to its key name in the hard prompt. Prepend is the default when the soft prompt is used alone. Positional is the placement we pair with LoRA.

Initialization. A per-key slot starts at the mean embedding of its key-name tokens, where the key name is the last component of the node’s dotted path. For example, `flightBooking.departureDate` starts at the mean of the `departure` and `Date` embeddings. Global tokens start at the mean embedding of the hard-prompt tokens, and depth embeddings are drawn from $\mathcal{N}(0, 0.02^2)$.

4 Experimental Setup

Patent. patent reaction extraction (USPTO-ORD) is a subset of the open USPTO-ORD corpus (Kearnes et al., 2021). Each input is a free-text procedure paragraph from a US patent, and the target is a canonicalized reaction JSON with 94 schema nodes, 52 leaves, and maximum depth 5. We evaluate on a held-out validation slice of $n=500$ using `field_em`, a leaf-path exact-match score on canonicalized gold outputs.

Booking. booking-email extraction is a proprietary internal booking-email corpus. Each input is a booking email, and the target is a booking JSON with flight, hotel, and transport sub-schemas (62 schema nodes, 59 leaves, maximum depth 2). We evaluate on three splits. Synthetic-ID ($n=1,475$) uses synthetic identifiers and contains flight bookings only. PII-Swap ($n=25$) is PII-shifted and contains all three booking types. Donated-OOD ($n=29$) is real out-of-distribution (OOD) data with no transport bookings.

Booking metric. For Booking, we report `macro_mandatory_f1` on each split and a weighted-by- n full-test-set summary ($n_{\text{total}}=1529$). `macro_mandatory_f1` is the unweighted mean across booking types (flight, hotel, transport) of F1 over the leaf keys marked mandatory in the booking schema. The weighted summary averages split-level values in proportion to split size.

Backbones. The primary backbone is Qwen3-4B-Instruct-2507 (Qwen Team, 2025), and Qwen3-8B (base) and Phi-3.5-mini-Instruct serve as cross-backbone tests. The base weights of every backbone remain frozen across all configurations.

Configurations. The four method names below are used consistently throughout the paper. *SP-only* is the schema-hierarchical soft prompt alone, in prepend placement, with no LoRA. *LoRA* is LoRA on attention projections ($r=16$) with no soft prompt. *Pos-SP + LoRA* pairs the soft prompt in positional placement with LoRA. *Prepend-SP + LoRA* pairs the soft prompt in prepend placement with LoRA. SP-only trains the prompt parameters, LoRA trains the LoRA adapter, and the two joint configurations train both. Headline configurations use seeds {42, 7, 123}, and ablations use seed 42 alone. All evaluations use `max_new_tokens=1536`.

Baselines and scope. Zero-shot uses the same textual schema prompt as the trained methods. Flat-SP uses the same number of trainable prompt tokens without per-key topology. These controls separate textual prompting, trainable-token count, schema structure, and LoRA capacity. Prior schema-aware extractors in §2 generally assume full or instruction fine-tuning, so they do not satisfy the shared frozen-backbone constraint.

5 Open-Benchmark Parameter Efficiency and Component Isolation

SP-only retains most of LoRA’s accuracy on Patent. On Qwen3-4B-Instruct, SP-only reaches 0.7654 `field_em` against the LoRA reference’s 0.8225, while training ~ 545 K parameters instead of ~ 15 M, a $\sim 28\times$ reduction (Table 1). The ranking is the same on all three backbones: zero-shot trails both trained methods, SP-only closes much of the gap, and LoRA stays ahead. A frozen-backbone prefix therefore recovers enough extraction quality to be a credible parameter-efficient adapter on this task.

Table 1: Patent `field_em` ($n=500$ validation slice) across three backbones. The SP-only column reports the three-seed mean over seeds {42, 7, 123}. The LoRA column reports the seed-42 reference. Multi-seed LoRA is reported on Booking in Tables 6 and 3.

Backbone	Zero-shot	SP-only	LoRA
Qwen3-4B-Instruct	0.1498	0.7654	0.8225
Qwen3-8B (base)	0.1953	0.7803	0.8308
Phi-3.5-mini	0.1853	0.7758	0.8395

Table 2: Patent matched controls and single-component ablations (Qwen3-4B, seed 42, SP-only, with the textual schema prompt removed). Delta is against the full schema-hierarchical prompt.

Configuration	<code>field_em</code>	Δ
Full hierarchy	0.7700	–
Flat-SP (equal tokens)	0.7553	–0.015
– depth embeddings	0.7409	–0.029
– DFS order (shuffled)	0.7539	–0.016
– semantic init (random)	0.7650	–0.005
– global tokens ($k_g=0$)	0.7826	+0.013

Schema structure, not token count, drives the open-task gain. Table 2 pairs the equal-token flat-SP control with single-component removals from the seed-42 Qwen3-4B SP-only configuration. The largest drops come from removing depth embeddings (–0.029) and shuffling DFS order (–0.016). Random initialization has a much smaller effect (–0.005), while removing global tokens improves this single run (+0.013). Thus the robust signal is schema topology (depth and order), not simply additional tokens near the input. The initialization and global-token deltas are small and remain single-seed evidence. Appendix D extends this control to Booking and the hard-prompt sweep.

On the Booking task, the best trained method is backbone-dependent. Training improves over zero-shot on every backbone, but the best trained method is not universal. Pos-SP + LoRA is strongest on Qwen3-4B-Instruct, while LoRA alone is strongest among the measured cells on the other two backbones. We therefore treat the Pos-SP + LoRA effect as a Qwen3-4B-Instruct result, not as a cross-backbone law. Appendix H further shows that the soft prompt and LoRA favor opposite backbones, so we read this pairing as backbone-specific rather than universally additive. The full weighted full-test-set table is in Appendix C.

Joint placement matters on Qwen3-4B-Instruct. Prepend-placed soft prompt with LoRA (Prepend-

SP + LoRA) behaves much like LoRA alone, and the soft prompt receives little gradient under joint training in this placement. Positional placement (Pos-SP + LoRA) is the configuration that changes the result: the slots help when they sit next to the schema fields they represent.

6 Leave-One-Category-Out Generalization Under PII Shift

This section asks whether the positional slots still help when an entire booking category, one of the three booking types of §4, is missing from training. The full-data booking runs train on flight, hotel, and transport, and categories can co-occur within an email. We now remove each minority category in turn and evaluate on PII-Swap ($n=25$), which contains all three categories and substitutes PII against a held-out vocabulary. Flight occurs in 99.1% of the 7,011 training emails. Holding it out would leave only about 1% of the corpus and conflate category generalization with a severe low-data regime. Hotel (9.9%) and transport (9.5%) form the symmetric leave-one-minority-category-out sweep. We train SP-only, LoRA, and Pos-SP + LoRA at seeds {42, 7, 123}. Full per-seed results are in Appendix E.

Positive and negative controls. When transport is held out, Pos-SP + LoRA improves overall F1 by +0.104 and reduces overall seed standard deviation from 0.077 to 0.011. When hotel is held out, the sign reverses (−0.061) and LoRA remains stronger. The only positive bootstrap interval that excludes zero is transport-held-out PII-Swap overall. Per-category intervals on this small split include zero. Bootstrap resampling uses the three paired seeds ($B=10,000$), while the paired t test has only $df=2$ and is treated as underpowered. We therefore present the booking result as a bounded mechanism study, not evidence of universal superiority.

Prompt-only capability control. A single deterministic prompt-only run using the matched textual schema prompt of §4 scores 0.843 on flight, 0.808 on hotel, 0.832 on transport, and 0.828 overall, with all values coming from one shared prediction file. That overall score is higher than either trained held-out checkpoint, but it is not a ranking row. This run uses vLLM, whereas the trained held-out checkpoints use the Hugging Face generation path, so the two are not measured on the same stack. We

report it only to establish that the backbone can already extract both held-out categories from text instructions. The question here is how much of that capability survives adaptation when a category is absent from training, not whether adaptation beats prompting.

7 Mechanism: Positional Scaffolding

The sweep in §6 supplies a positive control (transport) and a negative control (hotel) at nearly identical training frequencies. The remaining difference is structural congruence with the dominant flight category. Transport is another journey schema whose locations, passenger, and reservation fields reuse flight’s learned value spaces. Hotel instead introduces stay-specific rooms, addresses, and guest fields. Appendix F, Table 10, quantifies this with the fraction of each held-out field’s value-token vocabulary observed in flight training values. Transport’s improving pickup and dropoff fields have overlap 0.65–0.66, whereas hotel’s degrading name, room, and address fields are near 0.31–0.32.

Slot stability. Figure 2 compares each slot with its deterministic key-name initialization, for Pos-SP + LoRA and SP-only in both full-data and transport-held-out variants at three seeds each. The axes report drift L2, the Euclidean distance between a trained slot and that initialization. The joint model stays close to the full-data drift profile, whereas SP-only moves further under category holdout. Decoding each slot back to its nearest vocabulary tokens, by embedding similarity and by the `lm_head` projection, returns the same ranked tokens either way and shows that even the slots that moved farthest still decode to their own key name (Appendix G.2). These diagnostics are descriptive rather than proofs of internal semantics.

Position carries the mean effect and content stabilizes it. At inference, we replace every trained transport-subtree slot with either its initialization or a norm-matched random vector while preserving the slot positions. In Pos-SP + LoRA, random content retains 98% of mean overall F1 (0.737 against 0.752), but seed standard deviation grows from 0.011 to 0.057 (Appendix G.4, Table 11). This separates two roles: position supplies most of the reusable schema scaffold, while learned content stabilizes the result.

Boundary of the mechanism. Value overlap is predictive on the discriminative fields, but it is not

Table 3: Leave-one-minority-category-out control on Booking PII-Swap ($n=25$, Qwen3-4B-Instruct). “Held out” means the category was excluded from training. Scores are three-seed means of macro_mandatory_f1, the unweighted mean across the three booking types. Delta and 95% CI compare Pos-SP + LoRA with LoRA.

Held out	LoRA	Pos-SP + LoRA	Δ (95% CI)
transport	0.648 ± 0.077	0.752 ± 0.011	$+0.104$ $[+0.011, +0.163]$
hotel	0.483 ± 0.080	0.422 ± 0.081	-0.061 $[-0.208, +0.047]$

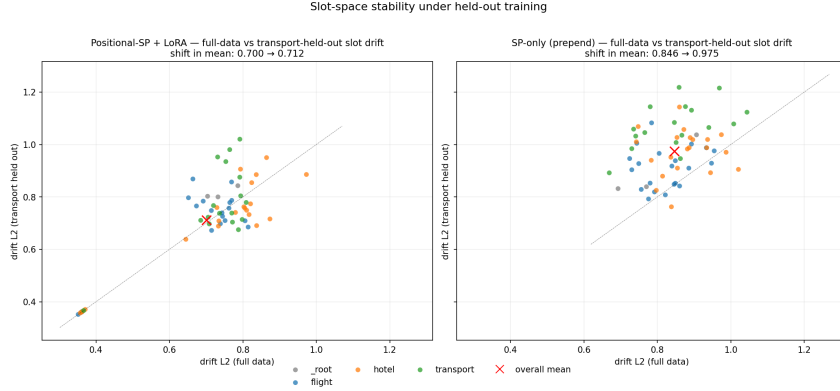


Figure 2: Per-slot drift L2 against initialization under full-data training (x) vs. transport-held-out training (y). One point is shown for each schema-node slot, colored by booking subtree with the root in gray. Pos-SP + LoRA stays close to the diagonal after transport-held-out training, whereas SP-only shifts upward, especially for transport slots. Additional details are in Appendix G.

sufficient. Date fields have high overlap without transferring, and key-name embedding coverage is similar across hotel and transport. Our supported claim is therefore operational rather than causal. Positional slots can preserve an unseen category without retraining when its field semantics and values align with a category abundant in training. Categories with novel field semantics still require adaptation. Appendix J probes whether the same positional effect appears in the inverse JSON-input direction.

8 Measured Multi-Tenant Serving

Protocol. We measure serving cost directly rather than projecting it from parameter counts. Two microbenchmarks run on the same model, Qwen3-4B-Instruct-2507 on one A100. The vLLM benchmark uses 64 fixed-length requests (512 input and 128 generated tokens), greedy decoding, and concurrency 32. Because the server API does not accept arbitrary input embeddings, we represent SP-only as 208 additional prompt positions. This is compute-equivalent for prefill and KV-cache cost, but it does not benchmark a custom prefix-injection API. LoRA uses vLLM’s unmerged multi-LoRA path. A separate Hugging Face+PEFT benchmark measures serialized artifacts, peak allocated mem-

Table 4: Measured single-task serving on one frozen Qwen3-4B-Instruct (vLLM, $1 \times A100$). Artifact sizes are measured separately with Hugging Face+PEFT.

Method	req/s	p50 ms	MB/task
Base	25.16	1252	—
SP-only	24.60	1284	2.183
LoRA	19.08	1660	47.225
SP + LoRA	18.74	1691	49.4

ory, prefix transfer, and warm adapter selection.

Single-task cost. SP-only is within 2% of base throughput and stores a 2.183 MB task artifact, versus 47.225 MB for LoRA. At 32 resident tasks, these measured artifacts total about 70 MB for SP versus 1.51 GB for LoRA before allocator and runtime overhead, leaving correspondingly more memory available to the KV cache. We do not separately benchmark that capacity effect. In the Hugging Face+PEFT harness, transferring a prefix takes 0.113 ms and warm adapter selection takes 8.219 ms. Peak allocated memory for one task is base-dominated (SP-only 9.46 GB and LoRA 9.11 GB), so we do not claim a single-task memory reduction. The SP cost is 208 extra positions and 94.7 ms additional time-to-first-token in that harness.

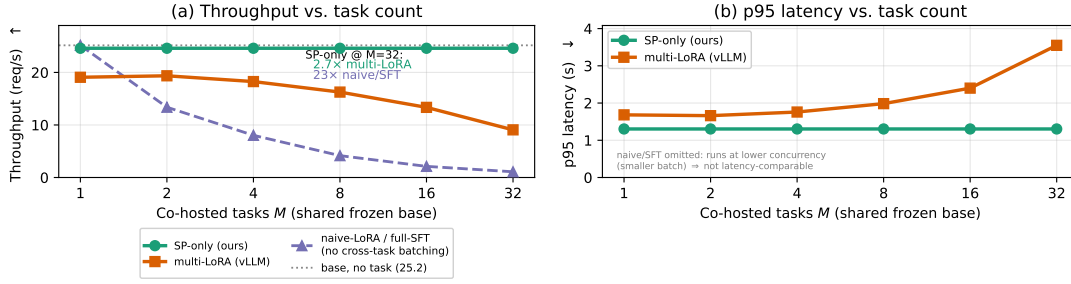


Figure 3: Multi-tenant serving versus task diversity M (vLLM, one A100, concurrency 32). Multi-LoRA directly measures batches with M adapter IDs. SP-only repeats the measured 208-position prefix workload. The no-cross-task-batching proxy uses the measured base concurrency sweep at effective concurrency $\lfloor 32/M \rfloor$.

Task-count scaling. Let M be the number of distinct tasks interleaved on one endpoint. Figure 3 measures the regime that motivates this work, namely many interleaved, individually low-traffic tasks. At $M=32$, SP-only still sustains 24.60 requests/s, unchanged from its single-task rate in Table 4 because task identity changes only the prompt embeddings, not the weights or kernels. Multi-LoRA falls to 9.07 requests/s, so SP-only sustains $2.7\times$ its throughput, and the measured no-cross-task-batching proxy falls to 1.09 ($23\times$). SP-only p95 latency is 1.30 s versus 3.55 s for multi-LoRA. The advantage is not that LoRA cannot batch. Modern multi-LoRA can batch, but it needs adapter-aware kernels whose cost grows with adapter diversity. SP instead mixes task prefixes in an ordinary shared-weight forward pass.

9 Conclusion

Schema-hierarchical soft prompts encode stable JSON topology in a compact frozen-backbone adapter, and controlled ablations identify depth and DFS order as the main contributors. The booking study bounds that claim, because positional slots preserve a held-out category only in the structurally aligned transport condition, not hotel. Measured serving identifies the regime where this lower-capacity adapter is useful, namely many interleaved tasks for which SP-level accuracy is sufficient. Categories with novel field semantics and accuracy-critical tasks should still use LoRA or fuller adaptation.

Limitations

Our evidence covers two extraction tasks and a few SLM backbones, at three seeds for the minority-category sweep and one for the component ab-

lations, so the joint booking gain is specific to Qwen3-4B-Instruct rather than universal. The proprietary booking corpus cannot be released, which limits independent replication, so the open USPTO-ORD result carries the primary empirical claim. Our diagnostics bound a mechanism rather than prove a causal account, and the positional design assumes a stable schema. Serving uses fixed-length synthetic requests rather than production traffic, with a compute-equivalent prefix-position proxy for SP in vLLM rather than a deployed fleet. We also do not compare directly against schema-aware systems, whose full or instruction fine-tuning violates our shared frozen-backbone constraint. Appendix I reports a negative result for a training-time attention-routing regularizer that we do not recommend for deployment.

Ethics Statement

Patent uses the open USPTO-ORD corpus (Kearnes et al., 2021) (no personal data). Booking is a proprietary internal booking-email corpus whose PII is removed or substituted before training and evaluation, and the corpus itself is not released. Donor consent, the PII protocol, misuse considerations, and compute and carbon details are in Appendix A.

References

- Dhananjay Ashok and Zachary C. Lipton. 2023. PromptNER: Prompting for named entity recognition.
- Tong Ding, Wanhua Li, Zhongqi Miao, and Hanspeter Pfister. 2025. Tree of attributes prompt learning for vision-language models. In *Proc. ICLR*.
- Arthur Hemmer, Mickaël Coustaty, Nicola Bartolo, Jérôme Brachat, and Jean-Marc Ogier. 2023. Lazy-

- k: Decoding for constrained token classification. In *Proc. EMNLP*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*.
- Abhinav Jain, Swarat Chaudhuri, Thomas Reps, and Chris Jermaine. 2024. Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. In *Proc. NeurIPS*.
- Steven M. Kearnes, Michael R. Maser, Michael Wlekliński, Anton Kast, Abigail G. Doyle, Spencer D. Dreher, Joel M. Hawkins, Klavs F. Jensen, and Connor W. Coley. 2021. The open reaction database. *Journal of the American Chemical Society*, 143(45):18820–18826.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proc. EMNLP*.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. ACL-IJCNLP*.
- Jiang Liu, Hao Fei, Fei Li, Jingye Li, Bobo Li, Liang Zhao, Chong Teng, and Donghong Ji. 2024. TKDP: Threefold knowledge-enriched deep prompt tuning for few-shot named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6397–6409.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proc. ACL (Volume 2: Short Papers)*.
- Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proc. ACL*.
- Yaxi Lu, Haolun Li, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Zhiyuan Liu, Fangming Liu, and Maosong Sun. 2025. Learning to generate structured output with schema reinforcement learning. In *Proc. ACL (Long Papers)*.
- Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D’Antoni. 2024. Grammar-aligned decoding. In *Proc. NeurIPS*.
- Nusrat Jahan Prottasha, Asif Mahmud, Md. Shohanur Islam Sobuj, Prakash Bhat, Md Kowsher, Niloufar Yousefi, and Ozlem Ozmen Garibay. 2024. Parameter-efficient fine-tuning of large language models using semantic knowledge tuning. *Scientific Reports*.
- Qwen Team. 2025. Qwen3 technical report.
- David M. Schmidt and Philipp Cimiano. 2024. Grammar-constrained decoding for structured information extraction with fine-tuned generative models applied to clinical trial abstracts. *Frontiers in Artificial Intelligence*, 7.
- Darren Yow-Bang Wang, Zhengyuan Shen, Soumya Smruti Mishra, Zhichao Xu, Yifei Teng, and Haibo Ding. 2025. SLOT: Structuring the output of large language models. In *Proc. EMNLP Industry Track*.
- Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. InstructUIE: Multi-task instruction tuning for unified information extraction.
- Xiang Yuan, Xinrong Chen, Haochen Li, Hang Yang, Guanyu Wang, Weiping Li, and Tong Mo. 2025. Stepwise schema-guided prompting framework with parameter efficient instruction tuning for multimedia event extraction. In *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*.
- Chenchi Zhang, Jun Xiao, Lei Chen, Jian Shao, and Long Chen. 2023. TreePrompt: Learning to compose tree prompts for explainable visual grounding.
- Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. SGLang: Efficient execution of structured language model programs. In *Proc. NeurIPS*.

A Ethics: Details

Internal corpus and PII handling. Booking (booking-email extraction) is a proprietary internal booking-email corpus from a single operating organization. The original corpus contains personally identifiable information (passenger names, contact details, booking and reservation identifiers, addresses). PII is removed or replaced systematically before training and evaluation. PII-Swap (PII-Swap) is constructed by deterministic substitution of PII tokens against a held-out PII vocabulary. Synthetic-ID (Synthetic-ID) uses synthetic identifiers throughout. Donated-OOD (Donated-OOD) is post-processed to remove residual PII. We cannot release the corpus, because it is proprietary to the operating organization, and this constraint applies even in the PII-removed form.

Donated emails (Donated-OOD). The Donated-OOD split consists of booking emails voluntarily donated by individuals for research use. Donor consent and the PII-removal protocol were validated by the operating organization’s data-governance

review before the emails entered the training and evaluation pipeline.

Potential misuse. The trained soft-prompt artifact is a small parameter set ($\sim 374\text{--}545$ K) on a frozen backbone. It does not give an attacker generative capability beyond the underlying backbone. The same architecture could in principle be used to train extractors over sensitive structured text. This is a generic risk of all parameter-efficient fine-tuning and is not specific to schema-hierarchical soft prompts.

Compute and carbon. We do not report a carbon estimate because the cluster power draw and grid-carbon logs needed for a defensible calculation were not available for all runs. Hardware and training schedules are in Appendix B.

B Reproducibility

This appendix consolidates the hyperparameters, training schedule, software environment, and compute footprint needed to reproduce the runs in the paper.

Hyperparameters. Hyperparameters are identical across Patent and Booking, with only the prefix length differing because the schemas differ. Table 5 groups them by method (SP-only, LoRA, and the joint configurations Prepend-SP + LoRA / Pos-SP + LoRA).

We do not use linear LR warmup. Instead the schedule is a three-phase curriculum (LoRA-warmup, then SP-only, then joint), each phase running its own cosine decay. SP-only and LoRA skip the inapplicable phases. The SP-only phase length of 4500 steps corresponds to roughly 10 epochs on the internal training set ($n_{\text{train}}=7011$, effective batch 16, ≈ 438 steps/epoch). SP-only needs more epochs than LoRA because only $\sim 370\text{K}$ parameters train.

Compute. Training runs use a single NVIDIA A100 80GB GPU. Wall-clock time depends on backbone, sequence length, and cluster load, so we report the fixed training schedules in Table 5 rather than a single aggregate GPU-hour estimate. Patent ($n_{\text{train}}=80,000$) takes proportionally more wall time per epoch than Booking ($n_{\text{train}}=7011$), but the schedules use the same step counts.

Software environment. We use Python 3.10+, PyTorch $\geq 2.1.0$ with the cu121 wheels, transformers $\geq 4.46.0$, peft 0.18.1, accelerate $\geq 0.33.0$,

and datasets $\geq 2.20.0$. The server CUDA driver is 12.2. The planned release pins exact versions via requirements.txt and a conda environment file.

Seeds. Headline three-seed runs use seeds {42, 7, 123}. Single-seed ablations use seed 42. The seed controls the dataloader shuffling, the per-slot init noise ($0.01 \cdot \mathcal{N}(0, 1)$ on per-key slots, $\mathcal{N}(0, 0.02^2)$ on depth embeddings), and the dropout RNG.

Decoding. All evaluations use greedy decoding (no temperature, no top- k , no top- p) with max_new_tokens=1536. The train_v1 system prompt for Booking is ~ 770 tokens (fields plus descriptions) and will be included verbatim in the release package.

Data.

- Patent (patent reaction extraction (USPTO-ORD)): a deterministic 80,000/10,000/10,000 train/valid/test split of the open USPTO-ORD-100K chemistry-procedures corpus, published with (Kearnes et al., 2021) and distributed as a prebuilt archive at https://github.com/qai222/LLM_organic_synthesis under CC-BY-SA 4.0. We use the canonicalized reaction-JSON variant. The canonicalization script and the $n=500$ validation slice will be included in the release package.
- Booking (booking-email extraction): a proprietary internal booking-email corpus from a single operating organization. Training set $n_{\text{train}}=7011$, validation $n_{\text{val}}=321$. Evaluation splits are described in §4. The corpus is not released. See the Ethics Statement for provenance and PII handling.

Public artifacts. The soft-prompt module is available at <https://github.com/microsoft/Hierarchical-Soft-Prompt>.

C Booking Weighted Full-Test-Set Table

Table 6 reports the weighted full-test-set result summarized in §5.

D Per-Key vs. Flat Architecture Ablation

Main-text Table 2 reports the equal-token control for Patent. Across the hard-prompt sweep, SP-only (schema-hierarchical) improves monotonically as the hard prompt shrinks ($0.7381 \rightarrow 0.7661 \rightarrow 0.7700$), while the flat variant does not show the

Table 5: Training hyperparameters by method. “Joint” covers Prepend-SP + LoRA and Pos-SP + LoRA, which use the same schedule.

	SP-only	LoRA	Joint
Optimizer	AdamW	AdamW	AdamW
LR (soft prompt)	10^{-3}	—	10^{-3}
LR (LoRA)	—	10^{-4}	10^{-4}
Weight decay (LoRA)	—	0.01	0.01
Weight decay (SP)	0	—	0
Gradient clip	1.0	1.0	1.0
LR schedule	cosine	cosine	cosine
$\eta_{\min}$	10^{-6}	10^{-6}	10^{-6}
Per-device batch size	4	4	4
Gradient accumulation	4	4	4
Effective batch size	16	16	16
LoRA-warmup phase	—	1000	500
SP-only phase	4500	—	500
Joint phase	—	2000	2000
Total gradient steps	4500	3000	3000
Eval interval (steps)	500	500	500
LoRA r	—	16	16
LoRA α	—	32	32
LoRA dropout	—	0.05	0.05
LoRA targets	—	q,k,v,o-proj	q,k,v,o-proj
SP k_g	20	—	20
SP t (tokens/slot)	2	—	2
SP prefix length	208/144	—	208/144
Trainable params (T1/T2)	~ 545 K/ ~ 374 K	~ 15 M	~ 15 M + SP

Table 6: Booking weighted-by- n full-test-set macro_mandatory_f1 ($n=1529$ across Synthetic-ID/PII-Swap/Donated-OOD). Split scores are weighted by split size. Zero-shot uses the matched train_v1 system prompt.

Backbone	Zero-shot	SP-only	LoRA	Pos-SP + LoRA
Qwen3-4B-Inst	0.6406	0.7741	0.6411	0.7841
Qwen3-8B (base)	0.2702	0.7664	0.7929	0.7727
Phi-3.5-mini	0.3811	0.6547	0.7483	0.7453

same monotonic trend. On Booking at three seeds, schema-hierarchical-vs-flat differences are within seed noise (+0.006 on Synthetic-ID, +0.011 on PII-Swap, −0.038 on Donated-OOD). Per-key structure helps conditionally, either on the deeper open schema when hard text is minimized, or when paired with LoRA and placed positionally (§6).

E Multi-Seed Leave-One-Category-Out Results

This appendix expands main-text Table 3: three seeds $\times$ three methods on Qwen3-4B-Instruct (train_v1, max_new_tokens=1536), with transport and hotel separately excluded from training.

Per-seed PII-Swap F1. Table 7 reports per-seed PII-Swap F1 per booking type for each method. Pos-SP + LoRA reaches the highest mean overall (0.752 ± 0.011), with seed variance more than $7 \times$

tighter than LoRA (0.648 ± 0.077). On the held-out transport type itself, Pos-SP + LoRA sits at 0.655 ± 0.013 against LoRA’s 0.528 ± 0.108 , with both higher mean and tighter variance.

Hotel held out. Table 8 is the matched negative control. Pos-SP + LoRA does not exceed LoRA on the held-out hotel type or overall.

Control splits. On Synthetic-ID (flight-only, $n=1,475$), all three methods sit within ~ 0.02 across seeds (means: LoRA 0.786, Pos-SP + LoRA 0.779, SP-only 0.768). On Donated-OOD ($n=29$, no transport bookings, no PII shift), the three-seed means are LoRA 0.880, Pos-SP + LoRA 0.818, SP-only 0.642. LoRA is ahead because the held-out-category pressure is absent on this split.

Bootstrap CIs and paired tests. Table 9 reports transport-held-out bootstrap 95% CIs ($B=10,000$

Table 7: Booking PII-Swap ($n=25$) per-seed F1 under transport-held-out training. Columns are per-booking-type and overall. Bottom row of each method shows three-seed mean $\pm$ std.

Method	seed	flight	hotel	transport	overall
LoRA	42	0.867	0.731	0.670	0.756
	7	0.868	0.458	0.408	0.578
	123	0.814	0.514	0.507	0.612
	mean $\pm$ std	0.850 ± 0.025	0.568 ± 0.117	0.528 ± 0.108	0.648 ± 0.077
SP-only	42	0.904	0.306	0.293	0.501
	7	0.854	0.620	0.625	0.700
	123	0.885	0.347	0.383	0.538
	mean $\pm$ std	0.881 ± 0.020	0.424 ± 0.140	0.434 ± 0.140	0.580 ± 0.086
Pos-SP + LoRA	42	0.921	0.729	0.651	0.767
	7	0.868	0.715	0.641	0.741
	123	0.901	0.673	0.673	0.749
	mean $\pm$ std	0.896 ± 0.022	0.706 ± 0.024	0.655 ± 0.013	0.752 ± 0.011

Table 8: Booking PII-Swap ($n=25$) per-seed macro_mandatory_f1 under hotel-held-out training (Qwen3-4B-Instruct). Bottom rows report three-seed mean $\pm$ std.

Method	seed	flight	hotel	transport	overall
LoRA	42	0.877	0.125	0.143	0.382
	7	0.894	0.400	0.442	0.579
	123	0.878	0.292	0.300	0.490
	mean $\pm$ std	0.883 ± 0.008	0.272 ± 0.113	0.295 ± 0.122	0.483 ± 0.080
SP-only	42	0.799	0.303	0.331	0.477
	7	0.854	0.278	0.293	0.475
	123	0.931	0.125	0.120	0.392
	mean $\pm$ std	0.861 ± 0.054	0.235 ± 0.079	0.248 ± 0.092	0.448 ± 0.040
Pos-SP + LoRA	42	0.933	0.069	0.075	0.359
	7	0.895	0.111	0.107	0.371
	123	0.873	0.347	0.392	0.537
	mean $\pm$ std	0.900 ± 0.025	0.176 ± 0.122	0.191 ± 0.142	0.422 ± 0.081

resamples over the three seeds, with replacement) and paired- t deltas. The paired test has only $df=2$ and is severely underpowered, so the bootstrap CI is primary. Hotel-held-out PII-Swap gives $\Delta_{\text{hotel}} = -0.096$ with CI $[-0.289, +0.056]$ and $\Delta_{\text{overall}} = -0.061$ with CI $[-0.208, +0.047]$. On Donated-OOD, the held-out hotel delta is significantly negative (-0.161 , CI $[-0.227, -0.048]$). These controls bound the positive result to transport-held-out PII-Swap overall.

F Transfer Analysis for the Held-Out Categories

Value-overlap definition. For each held-out field, we form its normalized training value-token vocabulary and report the fraction also observed among flight-category training values. This descriptive statistic asks whether an extractor trained predominantly on flight has seen the same kinds of values. Table 10 pairs it with held-out-category per-field F1.

Boundary conditions. As §7 notes, value overlap alone does not explain the sign change. The evidence supports a whole-schema congruence account, in which value and field semantics jointly gate transfer, rather than a single scalar law.

G Slot Interpretability: Full Diagnostics

This appendix supplements §7 with the per-slot diagnostic protocol and the full slot-intervention results.

G.1 Per-slot drift analysis

For each per-key slot i , we compute $\|\text{trained}_i - \text{init}_i\|_2$ and $\cos(\text{trained}_i, \text{init}_i)$, where trained_i is the mean across the $t=2$ tokens of the slot with depth embedding added, and init_i is the deterministic mean of the key-name token embeddings. Mean drift L2 across the $K=62$ per-key slots is 0.700 for Pos-SP + LoRA on full data and 0.712 under transport holdout (+1.7%), against 0.846 and 0.975 (+15%) for SP-only. Figure 4 shows the per-

Table 9: Bootstrap 95% CIs and paired deltas on PII-Swap ($n=25$), using $B=10,000$ resamples over the three seeds. The paired- t test has $df=2$ and is underpowered, so the bootstrap CI is the primary inferential statistic.

Comparison	type	mean Δ	95% CI	t	p
Pos-SP + LoRA – LoRA	flight	+0.046	$[-0.001, +0.086]$	+1.83	0.208
Pos-SP + LoRA – LoRA	hotel	+0.138	$[-0.002, +0.257]$	+1.83	0.209
Pos-SP + LoRA – LoRA	transport	+0.127	$[-0.019, +0.233]$	+1.68	0.234
Pos-SP + LoRA – LoRA	overall	+0.104	$[+0.011, +0.163]$	+2.21	0.158
Pos-SP + LoRA – SP-only	flight	+0.015	$[+0.014, +0.017]$	+14.72	0.005
Pos-SP + LoRA – SP-only	hotel	+0.281	$[+0.094, +0.423]$	+2.88	0.102
Pos-SP + LoRA – SP-only	transport	+0.221	$[+0.016, +0.357]$	+2.12	0.168
Pos-SP + LoRA – SP-only	overall	+0.173	$[+0.041, +0.266]$	+2.55	0.125

Table 10: Per-field transfer on PII-Swap (three-seed mean) versus value-token overlap with flight training values. Transport’s improving fields reuse flight value spaces, whereas hotel’s distinctive fields do not.

Held out	field	LoRA	Pos-SP + LoRA	Δ	overlap
transport	passenger_name	0.595	0.752	+0.157	1.00
	reservationId	0.360	0.480	+0.120	0.44
	pickup_location	0.630	0.732	+0.102	0.65
	dropoff_location	0.630	0.732	+0.102	0.66
hotel	hotel_name	0.347	0.208	−0.139	0.31
	room_type	0.302	0.153	−0.149	0.32
	address	0.320	0.162	−0.159	0.32

slot drift profile across all six checkpoints. Figure 6 pairs the dendrogram of Figure 5 with a heatmap view of the same pairwise cosine matrix.

G.2 Token-cloud cross-readout

For each selected schema key we compute three vector variants of its slot. The *key-only* variant is the initialization reference, *SP-only* is the trained slot with its depth embedding, and *combined* is the mean of the key and slot vectors. Each variant is passed through two top-K readouts, embedding nearest-neighbors and the raw lm_head logit. We prefer raw logits over softmax probabilities, which compress to $\sim 10^{-5}$ – 10^{-3} over a 150K vocabulary and round illegibly. The two readouts return the same ranked tokens for every (slot, variant, cell) combination. Figures 7 and 8 show the readout for the full-data Pos-SP + LoRA checkpoint. Figures 9 and 10 show the same readout for the transport-held-out SP-only checkpoint, the configuration with the largest slot drift.

G.3 Global tokens: task-identity vs. schema-content

For each architecture we compute the cosine similarity of each global token to every per-key slot’s mean. The per-global mean cosine to the per-key slot space is near zero in every architecture (range -0.002 to $+0.004$), and the per-global max co-

sine to any single per-key slot sits in 0.05 to 0.14 (Figure 11). The globals occupy a subspace approximately orthogonal to the per-key schema-content space. This is consistent with a task-identity interpretation, in which the globals encode something other than schema content (a “you are doing structured JSON extraction” signal).

G.4 Slot intervention: full results

This subsection reports the three-seed slot intervention experiment summarized in §7. We additionally report a magnitude-rescaling probe on the full-data SP-only checkpoint (seed 42).

Protocol. For a given checkpoint and a target transport subtree, we modify the corresponding per-key slots on disk at the start of evaluation. The remaining slots (other subtrees and the global tokens) are untouched. Three modifications are evaluated:

- **init.** The target slot is replaced with the deterministic mean of the tokenized key-name embeddings, repeated across the $t=2$ token positions. This is the initialization used at training time, without the small Gaussian perturbation.
- **random.** The target slot is replaced with a random unit vector, drawn from a fixed seed and rescaled to match the norm of the init

Table 11: Inference-time slot intervention on the transport-held-out checkpoints, three-seed mean $\pm$ std over seeds {42, 7, 123}. PII-Swap macro_mandatory_f1 ($n=25$), Qwen3-4B-Instruct. The *trained* row is the unmodified checkpoint. The *init* and *random* rows replace transport-subtree slots at inference while leaving all other slots untouched.

Method	slot variant	flight	hotel	transport	overall
Pos-SP + LoRA	trained	0.896 ± 0.022	0.706 ± 0.024	0.655 ± 0.013	0.752 ± 0.011
	init	0.881 ± 0.026	0.710 ± 0.085	0.664 ± 0.041	0.752 ± 0.050
	random	0.873 ± 0.028	0.707 ± 0.100	0.632 ± 0.041	0.737 ± 0.057
SP-only	trained	0.881 ± 0.020	0.424 ± 0.140	0.434 ± 0.140	0.580 ± 0.086
	init	0.898 ± 0.019	0.551 ± 0.240	0.590 ± 0.225	0.680 ± 0.153
	random	0.906 ± 0.014	0.550 ± 0.281	0.557 ± 0.298	0.671 ± 0.190

vector. This decouples the position of the slot from its semantic content.

- **boost.** The target slot is multiplied by a fixed factor ($1.5\times$ or $2.0\times$). Applied to the full-data SP-only checkpoint, this tests whether amplifying the trained slot direction improves F1.

Full-data SP-only magnitude rescaling. On the full-data SP-only checkpoint, multiplying all per-key slots by $1.5\times$ at inference yields PII-Swap overall 0.858, above the three-seed-mean baseline of 0.8445. The $2.0\times$ scale gives 0.817, indicating that the gain saturates and reverses for larger factors. This probe checks the role of slot magnitude in the SP-only setting.

H Base-vs-Instruct Ablation

The headline analysis in §5 reports a Qwen3-4B-Instruct LoRA weighted full-test-set (FTS) of 0.6411, well below the LoRA reference on Qwen3-8B (0.7929) and Phi-3.5-mini-Instruct (0.7483). We probe this gap by training the same configurations on the non-instruction-tuned Qwen3-4B-Base model.

Setup. Five matched cells are trained on Qwen3-4B-Base (seed 42, train_v1 prompt, full Booking training set, max_new_tokens=1536) and compared cell-for-cell against their Qwen3-4B-Instruct twins: SP-only (prepend), SP-only (positional), LoRA alone, Prepend-SP + LoRA, and Pos-SP + LoRA. Here SP-only (positional) uses the positional placement without LoRA, while SP-only (prepend) is the default SP-only of §4. All training settings are identical except the backbone tuning state. Table 12 reports the comparison. Cross-family replication is not available because Phi-3.5-mini-Base is not published on the public hub and

Table 12: Booking weighted-by- n full-test-set macro_mandatory_f1 ($n=1529$), Qwen3-4B-Base vs. Qwen3-4B-Instruct, seed 42. Delta is base minus instruct.

Configuration	Base	Instruct	Δ
SP-only (prepend)	0.669	0.774	-0.105
SP-only (positional)	0.404	0.462	-0.058
LoRA alone	0.796	0.641	$+0.155$
Prepend-SP + LoRA	0.795	0.646	$+0.149$
Pos-SP + LoRA	0.782	0.786	-0.005

Qwen3-8B is already a base model (no instruct twin available).

Findings. Table 12 supports three conclusions.

- *LoRA-bearing methods on base reach or beat instruct.* LoRA alone ($+0.155$), Prepend-SP + LoRA ($+0.149$), and Pos-SP + LoRA ($\Delta = -0.005$) on Qwen3-4B-Base equal or exceed their instruct twins on weighted-FTS.
- *The Qwen3-4B-Instruct LoRA underperformance is instruct-interference-specific.* LoRA on Qwen3-4B-Base reaches 0.796, in line with the Qwen3-8B (0.7929) and Phi-3.5-mini (0.7483) LoRA references. The underperformance on Qwen3-4B-Instruct (0.6411) comes from the instruction-tuning of *this specific instruct release* interacting with LoRA fine-tuning on the new task, not from a Qwen-family or LoRA-method property.
- *SP-only on base underperforms instruct.* Without LoRA, the schema-hierarchical soft prompt cannot substitute for the instruction-tuning the base lacks. Both placements on Qwen3-4B-Base sit below their instruct twins (SP-only prepend $\Delta = -0.105$ and SP-only positional $\Delta = -0.058$).

Mechanistic reading. This is consistent with §7. LoRA absorbs the value-extraction and instruction-following work, and the schema-hierarchical positional slots act as positional scaffolds. On Qwen3-4B-Instruct, the instruction-tuning interferes with LoRA fine-tuning on the new task (the LoRA drop to 0.6411). The joint Pos-SP + LoRA configuration recovers because the slots supply the structural cue that LoRA on its own does not. On a base backbone, LoRA has the full capacity to do both jobs and the slots’ marginal contribution is small.

Scope. This ablation is at seed 42 only.

I Hard Attention Routing (Negative Result)

We evaluated a training-time per-key attention-routing regularizer that constrains each gold-value-span token to attend only to its own key’s slot. The mask is a 4-D additive tensor stacked with the standard causal mask during training. Evaluation and generation run with ordinary causal attention, so there is no inference-time overhead. Under three-seed paired-baseline analysis the regularizer produced no measurable effect on either task (Patent mean $\Delta = -0.009$, Booking OOD mean $\Delta = +0.077$ with $\sigma_{\Delta} = 0.18$, 95% CI [-0.070,+0.135]). The full method description, per-seed paired-delta tables, and geometric mechanism diagnostics are released with the code. We do not recommend the regularizer for deployment.

J Generality Probe: JSON-Input QA

The main paper studies text-to-JSON extraction. As a generality probe we additionally evaluate the inverse direction (JSON-input free-form QA) on two real-world JSON-native datasets, PubChem chemistry and SEC multi-company financial. Per-key positional placement beats a flat prepended baseline by +5.3 to +10.2 F1 at Qwen 0.5B (three seeds, both bf16 and 4-bit, both domains). The gap narrows to within seed noise at Qwen 1.5B as base capacity grows. On SmolLM2 (Llama architecture), per-key placement helps chemistry on 135M and 1.7B by +2.0 to +3.1 F1, but shows no reliable effect on financial. The probe is consistent with the positional-scaffold mechanism of §7 generalizing to the inverse direction, with effect size shrinking as base capacity grows.

Per-slot drift: L2 magnitude vs. directional alignment with init

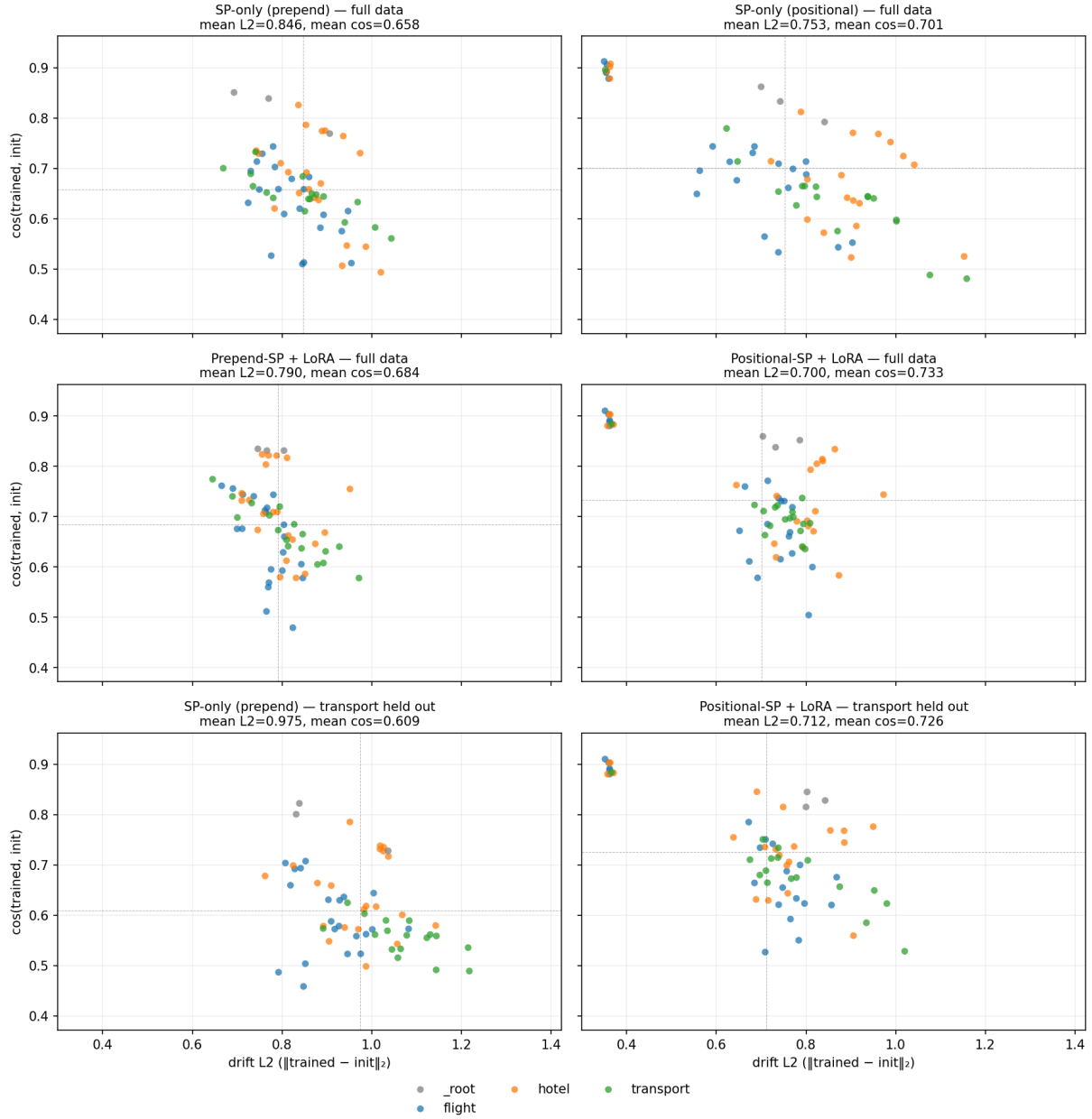


Figure 4: Per-slot drift profile across all six checkpoints. Points are per-key slots (one per schema node), colored by booking-type subtree. The x -axis is drift L2 from init, and the y -axis is cosine to init. Pos-SP + LoRA separates low-drift structural fields from higher-drift value fields, and this pattern persists under transport-held-out training. In SP-only under transport-held-out training, the low-drift cluster is weaker and transport slots (green) move farthest from initialization.

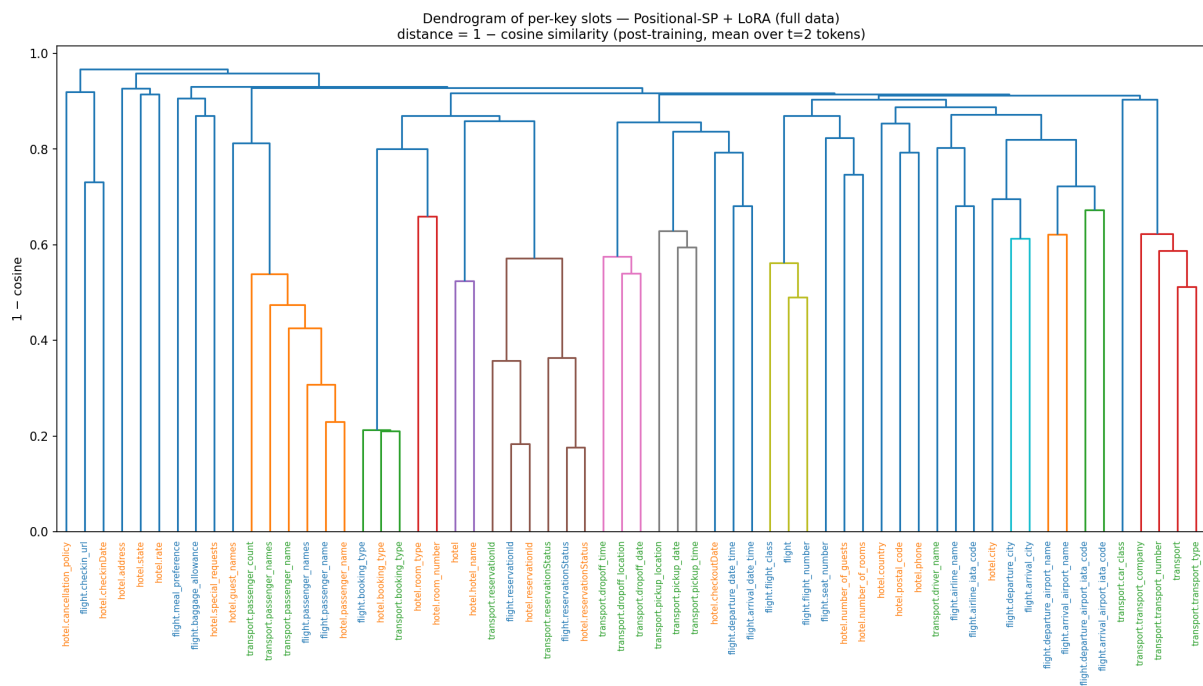


Figure 5: Hierarchical clustering of the 62 per-key slots in Pos-SP + LoRA (full data, seed 42), distance = 1 – cos. Dendrogram labels are colored by booking-type subtree. The tightest clusters mix booking types (same-field-name across subtrees), showing that the slot space recovers field-name identity rather than the booking-type partition.

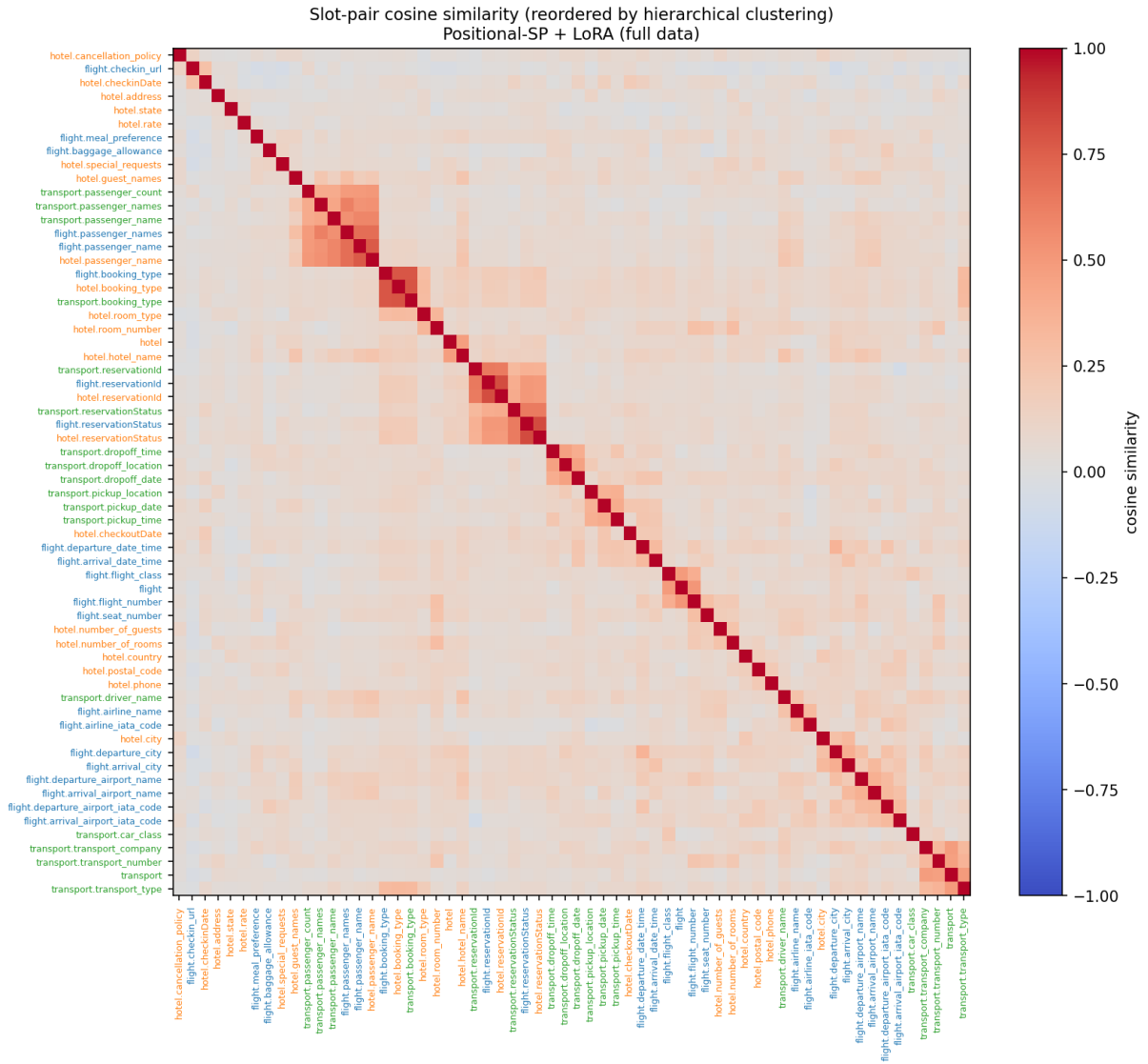


Figure 6: Pairwise cosine matrix of the 62 per-key slots in Pos-SP + LoRA, rows / columns reordered by the dendrogram of Figure 5. Red diagonal blocks hold same-field-name slots across booking subtrees.

Embedding NN — Positional-SP + LoRA — full data — scope=curated — green = preserved from key-only neighbours

key-only	SP-only (learned)	combined (key + SP)
'reservation' (+0.76) 'status' (+0.68) 'Status' (+0.59) 'status' (+0.58) 'status' (+0.57) 'Reservation' (+0.57) 'reservation' (+0.55) 'status' (+0.53)	'reservation' (+0.35) 'status' (+0.32) 'Status' (+0.27) 'status' (+0.27) 'reservation' (+0.27) 'status' (+0.27) 'Reservation' (+0.26) 'status' (+0.25)	'reservation' (+0.55) 'status' (+0.50) 'Status' (+0.43) 'status' (+0.42) 'status' (+0.42) 'Reservation' (+0.41) 'reservation' (+0.41) 'status' (+0.38)
'booking' (+0.77) 'type' (+0.66) 'Booking' (+0.58) 'booking' (+0.58) 'TYPE' (+0.55) 'booking' (+0.55) 'Booking' (+0.54) 'booking' (+0.52)	'booking' (+0.34) 'type' (+0.28) 'booking' (+0.25) 'Booking' (+0.25) 'Booking' (+0.23) 'booking' (+0.23) 'Type' (+0.23) 'Type' (+0.23)	'booking' (+0.54) 'type' (+0.46) 'Booking' (+0.40) 'booking' (+0.40) 'booking' (+0.38) 'Booking' (+0.37) 'TYPE' (+0.37) 'Type' (+0.36)
'country' (+1.00) 'country' (+0.82) 'Country' (+0.77) 'country' (+0.74) 'Country' (+0.70) 'country' (+0.70) 'country' (+0.69) 'country' (+0.64)	'country' (+0.53) 'country' (+0.43) 'Country' (+0.41) 'country' (+0.39) 'Country' (+0.38) 'country' (+0.37) 'country' (+0.37) 'country' (+0.36)	'country' (+0.68) 'country' (+0.55) 'Country' (+0.52) 'country' (+0.49) 'Country' (+0.48) 'country' (+0.48) 'country' (+0.47) 'country' (+0.45)
'address' (+1.00) 'address' (+0.81) 'address' (+0.77) 'Address' (+0.76) 'Address' (+0.76) 'address' (+0.75) 'address' (+0.73) 'address' (+0.70)	'address' (+0.59) 'address' (+0.52) 'Address' (+0.48) 'address' (+0.48) 'Address' (+0.48) 'address' (+0.44) 'address' (+0.44) 'addressing' (+0.43)	'address' (+0.72) 'address' (+0.63) 'address' (+0.58) 'address' (+0.58) 'Address' (+0.58) 'address' (+0.55) 'address' (+0.54) 'address' (+0.52)
'booking' (+0.77) 'type' (+0.66) 'Booking' (+0.58) 'booking' (+0.58) 'TYPE' (+0.55) 'booking' (+0.55) 'Booking' (+0.54) 'booking' (+0.52)	'booking' (+0.33) 'type' (+0.28) 'booking' (+0.24) 'Booking' (+0.24) 'TYPE' (+0.23) 'Type' (+0.23) 'booking' (+0.23) 'Type' (+0.22)	'booking' (+0.53) 'type' (+0.46) 'booking' (+0.39) 'Booking' (+0.39) 'TYPE' (+0.38) 'booking' (+0.37) 'Booking' (+0.37) 'Type' (+0.36)
'reservation' (+0.76) 'status' (+0.68) 'Status' (+0.59) 'status' (+0.58) 'status' (+0.57) 'Reservation' (+0.57) 'reservation' (+0.55) 'status' (+0.53)	'reservation' (+0.31) 'status' (+0.26) 'reservation' (+0.24) 'Reservation' (+0.23) 'status' (+0.22) 'Status' (+0.22) 'status' (+0.21) 'status' (+0.21)	'reservation' (+0.51) 'status' (+0.44) 'Reservation' (+0.38) 'Status' (+0.38) 'reservation' (+0.38) 'status' (+0.37) 'status' (+0.37) 'status' (+0.34)
'transport' (+0.77) 'company' (+0.70) 'transport' (+0.63) 'transport' (+0.60) 'Transport' (+0.59) 'transport' (+0.57) 'transportation' (+0.54) 'transportation' (+0.54)	'transport' (+0.31) 'transport' (+0.27) 'company' (+0.26) 'transport' (+0.25) 'Transport' (+0.24) 'transportation' (+0.24) 'transportation' (+0.22) 'Transportation' (+0.22)	'transport' (+0.52) 'company' (+0.45) 'transport' (+0.43) 'transport' (+0.41) 'Transport' (+0.40) 'transportation' (+0.38) 'transportation' (+0.37) 'transporting' (+0.36)
'off' (+0.63) 'drop' (+0.62) 'location' (+0.54) 'drop' (+0.47) 'Drop' (+0.46) 'Off' (+0.45) 'Drop' (+0.45) 'Location' (+0.45)	'off' (+0.25) 'drop' (+0.24) 'drop' (+0.20) 'location' (+0.18) 'Drop' (+0.17) 'off' (+0.17) 'OFF' (+0.17) 'dropped' (+0.17)	'off' (+0.45) 'drop' (+0.44) 'location' (+0.36) 'drop' (+0.35) 'Drop' (+0.32) 'Drop' (+0.31) 'Off' (+0.31) '_off' (+0.31)

Figure 7: Embedding-NN readout for a curated 8-key set, Pos-SP + LoRA full-data. Rows are selected keys. Columns show key-only initialization, trained slot, and combined key-plus-slot vectors. Green marks tokens preserved from key-only initialization.

lm_head logit readout — Positional-SP + LoRA — full data — scope=curated — green = preserved from key-only neighbours

key-only	SP-only (learned)	combined (key + SP)
'reservation' (+0.67) 'status' (+0.54) 'Status' (+0.49) 'reservation' (+0.48) 'Reservation' (+0.47) 'status' (+0.46) '-status' (+0.45) 'Reservation' (+0.44)	'reservation' (+0.66) 'status' (+0.53) 'reservation' (+0.50) 'Status' (+0.47) 'status' (+0.46) 'Reservation' (+0.46) 'Reservation' (+0.45) 'reservations' (+0.44)	'reservation' (+0.67) 'status' (+0.53) 'reservation' (+0.49) 'Status' (+0.48) 'Reservation' (+0.46) 'status' (+0.46) 'Reservation' (+0.45) 'status' (+0.43)
'booking' (+0.56) 'booking' (+0.42) type (+0.41) 'booking' (+0.41) 'Booking' (+0.40) 'booking' (+0.38) 'booking' (+0.36) 'bookings' (+0.35)	'booking' (+0.57) 'booking' (+0.42) type (+0.41) 'Booking' (+0.40) 'Booking' (+0.40) 'Type' (+0.38) 'Type' (+0.37) '_booking' (+0.37)	'booking' (+0.57) 'booking' (+0.42) type (+0.41) 'Booking' (+0.41) 'Booking' (+0.40) 'booking' (+0.38) - 'Type' (+0.36) 'Type' (+0.36)
'country' (+1.08) 'country' (+0.86) 'Country' (+0.81) 'Country' (+0.77) '-country' (+0.75) 'country' (+0.75) 'country' (+0.72) 'country' (+0.72)	'country' (+1.16) 'country' (+0.92) '-country' (+0.87) 'Country' (+0.87) 'Country' (+0.83) 'country' (+0.79) 'country' (+0.77) 'country' (+0.77)	'country' (+1.13) 'country' (+0.90) 'Country' (+0.85) '-country' (+0.83) 'Country' (+0.81) 'country' (+0.77) 'country' (+0.76) 'country' (+0.75)
'address' (+1.11) 'address' (+0.87) 'Address' (+0.82) 'Address' (+0.81) [] (+0.81) '-address' (+0.78) 'ADDRESS' (+0.78) '_address' (+0.78)	'address' (+1.23) 'address' (+1.06) 'Address' (+0.98) 'Address' (+0.97) [] (+0.93) 'address' (+0.91) '-address' (+0.89) 'addresses' (+0.87)	'address' (+1.19) 'address' (+0.99) 'Address' (+0.93) 'Address' (+0.92) [] (+0.89) 'address' (+0.86) '-address' (+0.86) 'addresses' (+0.83)
'booking' (+0.56) 'booking' (+0.42) type (+0.41) 'booking' (+0.41) 'Booking' (+0.40) 'booking' (+0.38) 'booking' (+0.36) 'bookings' (+0.35)	'booking' (+0.55) 'booking' (+0.41) type (+0.41) 'Booking' (+0.38) 'Booking' (+0.38) 'Type' (+0.37) 'Type' (+0.36) 'booked' (+0.36)	'booking' (+0.56) 'booking' (+0.42) type (+0.41) 'Booking' (+0.40) 'Booking' (+0.39) 'booking' (+0.37) - 'Type' (+0.36) 'Type' (+0.36)
'reservation' (+0.67) 'status' (+0.54) 'Status' (+0.49) 'reservation' (+0.48) 'Reservation' (+0.47) 'status' (+0.46) '-status' (+0.45) 'Reservation' (+0.44)	'reservation' (+0.66) 'reservation' (+0.51) 'Status' (+0.50) 'Reservation' (+0.47) 'Reservation' (+0.45) 'Status' (+0.44) 'reservations' (+0.44) 'status' (+0.44)	'reservation' (+0.67) 'Status' (+0.52) 'reservation' (+0.50) 'Reservation' (+0.47) 'Status' (+0.46) 'status' (+0.45) 'Reservation' (+0.44) 'status' (+0.43)
'transport' (+0.67) 'company' (+0.55) 'transport' (+0.53) 'Transport' (+0.51) 'transport' (+0.51) 'transports' (+0.50) 'transport' (+0.49) - [] (+0.46)	'transport' (+0.65) 'transport' (+0.53) 'transport' (+0.50) 'Transport' (+0.49) 'company' (+0.48) 'transportation' (+0.47) 'transports' (+0.46) [] (+0.45)	'transport' (+0.66) 'transport' (+0.53) 'company' (+0.51) 'transport' (+0.51) 'Transport' (+0.50) 'transports' (+0.48) 'transportation' (+0.46) [] (+0.45)
'off' (+0.48) 'drop' (+0.48) 'location' (+0.35) - 'drop' (+0.35) 'offs' (+0.35) '-drop' (+0.34) 'Drop' (+0.34) 'OFF' (+0.33)	'off' (+0.56) 'drop' (+0.53) 'drop' (+0.42) 'offs' (+0.41) '-drop' (+0.38) 'OFF' (+0.37) 'Drop' (+0.37) 'drops' (+0.36)	'off' (+0.51) 'drop' (+0.50) 'drop' (+0.38) 'offs' (+0.37) '-drop' (+0.36) 'OFF' (+0.35) 'Drop' (+0.35) 'Drop' (+0.34)

Figure 8: lm_head logit readout for the same 8 keys, same layout as Figure 7. Both readouts return the same ranked tokens. For value-rich slots, trained-slot logits are *higher* than key-only logits, consistent with training rotating and amplifying the slot within the key-name region.

Embedding NN — SP-only (prepend) — transport held out — scope=curated — green = preserved from key-only neighbours

key-only	SP-only (learned)	combined (key + SP)
'reservation' (+0.76) 'status' (+0.68) 'status' (+0.59) 'status' (+0.58) 'status' (+0.57) 'Reservation' (+0.57) 'reservation' (+0.55) 'status' (+0.53)	'reservation' (+0.31) 'status' (+0.28) 'status' (+0.24) 'status' (+0.24) 'status' (+0.24) 'status' (+0.23) 'status' (+0.23) 'reservation' (+0.23)	'reservation' (+0.51) 'status' (+0.45) 'status' (+0.39) 'status' (+0.39) 'status' (+0.38) 'Reservation' (+0.37) 'status' (+0.37) 'reservation' (+0.37)
'booking' (+0.77) 'type' (+0.66) 'Booking' (+0.58) 'booking' (+0.58) 'TYPE' (+0.55) 'booking' (+0.55) 'Booking' (+0.54) 'booking' (+0.52)	'booking' (+0.31) 'type' (+0.25) 'Booking' (+0.24) 'booking' (+0.23) 'booking' (+0.23) 'Booking' (+0.22) '_booking' (+0.21) '_TYPE' (+0.20)	'booking' (+0.50) 'type' (+0.42) 'Booking' (+0.39) 'booking' (+0.38) 'booking' (+0.37) 'Booking' (+0.35) '_TYPE' (+0.35) '_booking' (+0.34)
'country' (+1.00) 'country' (+0.82) 'Country' (+0.77) 'country' (+0.74) 'Country' (+0.70) 'country' (+0.70) 'country' (+0.69) '-country' (+0.64)	'country' (+0.51) 'country' (+0.42) 'Country' (+0.40) 'country' (+0.37) 'country' (+0.36) 'country' (+0.35) 'Country' (+0.35) '-country' (+0.34)	'country' (+0.66) 'country' (+0.54) 'Country' (+0.51) 'country' (+0.48) 'country' (+0.47) 'country' (+0.46) 'Country' (+0.46) '-country' (+0.43)
'address' (+1.00) 'address' (+0.81) 'address' (+0.77) 'Address' (+0.76) 'Address' (+0.76) 'address' (+0.75) 'address' (+0.73) '□□' (+0.70)	'address' (+0.55) 'address' (+0.45) 'Address' (+0.41) 'address' (+0.41) 'address' (+0.41) 'Address' (+0.41) 'address' (+0.41) '(address' (+0.38)	'address' (+0.69) 'address' (+0.56) 'Address' (+0.52) 'address' (+0.52) 'Address' (+0.52) 'address' (+0.52) 'address' (+0.51) '(address' (+0.48)
'booking' (+0.77) 'type' (+0.66) 'Booking' (+0.58) 'booking' (+0.58) 'TYPE' (+0.55) 'booking' (+0.55) 'Booking' (+0.54) 'booking' (+0.52)	'booking' (+0.29) 'booking' (+0.22) 'Booking' (+0.22) 'booking' (+0.20) '_booking' (+0.20) 'type' (+0.20) 'Booking' (+0.19) '(type' (+0.17)	'booking' (+0.48) 'type' (+0.37) 'Booking' (+0.36) 'booking' (+0.36) 'booking' (+0.34) 'Booking' (+0.33) '_TYPE' (+0.32) '_booking' (+0.31)
'reservation' (+0.76) 'status' (+0.68) 'status' (+0.59) 'status' (+0.58) 'status' (+0.57) 'Reservation' (+0.57) 'reservation' (+0.55) 'status' (+0.53)	'reservation' (+0.29) 'status' (+0.27) 'status' (+0.23) 'status' (+0.23) 'status' (+0.23) 'status' (+0.23) 'status' (+0.22) '_STATUS' (+0.22)	'reservation' (+0.48) 'status' (+0.44) 'status' (+0.38) 'status' (+0.37) 'status' (+0.37) 'Status' (+0.36) 'status' (+0.36) 'Reservation' (+0.36)
'transport' (+0.77) 'company' (+0.70) 'transport' (+0.63) 'transport' (+0.60) 'Transport' (+0.59) 'transport' (+0.57) 'transportation' (+0.54) '□□' (+0.54)	'transport' (+0.30) 'company' (+0.25) 'Transport' (+0.24) 'transport' (+0.23) 'transportation' (+0.23) '□' (+0.23) 'transport' (+0.23) '-transport' (+0.22)	'transport' (+0.49) 'company' (+0.43) 'Transport' (+0.40) 'transport' (+0.38) 'transport' (+0.37) 'Transport' (+0.37) 'transportation' (+0.37) '□□' (+0.36)
'off' (+0.63) 'drop' (+0.62) 'location' (+0.54) 'drop' (+0.47) 'Drop' (+0.46) 'Off' (+0.45) 'Drop' (+0.45) 'Location' (+0.45)	'drop' (+0.21) 'off' (+0.20) 'location' (+0.19) 'location' (+0.17) 'DROP' (+0.16) 'drop' (+0.16) 'dropout' (+0.16) 'location' (+0.16)	'drop' (+0.41) 'off' (+0.41) 'location' (+0.37) 'Drop' (+0.31) 'drop' (+0.31) 'location' (+0.30) 'Location' (+0.30) 'drop' (+0.30)

Figure 9: Embedding-NN readout, SP-only transport-held-out. Despite drift L2 over 1 on transport slots, the trained-slot column still recovers the key-name tokens, consistent with the slot being amplified within the transport-key region rather than anti-aligned.

lm_head logit readout — SP-only (prepend) — transport held out — scope=curated — green = preserved from key-only neighbours

key-only	SP-only (learned)	combined (key + SP)
'reservation' (+0.67) 'status' (+0.54) 'status' (+0.49) 'reservation' (+0.48) 'Reservation' (+0.47) 'status' (+0.46) '-status' (+0.45) 'Reservation' (+0.44)	'reservation' (+0.67) 'status' (+0.54) 'reservation' (+0.48) 'statuses' (+0.48) 'status' (+0.47) '-status' (+0.47) 'status' (+0.47) 'getStatus' (+0.45)	'reservation' (+0.67) 'status' (+0.54) 'reservation' (+0.48) 'status' (+0.48) 'status' (+0.46) 'Reservation' (+0.46) '-status' (+0.46) 'statuses' (+0.45)
'booking' (+0.56) 'booking' (+0.42) 'type' (+0.41) 'booking' (+0.41) 'Booking' (+0.40) 'booking' (+0.38) 'booking' (+0.36) 'bookings' (+0.35)	'booking' (+0.59) 'booking' (+0.44) 'booking' (+0.44) 'booking' (+0.42) 'booking' (+0.41) 'type' (+0.40) 'booking' (+0.37) '[]' (+0.36)	'booking' (+0.58) 'booking' (+0.43) 'Booking' (+0.43) 'Booking' (+0.41) 'type' (+0.40) 'booking' (+0.40) 'booking' (+0.37) '[]' (+0.34)
'country' (+1.08) 'country' (+0.86) 'Country' (+0.81) 'Country' (+0.77) '-country' (+0.75) 'country' (+0.75) 'country' (+0.72) 'country' (+0.72)	'country' (+1.12) 'country' (+0.90) 'Country' (+0.85) '-country' (+0.80) 'Country' (+0.78) 'country' (+0.78) 'country' (+0.75) 'country' (+0.73)	'country' (+1.11) 'country' (+0.89) 'Country' (+0.83) '-country' (+0.78) 'Country' (+0.78) 'country' (+0.76) 'country' (+0.74) '(country' (+0.74)
'address' (+1.11) 'address' (+0.87) 'Address' (+0.82) 'Address' (+0.81) '[]' (+0.81) '-address' (+0.78) 'ADDRESS' (+0.78) '-address' (+0.78)	'address' (+1.19) 'address' (+0.95) 'Address' (+0.89) 'Address' (+0.87) '-address' (+0.86) '-address' (+0.84) 'address' (+0.83) 'addresses' (+0.83)	'address' (+1.16) 'address' (+0.92) 'Address' (+0.87) 'Address' (+0.85) '-address' (+0.83) '-address' (+0.82) 'ADDRESS' (+0.81) 'address' (+0.81)
'booking' (+0.56) 'booking' (+0.42) 'type' (+0.41) 'booking' (+0.41) 'Booking' (+0.40) 'booking' (+0.38) 'booking' (+0.36) 'bookings' (+0.35)	'booking' (+0.59) 'booking' (+0.46) 'Booking' (+0.43) 'Booking' (+0.40) 'booking' (+0.40) 'booking' (+0.39) 'bookings' (+0.36) '_type' (+0.34)	'booking' (+0.58) 'booking' (+0.44) 'Booking' (+0.42) 'Booking' (+0.40) 'booking' (+0.39) 'type' (+0.38) 'booking' (+0.37) 'bookings' (+0.35)
'reservation' (+0.67) 'status' (+0.54) 'status' (+0.49) 'reservation' (+0.48) 'Reservation' (+0.47) 'status' (+0.46) '-status' (+0.45) 'Reservation' (+0.44)	'reservation' (+0.68) 'status' (+0.55) 'statuses' (+0.53) 'status' (+0.49) 'reservation' (+0.49) 'status' (+0.48) 'status' (+0.48) 'getStatus' (+0.48)	'reservation' (+0.68) 'status' (+0.55) 'reservation' (+0.49) 'status' (+0.49) 'statuses' (+0.48) 'status' (+0.48) 'Reservation' (+0.47) '-status' (+0.46)
'transport' (+0.67) 'company' (+0.55) 'transport' (+0.53) 'transport' (+0.51) 'transport' (+0.51) 'transports' (+0.50) 'transport' (+0.49) '[]' (+0.46)	'transport' (+0.69) 'transport' (+0.53) 'transport' (+0.52) 'transport' (+0.52) 'company' (+0.51) 'transport' (+0.51) '[]' (+0.50) 'transport' (+0.50)	'transport' (+0.68) 'company' (+0.53) 'transport' (+0.53) 'transport' (+0.51) 'transport' (+0.51) 'transports' (+0.50) 'transports' (+0.49) 'transport' (+0.48)
'off' (+0.48) 'drop' (+0.48) '-location' (+0.35) 'drop' (+0.35) 'offs' (+0.35) '-drop' (+0.34) 'Drop' (+0.34) 'OFF' (+0.33)	'drop' (+0.47) 'off' (+0.46) 'DROP' (+0.36) 'OFF' (+0.36) 'location' (+0.35) 'getLocation' (+0.35) 'Drop' (+0.35) 'dropout' (+0.34)	'off' (+0.48) 'drop' (+0.47) '-location' (+0.35) 'OFF' (+0.34) 'Drop' (+0.34) 'drop' (+0.34) '-drop' (+0.34) 'DROP' (+0.33)

Figure 10: lm_head logit readout, SP-only transport-held-out. Same layout as Figure 8. Trained-slot logits for key-name tokens are comparable to or larger than the corresponding Pos-SP + LoRA logits in Figure 8.

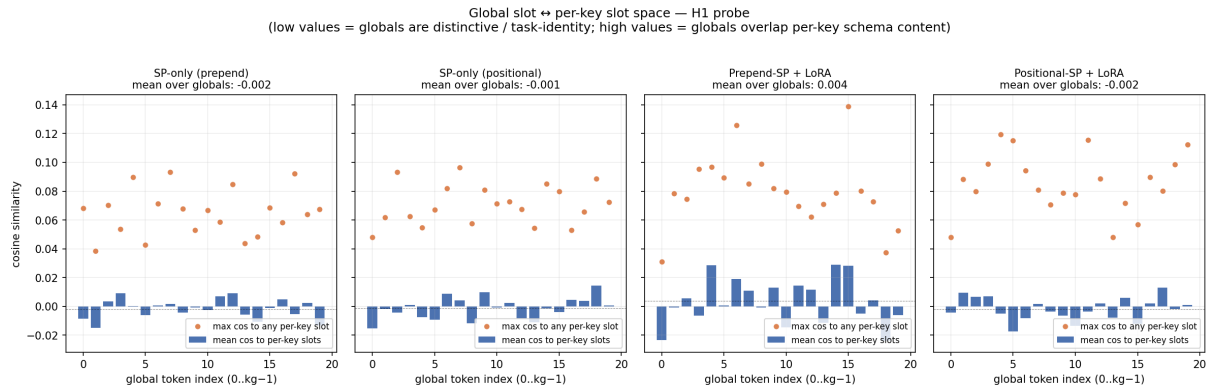


Figure 11: Cosine similarity between global tokens and per-key slots across the four full-data architectures. Bars show each global token’s mean cosine to the per-key slot space, and orange dots show its maximum cosine to any single per-key slot. Global tokens are approximately orthogonal to per-key schema content in every architecture.