FOCUSING SLOT: INFERENCE-TIME SLOT STEERING FOR ZERO-SHOT DIALOGUE STATE TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot dialogue state tracking (DST) aims to generalize to unseen domains without retraining or target-domain annotations. However, existing approaches that either append slot- or schema-level descriptions or add lightweight adapters, face two inference-time limitations that cause negative transfer: (i) domain bias, where shared slot names across domains bias the model toward the source domain; and (ii) slot-understanding bias, where target-only slots fail to adequately capture their relationships with and distinctions from existing slots, leading the model to mistakenly reuse source domain slot patterns. To address this, we present StDST, a training-free and plug-and-play activation steering method. Our approach constructs steering vectors for each slot using source-domain slot descriptions. Specifically, we combine a positive prototype and a negative prototype to form the vector for shared slots, and employ a global negative vector for target-only slots. During inference, we inject a small vector offset along these vectors to explicitly align slot semantics, which mitigates domain bias and slot-understanding bias, thereby reducing negative transfer. Through extensive experiments on the MultiWOZ and SGD datasets, StDST demonstrates the effectiveness across various domains, and the consistent gains achieved when pairing it with diverse architectures further attest to its utility. Our code is accessible at: `https://anonymous.4open.science/r/StDST-673F`.

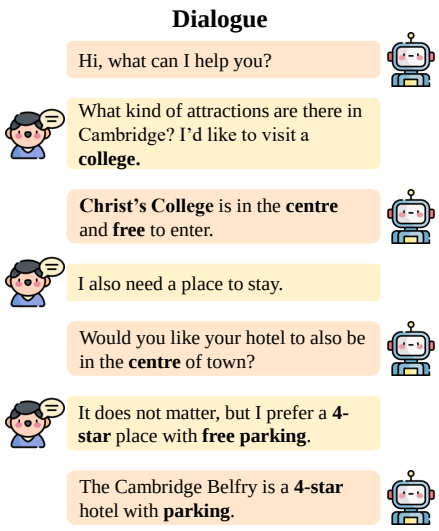
1 INTRODUCTION

Task-oriented dialogue systems (TODs) support goal-oriented tasks by maintaining a structured belief state that captures constraints over domain, slot, and value throughout the conversation. Dialogue state tracking (DST) is the core component that infers and updates this evolving state from user utterances and system actions, enabling responses and backend operations that remain aligned with user goals. Zero-shot DST aims to infer domain, slot, and value constraints in unseen domains without in-domain supervision by leveraging transferable semantics from schema and slot descriptions and knowledge in pretrained language models (PLMs). The goal is to maintain an accurate belief state under domain or schema shift, thereby reducing annotation and retraining costs and enabling rapid and scalable deployment to new domains.

In recent years, with the advancement of language models, mainstream approaches to zero-shot DST have primarily revolved around two categories of models. The first category comprises methods based on PLMs, which typically leverage slot descriptions to guide the model in generating values for target domains. Examples include T5-DST (Lin et al., 2021b), SDP-DST (Lee et al., 2021), and DualLoRA (Luo et al., 2024), which incorporate slot information through techniques such as appending descriptions, inserting prefixes, or employing dual low-rank adaptation paths. The second category consists of methods based on large language models (LLMs), such as ICL-DST (Hu et al., 2022) and LDST (Feng et al., 2023), which enhance zero-shot reasoning capabilities via instruction tuning or task reformulation. Despite the progress achieved by these approaches, they still face the issue of negative transfer (NT). This occurs when knowledge acquired from training on source domains harms performance on target domains, as shown in Figure 1. The issue arises because they erroneously rely on the pattern of the source domain, leading to biases when applied to new domains.

Specifically, we have observed two typical patterns of negative transfer: *domain bias* for shared slots and *slot-understanding bias* for target-only slots. *Domain bias* occurs when the source and target domains share the same slot name; in such cases, the model tends to output values common

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.



(a) **Domain Bias**

Shared Slot	Source Domain (Attraction)	Target Domain (Hotel)
Area	Centre	Centre ✗ Gold: don't care
Type	College	College ✗ Gold: None
Name	Christ's College	Christ's College ✗ Gold: Cambridge Belfry

(b) **Slot-Understanding Bias**

Target-only Slot	Dialogue Cue	Target Domain (Hotel)
Hotel-Stars	"4-star place"	Expensive ✗ Gold: 4
Hotel-Parking	"free parking"	Free ✗ Gold: Yes

Figure 1: An illustration of two types of biases in zero-shot dialogue state tracking. For shared slots, the model tends to copy source-domain values into the target domain, leading to domain bias. For target-only slots, the model often fails to capture the slot semantics and predicts source-familiar but invalid values, resulting in slot-understanding bias.

in the source domain. *Slot-understanding bias* arises for slots that are unique to the target domain. Due to a lack of semantic understanding of these unfamiliar slots, the model incorrectly replicates the filling patterns of similar slots from the source domain (e.g., applying the pattern for "price" to predict values for "stars"). Together, these two types of bias lead to a decline in model performance in new domains, thereby limiting zero-shot generalization capability. Following Dong et al. (2024), we quantify both errors using the negative transfer (NT) rate, which measures the proportion of incorrect predictions whose predicted values appear in the source domains. The formal definition and calculation details of the NT rate are provided in Appendix E. Figure 2 reports the NT rate for both error types, highlighting their significance.

To address the aforementioned issues, we propose StDST, a training-free, plug-and-play inference-time activation steering method. The core idea is to extract from the source domain a universal semantic representation of slots and then explicitly rectify the model's slot representations during inference via lightweight vector injection, thereby mitigating negative transfer. For *shared slots*, we construct a positive prototype to capture cross-domain commonalities of the slot, alongside a negative prototype to suppress confusion with other slots. The key lies in emphasizing the inherent semantics of the slot itself, rather than which domain it belongs to. For *target-only slots*, we construct a global negative vector to push their representations away from the semantic space of all source domain slots, thus avoiding erroneous pattern reuse. The entire process requires no target-domain annotations, no fine-tuning, and no modifications to the model architecture, achieving plug-and-play operation. The proposed method seamlessly integrates with models of different architectures, improves zero-shot performance, reduces negative transfer rates, and demonstrates strong generalization capability and robustness. Through extensive experiments on the MultiWOZ and SGD datasets, we validate the effectiveness of StDST. Our main contributions are as follows:

- To our knowledge, we are the first to apply steering to zero-shot dialogue state tracking, offering a training-free plug-and-play inference method that alleviates negative transfer.
- We introduce slot prototype steering vector and global negative steering vector on the source domains and apply them at inference to mitigate the negative transfer in the target domain. The approach is model-agnostic and training-free.
- Extensive experiments are conducted on the MultiWOZ and SGD datasets to demonstrate the effectiveness of the proposed method.

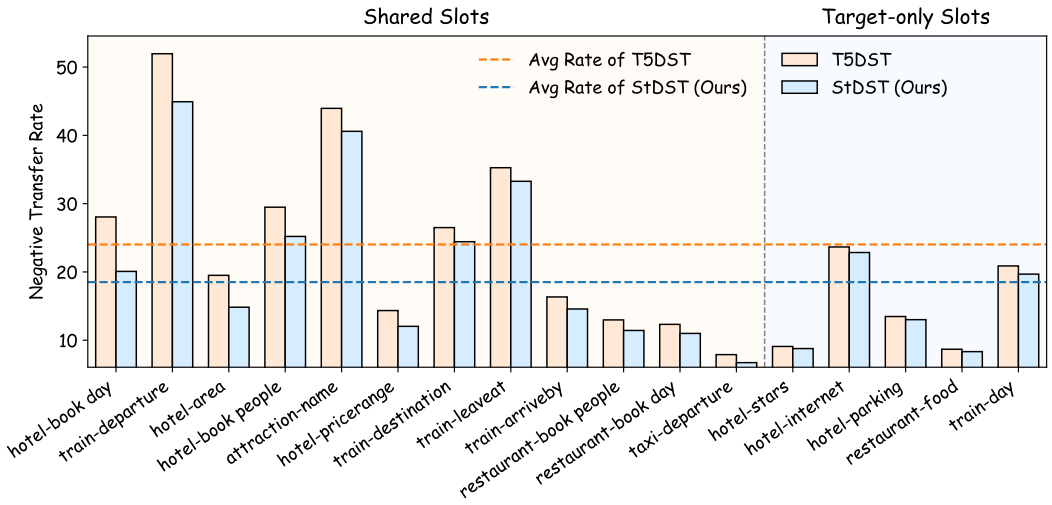


Figure 2: Negative Transfer Rate on the MultiWOZ dataset. The lower negative transfer rate is better.

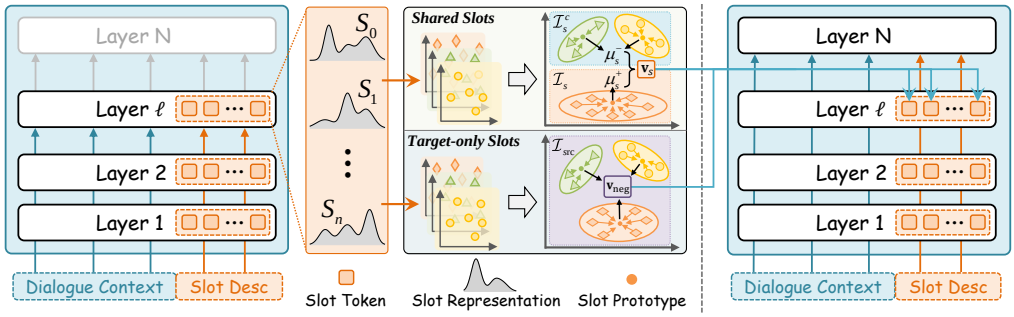
2 Related Work

2.1 Dialogue State Tracking

In recent years, research on DST has fallen into two broad categories, one based on PLMs and the other based on LLMs. The former focuses on BERT, T5, and GPT and primarily frames DST as either a classification task or a generative sequence-to-sequence task. Classification approaches (Rastogi et al., 2017; Zhong et al., 2018; Ma et al., 2019; Lee et al., 2019; Chao & Lane, 2019; Gao et al., 2020; Heck et al., 2020; Dai et al., 2021; Balaraman & Magnini, 2021) usually assume a fixed ontology and train a separate classifier for each slot, which limits them to the training vocabulary and weakens generalization to unseen domains. Generative approaches (Radford et al., 2019; Wu et al., 2019; Lin et al., 2021b;a; Li et al., 2021; Lee et al., 2021; Qixiang et al., 2022; Yu et al., 2022; Shin et al., 2022; Zhu et al., 2022; Wang et al., 2022; Madotto et al., 2021; Cho et al., 2023; Aksu et al., 2023; Luo et al., 2024) concatenate the dialogue context with the relevant slot information to model multiple domains, emphasizing zero-shot transfer, and they guide the model to generate slot values in unseen domains through natural language pattern prompts. The latter focuses on casting state tracking as a generative task that is prompt-driven, constrained by structured outputs, and able to generalize to unseen domains. Moreover, there is also work (Feng et al., 2023; Dong et al., 2024) that applies instruction tuning with LoRA to fine-tune LLMs so they become more robust to instructions. In this paper, we focus on zero-shot DST and propose a method that requires no training and introduces no additional or manually constructed data, thereby reducing the training burden on the LLM.

2.2 Activation Steering

The core idea of activation steering is to apply small, targeted linear interventions to intermediate activations during inference so that model behavior can be adjusted without training or controlled through knowledge editing. Existing work typically uses this technique to steer models to markedly modulate attributes such as sentiment, toxicity, sycophancy, and truthfulness without altering the weights (Rimsky et al., 2024; Chen et al., 2025; Gan et al., 2025; Su et al., 2025; Wang et al., 2025b; Scialanga et al., 2025; Liu et al., 2025; Tang et al., 2025). Many studies use steering to control model behavior. Stolfo et al. (2025) derive instruction-specific vectors from activation differences between instructed and non-instructed settings, which allows control over output format, length, and wording even without explicit textual prompts. SADI (Wang et al., 2025a) extracts key units from contrastive pairs, binarizes them into recognition masks, and applies interventions scaled to input semantics for precise guidance. MAT-STEER (Nguyen et al., 2025) gates variable-strength interventions at tokens linked to the target attribute and mitigates attribute conflicts with a distribution-alignment objective alongside orthogonality and sparsity constraints. CP (Cheng et al., 2025) introduces an additional auxiliary prompt and uses steering to elicit higher-quality sentence embeddings. In this work, we



(a) Slot Steering Vector Construction **(b) Inference-Time Slot Steering**

Figure 3: The whole process of the proposed method.

dynamically construct two complementary steering vectors: a slot-prototype contrastive vector for shared slots, and a global negative vector for target-only slots, which are injected during inference to directly mitigate domain and slot-understanding biases.

3 METHOD

In this paper, we propose StDST, a plug-and-play framework for zero-shot dialogue state tracking. StDST constructs a steering vector for each slot in the source domain by treating slots with the same name as positives and other slots as negatives, and also builds a global negative vector to handle target-only slots at inference time. During inference, it steers the corresponding layers so the representations move toward the target domain slots and away from others, which mitigates negative transfer. The whole process of the proposed method is shown in Figure 3.

3.1 POSITIVE-NEGATIVE EXAMPLE SELECTION

Unlike prior steering work, zero-shot DST datasets do not naturally contain positive–negative pairs, which makes it hard to build steering vectors directly from the data. We also avoid reconstructing a negative set with an LLM or manual annotation, which contradicts our training-free goal, and the long dialogue context typical of zero-shot DST would make such reconstruction cumbersome. Following the observation in DualLoRA (Luo et al., 2024) that models should prioritize slot information over dialogue context, we take slot information as the key signal. Since the target domain usually shares slot names with the source domain, we steer at the slot level, where same-named slots act as positives and differently named slots act as negatives. For each source-domain slot, we define the positive set $\mathcal{I}_s$ and the negative set $\mathcal{I}_s^c$. By collecting these sets solely from the source domain, we achieve steering without parameter updates or target-domain annotations. Let $\mathcal{I}_{\text{src}}$ be the set of source-domain inputs. For $i \in \mathcal{I}_{\text{src}}$, let $n(i)$ be the slot name of input i. We define

$$\begin{aligned} \mathcal{I}_s &= \{\, i \in \mathcal{I}_{\text{src}} : n(i) = s \,\}, \\ \mathcal{I}_s^c &= \{\, i \in \mathcal{I}_{\text{src}} : n(i) \neq s \,\} \end{aligned} \tag{1}$$

3.2 SLOT STEERING VECTOR CONSTRUCTION

Given the positive and negative sets in Section 3.1, we extract slot representations from source-domain data and build the steering direction vector.

3.2.1 SLOT PROTOTYPE CONTRASTIVE STEERING VECTOR

Since many slots are shared between the source and target domains, we construct a slot prototype contrastive steering vector and, during inference, steer target-domain slot representations toward their semantic prototypes and away from source-domain priors, thereby mitigating domain bias. The process is shown in Figure 3(a).

Formally, for input i, we write $T^{(i)} = [C^{(i)}_{1:m}; S^{(i)}_{1:n}]$, where C is the dialogue context and S is the slot description, and m, n are their lengths. Since mainstream zero-shot DST systems use the Transformer backbone, we feed $T^{(i)}$ into the Transformer and take the ℓ-th layer, the hidden states can be written as:

$$\mathbf{H}^{(\ell)}(T^{(i)}) = \big[\, \mathbf{H}^{(\ell)}(C^{(i)}_{1:m});\, \mathbf{H}^{(\ell)}(S^{(i)}_{1:n}) \,\big] \tag{2}$$

The slot representation at layer ℓ is

$$\mathbf{H}^{(\ell)}(S^{(i)}) = \big[h^{(\ell)}(S^{(i)}_1), \ldots, h^{(\ell)}(S^{(i)}_n)\big] \in \mathbb{R}^{n\times d} \tag{3}$$

where $h^{(\ell)}(S^{(i)}_t)$ is the hidden state of the t-th slot token and d is the hidden size. We average the token vectors to obtain the slot representation $x^{(i)}$:

$$x^{(i)} = \frac{1}{n}\sum_{t=1}^{n} h^{(\ell)}\big(S^{(i)}_t\big) \in \mathbb{R}^d \tag{4}$$

Let $n(i)$ denote the slot name of input i. Using the sets $\mathcal{I}_s$ and $\mathcal{I}^c_s$ from Section 3.1, the positive μ^+_s prototype and negative prototype μ^-_s for slot s are

$$\mu^+_s = \frac{1}{|\mathcal{I}_s|}\sum_{i\in\mathcal{I}_s} x^{(i)} \quad \mu^-_s = \frac{1}{|\mathcal{I}^c_s|}\sum_{i\in\mathcal{I}^c_s} x^{(i)} \tag{5}$$

For numerical stability, we apply ℓ_2-normalization with a small constant $\varepsilon > 0$:

$$\widehat{\mu}^{\pm}_s = \frac{\mu^{\pm}_s}{\|\mu^{\pm}_s\|_2 + \varepsilon} \tag{6}$$

where $\widehat{\mu}^+_s$ captures the semantics of slot s, and $\widehat{\mu}^-_s$ represents semantics unrelated to s aggregated from other slots in the source domains. The vector $\mathbf{v}_s$ encodes a direction in the slot-semantics space that emphasizes features specific to s and suppresses features shared with other slots, thereby mitigating domain bias. Specifically, the steering vector is formulated as:

$$\mathbf{v}_s = \alpha\,\widehat{\mu}^+_s - \beta\,\widehat{\mu}^-_s \tag{7}$$

where α and β weight the contributions of the positive and negative prototypes in the contrastive direction. Because slot prior strength varies across domains, the positive and negative prototypes can become overly strong or too weak. We therefore introduce two separate coefficients to balance their influence and control the steering direction.

3.2.2 Global Negative Steering Vector

Since the target domain still includes slots that are absent in the source domain, we construct a global negative vector to prevent these slots from over-relying on source-domain slot-value patterns and thereby reduce slot-understanding bias. Accordingly, we construct the steering vector for such slots as follows:

$$\mathbf{v}_{\text{neg}} \;=\; -\frac{1}{|\mathcal{I}_{\text{src}}|}\sum_{i\in\mathcal{I}_{\text{src}}} x^{(i)} \;\in\; \mathbb{R}^d \tag{8}$$

which aggregates source-domain slot vectors and is different from the slot negative prototype μ^-_s.

3.2.3 Unified Slot Steering Vector

Let $\mathcal{S}_{\text{src}}$ be the set of slot names observed in the source domains, and let $\mathcal{S}_{\text{tgt}}$ be the set of slot names in the target domain. For each $s \in \mathcal{S}_{\text{tgt}}$, we define the unified direction by checking whether s also appears in the source domains:

$$\mathbf{u}_s \;=\; \begin{cases} \mathbf{v}_s, & \text{if } s \in \mathcal{S}_{\text{src}}, \\ \mathbf{v}_{\text{neg}}, & \text{if } s \notin \mathcal{S}_{\text{src}}. \end{cases} \tag{9}$$

Because the target domain often shares a subset of slot names with the source domains, we handle each target-domain slot s as follows. If s appears in the source domains, we reuse the slot prototype contrast vector $\mathbf{v}_s$ constructed from same-named positives and differently named negatives in the source domains. If s does not appear in the source domains, we use the global negative vector $\mathbf{v}_{\text{neg}}$ to push the slot representation away from the source-domain representation. In both cases, $\mathbf{u}_s$ is injected at the slot-description positions during inference.

3.3 Inference-time Slot Steering

During inference, we inject the steering direction from Section 3.2 into layer ℓ at the slot-description positions so that the representation moves toward its semantic prototype or away from the background source space. The process is shown in Figure 3(b).

For the i-th input $T^{(i)} = \left[C^{(i)}_{1:m};\, S^{(i)}_{1:n}\right]$, we update the slot-description states as

$$\mathbf{H}^{(\ell)}(S^{(i)})^{*} \;=\; \mathbf{H}^{(\ell)}(S^{(i)}) \;+\; \lambda_s\, \mathbf{u}_s, \tag{10}$$

where λ_s controls the steering strength. For each s, we set $\lambda_s = \lambda_1$ if $s \in \mathcal{S}_{\text{src}}$, and $\lambda_s = \lambda_2$ otherwise in practice. The procedure relies only on slot descriptions and source-domain data and does not require target-domain annotations.

4 Experiments

4.1 Datasets and Metrics

Experiments are conducted on the MultiWOZ dataset (Budzianowski et al., 2018; Eric et al., 2020) and the Schema-Guided-Dialogue (SGD) dataset (Rastogi et al., 2020), the benchmarks in TODs. The MultiWOZ dataset, extensively annotated, includes 10,000 human-to-human conversations across domains and topics, establishing it as a prevalent benchmark for the evaluation of TODs. The SGD dataset is a large-scale benchmark that features over 20,000 conversations across multiple domains, which is designed to evaluate zero-shot generalization by natural language schemas.

In zero-shot cross-domain DST, models are trained on visible domains initially and tested on unseen domains subsequently. Building upon previous research, Joint Goal Accuracy (JGA) is employed to evaluate both the baseline and our model. JGA, the primary metric for DST evaluation, indicates the percentage of turns for which all slots in the same domain are correctly identified, thus serving as a measure of model performance.

4.2 Baselines

We combine our method with some baselines to demonstrate the effectiveness. Moreover, we present the mainstream models in zero-shot DST, such as **TRADE** (Wu et al., 2019), **SUMBT** (Lee et al., 2019), **SGD-baseline** (Rastogi et al., 2020), **Seq2Seq-DU** (Feng et al., 2021), **GPT2-DST** (Li et al., 2021), **T5DST** (Lin et al., 2021b), **TransferQA** (Lin et al., 2021a), **SlotDM-DST** (Wang et al., 2022), **Prompter** (Aksu et al., 2023), **DCC** (Wang et al., 2023), **DualLoRA** (Luo et al., 2024), **CAPID** (Dong et al., 2024), and **LDST** (Feng et al., 2023). More details about baselines are depicted in the Appendix B.

4.3 Implementation Details

To evaluate our approach across multiple frameworks, we use two general models in zero-shot DST, namely T5DST (Lin et al., 2021b), LDST (Feng et al., 2023) and (Dong et al., 2024), as base models. To verify effectiveness across different model sizes, we combine our method with T5-small, T5-base, and LLaMA-based models. For more details, the training details for T5DST are consistent with the original T5DST. The configuration used in the main experiments is fixed in advance. After fixing the main configuration, we conduct parameter sweeps over the steering strengths, intervention layers, and intervention positions as post-hoc sensitivity analyses to characterize the behavior and robustness of StDST. The main experimental configuration remains fixed throughout these analyses. To reduce computational cost, we conduct most of our ablation experiments with T5-small. The domains

Table 1: Zero-shot JGA (%) on MultiWOZ. The upper block reports published prior methods for reference, while the lower block compares baselines with their StDST-enhanced variants. Results with ± are averaged over three random seeds. Green arrows indicate absolute improvements over the corresponding baseline.

Model	Backbone	Attr.	Hotel	Rest.	Train	Taxi	Avg.
Prior methods							
SUMBT	BERT-base	22.6	19.1	16.5	22.5	59.5	28.0
GPT2-DST	GPT2-base	23.7	18.5	21.1	24.3	59.1	29.3
SlotDM-DST	T5-small	33.9	18.9	20.8	37.0	66.3	35.4
Prompter	PPTOD-small	35.8	19.2	26.0	39.0	66.3	37.2
DCC	T5-small	35.8	24.8	22.9	40.2	65.9	37.9
DualLoRA	PPTOD-small	37.1	18.9	27.9	42.4	67.2	38.7
StDST-enhanced variants							
T5DST	T5-small	36.4±.21	20.9±.12	21.5±.14	30.5±.18	64.3±.09	34.7±.07
T5DST + StDST	T5-small	37.1±.13	22.6±.18	21.9±.11	31.3±.08	64.4±.06	**35.5**±.05 ↑0.74
T5DST	T5-base	38.9±.24	19.8±.28	25.7±.16	32.1±.10	65.2±.08	36.3±.08
T5DST + StDST	T5-base	39.3±.18	22.9±.14	26.3±.14	32.7±.12	65.2±.06	**37.3**±.06 ↑0.94
LDST	Llama3-8B	80.2±.18	69.8±.16	71.5±.14	69.6±.20	91.3±.09	76.5±.07
LDST + StDST	Llama3-8B	81.3±.13	71.9±.15	72.4±.12	72.1±.16	91.8±.07	**77.9**±.06 ↑1.42
CAPID	Llama3-8B	84.6±.17	72.9±.18	79.6±.16	91.2±.11	92.1±.09	84.1±.07
CAPID + StDST	Llama3-8B	85.9±.12	73.8±.14	81.2±.13	91.9±.09	92.4±.06	**85.0**±.05 ↑0.96

selected from the SGD dataset are consistent with DualLoRA (Luo et al., 2024). All experiments were conducted on the NVIDIA V100 GPUs.

4.4 Main Result

The result of the proposed method on the MultiWOZ dataset is as shown in Table 1. It applies across multiple architectures and yields consistent improvements, which demonstrates the effectiveness of the proposed method. The gains vary across different backbones and domains, while StDST consistently improves the corresponding baseline in average JGA. This indicates that the proposed steering mechanism is not tied to a specific architecture. Figure 2 provides more detailed results by reporting the negative transfer rate at the slot level, and after steering, most slots show a clear reduction in negative transfer. As shown in Table 2, experiments on the SGD dataset show that our approach improves the average performance across multiple model architectures, validating the generalization capability of the proposed method. More experiments about recent LLMs are presented in the Appendix C. The experiments on the sensitivity are presented in the Appendix D.

4.5 Ablation Study

To assess the effectiveness of our method, we conduct ablation studies on the slot prototype contrastive steering vector $\mathbf{v}_s$ and global negative steering vector $\mathbf{v}_{\text{neg}}$. As shown in Table 3, the proposed method attains the best scores across five domains, while removing either component degrades performance. Removing $\mathbf{v}_s$ causes a clear performance drop, indicating that steering shared slots in the target domain effectively mitigates domain bias. Removing $\mathbf{v}_s$ weakens the correction of shared-slot domain bias, whereas removing $\mathbf{v}_{\text{neg}}$ mainly affects target-only slots and thus weakens the mitigation of slot-understanding bias. Overall, the experiments demonstrate the effectiveness of the component of the proposed method.

4.6 Analysis

The following experiments are conducted as post-hoc analyses of the fixed main configuration to examine the sensitivity of StDST to different design choices and intervention settings.

Table 2: Zero-shot JGA (%) on the SGD dataset. The upper block reports prior methods for reference, while the lower block compares baselines with their StDST-enhanced variants. Results with ± are averaged over three random seeds. Green arrows indicate absolute improvements over the corresponding baseline.

Model	Buses	Events	Flights	Media	Msg.	Music	Pay.	Trains	Avg.
Prior methods									
SGD-baseline	9.7	23.5	23.9	18.0	10.2	15.5	11.5	13.6	15.7
Seq2Seq-DU	16.8	31.9	15.9	23.1	4.9	12.3	7.2	16.8	16.1
TransferQA	15.9	15.6	3.6	30.2	13.3	8.9	24.7	17.4	16.2
StDST-enhanced variants									
T5-small	43.3±.12	45.1±.10	16.1±.15	62.6±.18	51.3±.11	22.4±.07	21.7±.21	45.6±.23	38.6±.05
T5-small + StDST	44.6±.10	46.3±.12	17.2±.19	63.5±.15	50.7±.09	22.9±.15	22.6±.15	45.9±.10	**39.2**±.05 ↑0.61
T5-base	42.5±.10	52.7±.08	19.5±.14	77.2±.12	74.4±.22	38.4±.14	23.4±.10	50.3±.20	47.3±.05
T5-base + StDST	43.8±.08	53.8±.08	21.3±.06	76.9±.17	74.5±.10	39.1±.06	24.1±.11	51.1±.19	**48.0**±.04 ↑0.70
Llama3.2-1B	48.3±.08	56.6±.07	24.4±.06	62.0±.12	70.9±.18	41.6±.11	29.9±.20	45.7±.17	47.4±.05
Llama3.2-1B + StDST	49.5±.05	58.1±.04	24.3±.08	63.0±.06	71.2±.16	42.6±.16	30.9±.12	46.8±.19	**48.3**±.04 ↑0.85

Table 3: Ablation results on the MultiWOZ dataset. Removing either the slot prototype contrastive vector or the global negative vector degrades performance.

Model	Attr.	Hotel	Rest.	Train	Taxi
T5DST + StDST	**37.1**	**22.6**	**21.9**	**31.3**	**64.4**
w/o $\mathbf{v}_s$	36.6	22.0	21.8	31.1	64.3
w/o $\mathbf{v}_{\text{neg}}$	36.6	20.8	21.5	30.7	64.3

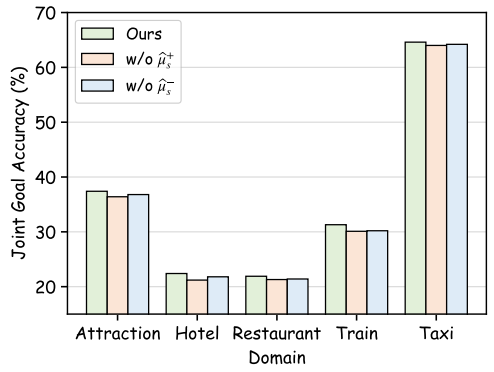
Effect of μ^+ and μ^-. To assess the effectiveness of our method, we conduct experiments on the positive prototype μ^+ and the negative prototype μ^-. As shown in Figure 4a, the proposed method attains the best scores across five domains, while removing either component degrades performance. The effect of removing the negative prototypes indicates that pushing away from non-target semantics helps suppress cross-domain interference, but is secondary to the positive pull. In contrast, removing the positive prototype causes larger drops, suggesting that anchoring toward $\widehat{\mu}_s^+$ is the dominant factor for mitigating domain bias.

Effect of Scaling Factor λ_1. To analyze the scaling factor λ_1 for each domain, we evaluated λ_1 at 0, 0.2, 0.4, 0.6, 0.8, and 1. As shown in Figure 4b, performance does not improve monotonically with larger λ_1, indicating that excessive steering can degrade the model. More experiments about α, β and λ_2 are depicted in the Appendix J and Appendix H.

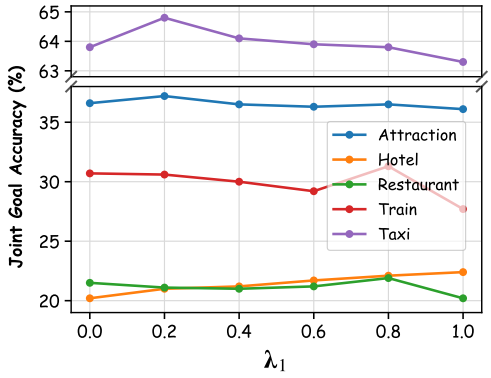
Effect of Steering Layer. Because layers encode different information, that higher layers emphasize global abstractions while lower ones capture local details, we evaluated constructing and injecting the steering vector at various layers. As shown in Table 4, higher layers capture more comprehensive features and consistently deliver larger gains across most domains. Accordingly, we use the final layer for both constructing and injecting the steering vector.

Effect of Steering Position. To find the best steering position, we experiment with three sites: the feed-forward network output (FFN), the multi-head attention output (Head), and the layer's residual-stream output (Hidden). As shown in Figure 4c, our method outperforms the baseline at all three positions. Among them, intervening on the hidden state works best, likely because hidden state perturbations propagate through residual connections and influence all attention layers.

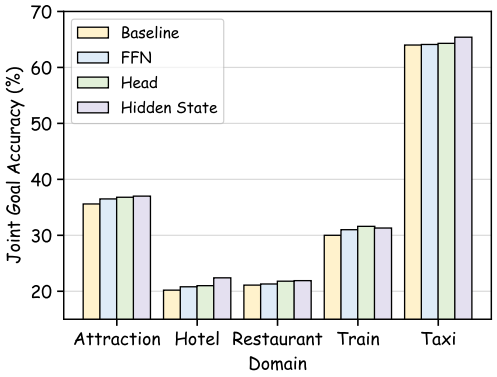
Effect of Steering under Slot-Description Ablation. As shown in Figure 4d, we ablate the natural-language slot descriptions and evaluate a minimal setting in which the domain is concatenated with the slot to serve as the cue, such as *attraction-name*. Across all five domains, steering consistently outperforms the baseline, indicating that the learned slot-specific steering direction acts as a semantic prior even without the textual slot description. In domains with many shared slots, such as attraction and taxi, our method yields more pronounced gains, while the other domains experience modest



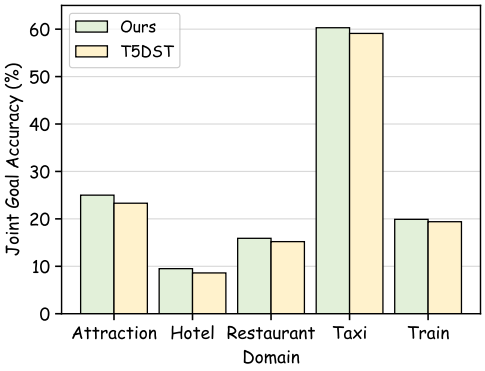
(a) Effect of μ_s^+ and μ_s^-.



(b) Effect of scaling factor λ_1.



(c) Effect of steering position.



(d) Effect under slot-description ablation.

Figure 4: Analysis of key design choices in StDST. The results show the effects of prototype components, steering strength, intervention position, and slot-description ablation.

Table 4: Effect of the layer used for construction and injection.

Layer	Attr.	Hotel	Rest.	Train	Taxi
-1	37.1	22.6	21.9	31.3	64.4
-2	36.6	21.8	21.6	31.6	64.5
-3	36.7	21.3	21.6	31.5	64.6
-4	36.6	21.1	21.6	30.7	64.5

yet consistent improvements. Beyond this setting, we also conducted cross-slot experiments. In inference, we applied steering to different slot-description types, such as *naive* and *slot type*. Detailed results and analysis are provided in the Appendix K.

5 Conclusion

In this paper, we present StDST, a plug-and-play steering method for zero-shot cross-domain DST. StDST requires no extra data or manually constructed datasets. We construct two slot steering mechanisms to mitigate domain bias and slot understanding bias, namely the slot prototype contrastive steering vector and the global negative steering vector. During target-domain inference, we inject this vector into Transformer layers to guide the model toward more accurate dialogue states. Experiments on MultiWOZ and SGD datasets show that StDST integrates with diverse baselines and consistently improves their performance.

AI USE STATEMENT

In this work, we used generative AI tools for feedback on research methodology and experimental design, analysis of experimental results, code editing and debugging, and manuscript editing. All AI-assisted outputs were reviewed and, where applicable, verified by the authors before being incorporated into the work. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 3 describes the proposed StDST method, and Section 4 provides the datasets, evaluation metrics, baselines, implementation details, ablation studies, and analyses. Sections A–L provide additional dataset and baseline details, supplementary experiments, and sensitivity and robustness analyses. Anonymous source code is also provided with the submission.

REFERENCES

Ibrahim Taha Aksu, Min-Yen Kan, and Nancy Chen. Prompter: Zero-shot Adaptive Prefixes for Dialogue State Tracking Domain Adaptation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4588–4603. Association for Computational Linguistics, July 2023. doi: 10.18653/v1/2023.acl-long.252. URL https://aclanthology.org/2023.acl-long.252.

Vevake Balaraman and Bernardo Magnini. Domain-aware dialogue state tracker for multi-domain dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 866–873, 2021.

Pawe\l Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5016–5026. Association for Computational Linguistics, October 2018. doi: 10.18653/v1/D18-1547. URL https://aclanthology.org/D18-1547.

Guan-Lin Chao and Ian Lane. BERT-DST: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. *arXiv preprint arXiv:1907.03040*, 2019.

Yongchao Chen, Srinagesh Sharma, Chuchu Fan, Harsh Jhamtani, and Chi Wang. Steering large language models between code execution and textual reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zifeng Cheng, Zhonghui Wang, Yuchen Fu, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. Contrastive Prompting Enhances Sentence Embeddings in LLMs through Inference-Time Steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3475–3487. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.174. URL https://aclanthology.org/2025.acl-long.174/.

Hyundong Cho, Andrea Madotto, Zhaojiang Lin, Khyathi Chandu, Satwik Kottur, Jing Xu, Jonathan May, and Chinnadhurai Sankar. Continual dialogue state tracking via example-guided question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3873–3886. Association for Computational Linguistics, December 2023. doi: 10.18653/v1/2023.emnlp-main.235. URL https://aclanthology.org/2023.emnlp-main.235/.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. Preview, Attend and Review: Schema-Aware Curriculum Learning for Multi-Domain Dialogue State Tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 879–885, 2021.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language*

Processing, pp. 8527–8540. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.485. URL https://aclanthology.org/2024.emnlp-main.485/.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 422–428. European Language Resources Association, May 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.53.

Yue Feng, Yang Wang, and Hang Li. A Sequence-to-Sequence Approach to Dialogue State Tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1714–1725. Association for Computational Linguistics, August 2021. doi: 10.18653/v1/2021.acl-long.135. URL https://aclanthology.org/2021.acl-long.135.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. Towards LLM-driven Dialogue State Tracking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 739–755, 2023.

Jinwei Gan, Zifeng Cheng, Zhiwei Jiang, Cong Wang, Yafeng Yin, Xiang Luo, Yuchen Fu, and Qing Gu. Steering when necessary: Flexible steering large language models with backtracking. *arXiv preprint arXiv:2508.17621*, 2025.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. From Machine Reading Comprehension to Dialogue State Tracking: Bridging the Gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 79–89, 2020.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 35–44. Association for Computational Linguistics, July 2020. doi: 10.18653/v1/2020.sigdial-1.4. URL https://aclanthology.org/2020.sigdial-1.4.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A Smith, and Mari Ostendorf. In-Context Learning for Few-Shot Dialogue State Tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2627–2643, 2022.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue State Tracking with a Language Model using Schema-Driven Prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4937–4949, 2021.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5478–5483. Association for Computational Linguistics, July 2019. doi: 10.18653/v1/P19-1546. URL https://aclanthology.org/P19-1546.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. Zero-shot Generalization in Dialog State Tracking through Generative Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1063–1074. Association for Computational Linguistics, April 2021. doi: 10.18653/v1/2021.eacl-main.91. URL https://aclanthology.org/2021.eacl-main.91.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and others. Zero-Shot Dialogue State Tracking via Cross-Task Transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021a.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul A Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. Leveraging Slot Descriptions for Zero-Shot Cross-Domain Dialogue StateTracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5640–5648, 2021b.

Sheng Liu, Haotian Ye, and James Zou. REDUCING HALLUCINATIONS IN LARGE VISION-LANGUAGE MODELS VIA LATENT SPACE STEERING. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiang Luo, Zhiwen Tang, Jin Wang, and Xuejie Zhang. Zero-shot cross-domain dialogue state tracking via dual low-rank adaptation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5746–5765. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.312. URL https://aclanthology.org/2024.acl-long.312/.

Yue Ma, Zengfeng Zeng, Dawei Zhu, Xuan Li, Yiying Yang, Xiaoyuan Yao, Kaijie Zhou, and Jianping Shen. An end-to-end dialogue state tracking system with machine reading comprehension and wide & deep classification. *arXiv preprint arXiv:1912.09297*, 2019.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. Continual Learning in Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7452–7467. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.emnlp-main.590. URL https://aclanthology.org/2021.emnlp-main.590.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20619–20634. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1007. URL https://aclanthology.org/2025.acl-long.1007/.

Gao Qixiang, Guanting Dong, Yutao Mou, Liwen Wang, Chen Zeng, Daichi Guo, Mingyang Sun, and Weiran Xu. Exploiting domain-slot related keywords description for Few-Shot Cross-Domain Dialogue State Tracking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2460–2465. Association for Computational Linguistics, December 2022. doi: 10.18653/v1/2022.emnlp-main.157. URL https://aclanthology.org/2022.emnlp-main.157.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and others. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 561–568. IEEE, 2017.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8689–8696, 2020.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via Contrastive Activation Addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Marco Scialanga, Thibault Laugel, Vincent Grari, and Marcin Detyniecki. SAKE: Steering activations for knowledge editing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15966–15978. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.777. URL https://aclanthology.org/2025.acl-long.777/.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. Dialogue Summaries as Dialogue States (DS2), Template-Guided Summarization for Few-shot Dialogue State Tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3824–3846. Association for Computational Linguistics, May 2022. doi: 10.18653/v1/2022.findings-acl.302. URL https://aclanthology.org/2022.findings-acl.302.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving Instruction-Following in Language Models through Activation Steering. In *The Thirteenth International Conference on Learning Representations*. Qeios Ltd, November 2025.

Jingran Su, Jingfan Chen, Hongxin Li, Yuntao Chen, Li Qing, and Zhaoxiang Zhang. Activation Steering Decoding: Mitigating Hallucination in Large Vision-Language Models through Bidirectional Hidden State Intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12964–12974. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.634/.

Xinyu Tang, Zhihao Lv, Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Enhancing cross-task transfer of large language models via activation steering. *arXiv preprint arXiv:2507.13236*, 2025.

Qingyue Wang, Yanan Cao, Piji Li, Yanhe Fu, Zheng Lin, and Li Guo. Slot dependency modeling for zero-shot cross-domain dialogue state tracking. In *Proceedings of the 29th International conference on computational linguistics*, pp. 510–520, 2022.

Qingyue Wang, Liang Ding, Yanan Cao, Yibing Zhan, Zheng Lin, Shi Wang, Dacheng Tao, and Li Guo. Divide, Conquer, and Combine: Mixture of Semantic-Independent Experts for Zero-Shot Dialogue State Tracking. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2048–2061. Association for Computational Linguistics, July 2023. doi: 10.18653/v1/2023.acl-long.114. URL https://aclanthology.org/2023.acl-long.114.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Xintong Wang, Jingheng Pan, Liang Ding, Longyue Wang, Longqin Jiang, Xingshan Li, and Chris Biemann. CogSteer: Cognition-inspired selective layer intervention for efficiently steering large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25507–25522. Association for Computational Linguistics, July 2025b. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1308. URL https://aclanthology.org/2025.findings-acl.1308/.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 808–819, 2019.

Dian Yu, Mingqiu Wang, Yuan Cao, Laurent El Shafey, Izhak Shafran, and Hagen Soltau. Knowledge-grounded Dialog State Tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3428–3435. Association for Computational Linguistics, December 2022. doi: 10.18653/v1/2022.findings-emnlp.250. URL https://aclanthology.org/2022.findings-emnlp.250.

Victor Zhong, Caiming Xiong, and Richard Socher. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1458–1467. Association for Computational Linguistics, July 2018. doi: 10.18653/v1/P18-1135. URL https://aclanthology.org/P18-1135.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. Continual Prompt Tuning for Dialog State Tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1124–1137, 2022.

A Dataset Statistic

We report the statistics for the MultiWOZ dataset used in our study in Table 5. Because the hospital and police domains are small and appear only in the training set, we conduct experiments on the remaining five domains. In addition, we list all the slots in each domain.

Moreover, Table 6 presents statistics for the unseen domains in the SGD dataset, including the total number of dialogues, the corresponding slot counts, and the number of slots overlapping with source domains.

Table 5: The statistics of the MultiWOZ dataset.

Domain	Train	Dev	Test	Slots
Attraction	2717	401	395	`area` `name` `type`
Hotel	3381	416	394	`area` `book day` `book people` `book stay` `internet` `name` `parking` `pricerange` `stars` `type`
Restaurant	3813	438	437	`area` `book day` `book people` `book time` `food` `name` `pricerange`
Taxi	1654	207	195	`arrive by` `departure` `destination` `leaveat`
Train	3103	484	494	`arriveby` `book people` `day` `departure` `destination` `leaveat`

Table 6: The statistics of the unseen domains in the SGD dataset. **Total Slots** indicates the number of slots in the domain, and **Overlapped** refers to the number of slots that also appear in the seen domains.

Domain	Dialogues	Total Slots	Overlapped
Buses	3,135	10	6
Events	4,519	9	8
Flights	3,644	13	12
Media	1,656	4	2
Messaging	298	2	1
Music	1,833	6	4
Payment	222	4	1
Trains	350	10	3

B Baseline Models

In this section, we describe each baseline in detail.

- **SUMBT**. SUMBT (Lee et al., 2019) learns attention-based relations between domain and slot types and candidate values in utterances using contextual embeddings, and predicts slot values in a nonparametric way, enabling a universal and scalable belief tracker that flexibly accommodates new values and ontologies.
- **SGD-Baseline**. SGD-Baseline (Rastogi et al., 2020) employs a single unified BERT model that conditions predictions on dynamic schema elements (intents and slots) via their natural language descriptions. By projecting dialogue context and schema embeddings into a shared semantic space, it enables zero-shot generalization to unseen services and APIs without requiring domain-specific parameters.
- **Seq2Seq-DU**. Seq2Seq-DU (Feng et al., 2021) formalizes dialogue state tracking as a sequence-to-sequence problem, employing dual BERT-based encoders for utterances and schema descriptions linked by an attention mechanism. By utilizing a decoder with a pointer generation mechanism, it jointly models intents and slots to effectively handle categorical, non-categorical, and unseen schemas.

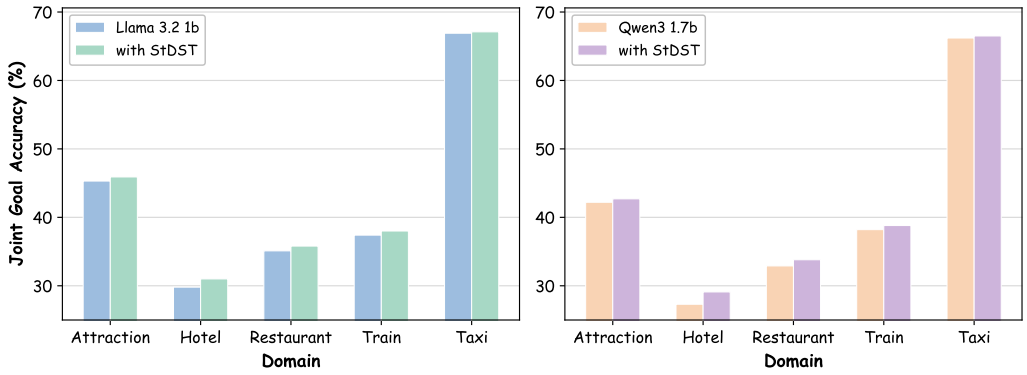


Figure 5: Recent LLMs' performance on the MultiWOZ dataset.

- **GPT2-DST**. GPT2-DST (Li et al., 2021) frames DST as ontology-free generative question answering with a pre-trained conditional language model, using natural-language slot queries to handle unseen constraints and slots and to generalize to new domains.
- **T5DST**. T5DST (Lin et al., 2021b) encodes dialogue context and slot descriptions with a pre-trained self-attentive encoder and generates slot values autoregressively, using slot-type-informed descriptions to share information across slots and enable zero-shot cross-domain transfer.
- **TransferQA**. TransferQA (Lin et al., 2021a) frames DST as generative question answering with a text-to-text transformer that unifies extractive and multiple choice QA, transfers knowledge from general QA corpora, and constructs unanswerable questions through negative sampling and context truncation so it can track both categorical and noncategorical slots and handle none values in zero-shot settings.
- **SlotDM-DST**. SlotDM-DST (Wang et al., 2022) models slot-to-slot, slot-to-value, and slot-to-context dependencies by combining slot-specific and shared slot parts with value demonstrations and constraint objects, enabling zero-shot generalization to unseen domains through shared cross-domain knowledge.
- **Prompter**. Prompter (Aksu et al., 2023) generates dynamic prefixes from target domain slot descriptions and inserts them into each layer's self-attention keys and values via prefix tuning, enabling zero-shot domain adaptation for DST and more reliable handling of non-valued slots.
- **DCC**. DCC (Wang et al., 2023) splits seen data into semantically independent subsets, trains a T5-adapter expert for each subset, and combines them with mixture-of-experts and ensemble inference to disentangle semantics and enable zero-shot generalization to unseen domains without external knowledge.
- **DualLoRA**. DualLoRA (Luo et al., 2024) adds two low-rank adaptation modules that separately target dialogue context processing and prompt optimization, maintaining prompt influence across transformer layers without extra inference latency and enabling zero-shot DST in new domains.
- **LDST**. LDST (Feng et al., 2023) fine-tunes smaller open source LLMs with domain slot instruction tuning to read, extract, and update dialogue states, enabling zero-shot and few-shot DST across domains while allowing local deployment and reducing privacy and request limit constraints.
- **CAPID**. CAPID (Dong et al., 2024) auto-generates dynamic, context-aware slot queries from the current dialogue and uses instruction-following contrastive decoding to penalize generations that deviate from the provided slot instructions.

Table 7: Sensitivity analysis under reduced prototype construction data and slot description interference.

Model	Attraction	Hotel	Restaurant	Train	Taxi
Reduced prototype construction data					
T5DST	36.4	20.9	21.5	30.5	64.3
T5DST + StDST	**36.7**	**21.5**	**21.7**	**31.0**	**64.3**
Slot description interference					
T5DST	36.4	20.9	21.5	30.5	64.3
T5DST + StDST	**36.9**	**22.0**	**21.8**	**31.1**	**64.4**

Table 8: Slot accuracy of the non-overlapping slot on the Multiwoz dataset.

Model	Hotel-Parking	Hotel-Stars	Hotel-Internet	Restaurant-Food	Train-day
T5-small	70.6	66.1	**70.3**	43.8	87.4
T5-small + StDST	**71.1**	**66.4**	70.1	**44.2**	**88.0**
T5-base	70.7	65.6	**68.9**	40.8	58.9
T5-base + StDST	**71.4**	**66.2**	68.8	**41.1**	**59.4**
Llama3.2 1b	78.5	81.0	71.7	48.8	88.9
Llama3.2 1b + StDST	**79.1**	**81.7**	**72.0**	**49.4**	**90.2**

C Model Performance on Recent LLMs

To validate our approach on the recent LLMs, we conducted experiments on the MultiWOZ dataset using recent LLMs, specifically Llama 3.2 1B and Qwen3 1.7B. As shown in Figure 5, our method demonstrates effective performance improvements even when applied to these recent LLMs.

D Sensitivity of Slot Prototypes

To investigate the sensitivity of the proposed method to slot prototypes, we have supplemented two experiments to demonstrate that our method can still ensure effectiveness when conditions fluctuate. We explored the effectiveness of the proposed method by significantly reducing the number of samples (i.e., using only 10% of the training data) for constructing the slot prototypes. As shown in Table 7, even with a small dataset, our method can still construct effective slot prototypes and improve model performance.

Moreover, we investigated the anti-interference ability of the proposed method by replacing 10% of the characters in slot descriptions with typos for constructing the slot prototypes. The results, presented in Table 7, indicate that our method can still improve model performance in the presence of noise, reflecting the anti-interference capability of the constructed slot prototypes.

E Negative Transfer Rate Analysis

To better analyze whether the model incorrectly reuses source-domain patterns in target domains, we report the negative transfer (NT) rate. Joint Goal Accuracy (JGA) measures whether the predicted dialogue state is correct, but it does not distinguish where the errors come from. In zero-shot cross-domain DST, a common error is that the model predicts a value that appears in the source domains but is incorrect for the target-domain slot. This indicates that the model may rely on source-domain value patterns instead of the target-domain dialogue context. Therefore, the NT rate measures the proportion of incorrect predictions whose predicted values belong to the source-domain value set. Formally, let $\hat{v}_i$ and v_i denote the predicted and gold value for the i-th target-domain slot prediction, respectively. Let $\mathcal{V}_{\text{src}}$ denote the set of values appearing in the source domains. The NT rate is defined as:

$$\text{NT} = \frac{\sum_i \mathbf{1}\left(\hat{v}_i \neq v_i \wedge \hat{v}_i \in \mathcal{V}_{\text{src}}\right)}{\sum_i \mathbf{1}\left(\hat{v}_i \neq v_i\right)}. \tag{11}$$

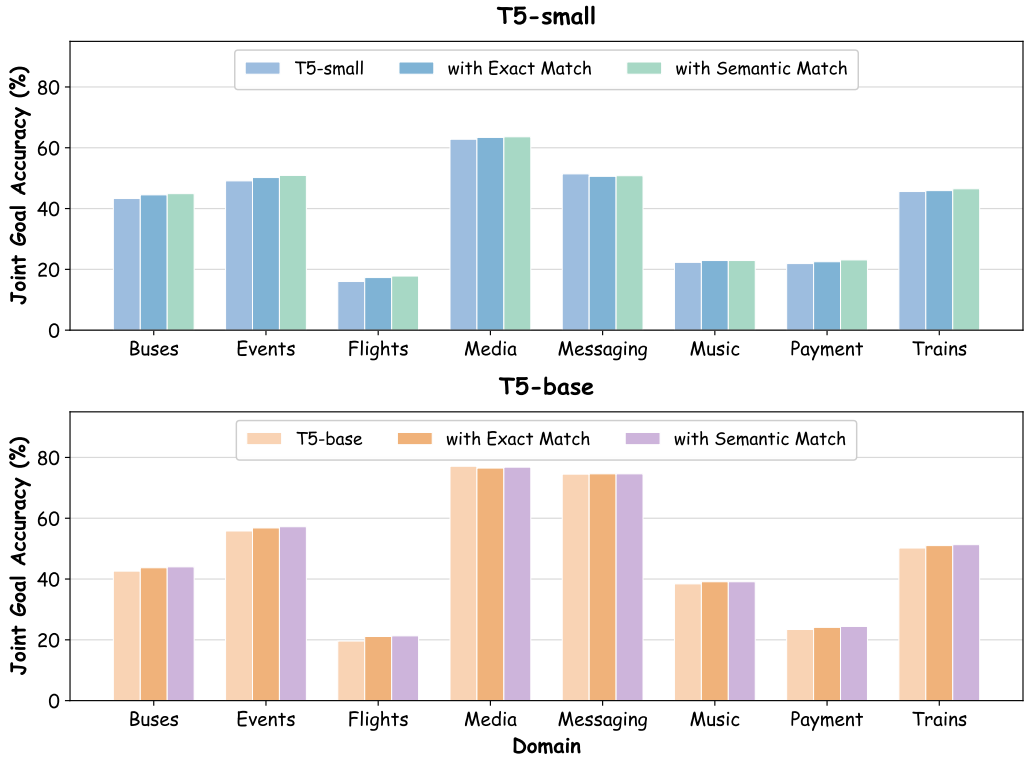


Figure 6: Effect of various matching methods on the SGD dataset.

A lower NT rate indicates that fewer errors are caused by incorrectly reusing source-domain values. Figure 7 reports the overall NT rate of different backbones before and after applying StDST. Across all evaluated models, StDST consistently reduces the NT rate. For example, on T5-small, the NT rate decreases from 24.8% to 18.9%, and on Llama3.2-3B, it decreases from 38.8% to 34.2%. These results demonstrate that StDST directly mitigates source-domain reuse errors, which further supports our motivation of reducing negative transfer in zero-shot cross-domain DST.

F Performance on Non-overlapping Domains.

To investigate completely non-overlapping domains, we adopt a domain from the MultiWOZ dataset that is fully non-overlapping with other domains—specifically, the hospital domain—for the relevant experiment. Notably, this domain contains only one slot: department. To construct such a domain, we carefully separated it from the training set and trained the backbone model exclusively using data from other domains. Even in this domain, the issue of reusing slot patterns from source domains persists. The goal of our method is to reduce erroneous reuse while leveraging knowledge from the source domain. For such completely slot-non-overlapping new domains, leveraging source domain knowledge is inherently challenging. However, as shown in the Table 9, our method can still mitigate erroneous reuse to a certain extent.

Table 9: Model performance on the hospital domain in the MultiWOZ dataset.

Model	Hospital
T5-DST	60.8
T5-DST + StDST	**61.1**

G Effect of Various Matching Methods.

In the MultiWOZ dataset, slot names across different domains are exactly matched; thus, we adopt an exact matching strategy. However, the scenario is more complex in the SGD dataset. To investigate

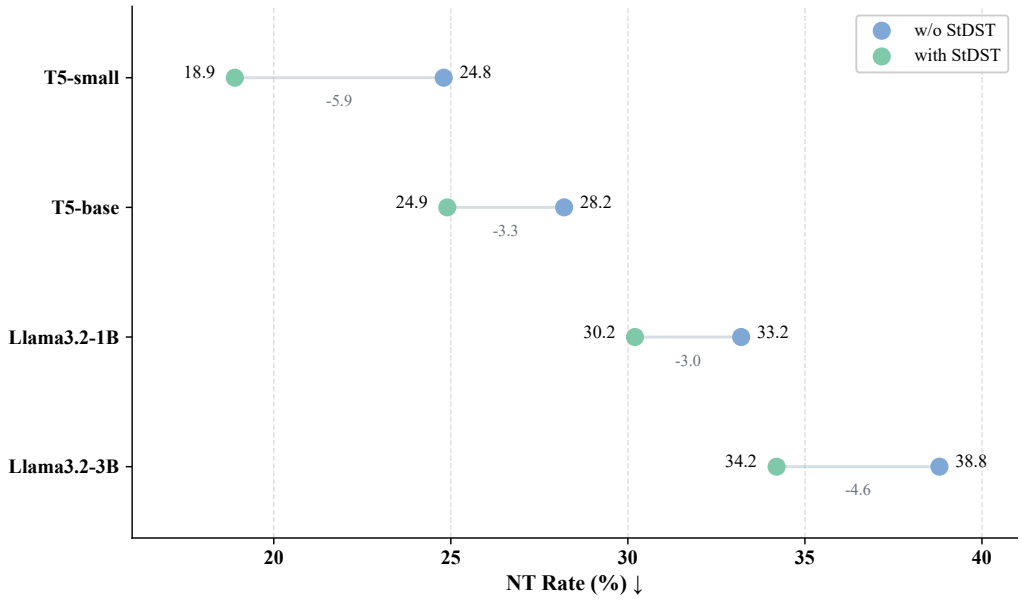


Figure 7: Overall negative transfer (NT) rate across different backbones. NT rate measures the proportion of incorrect predictions whose predicted values appear in the source domains. Lower values indicate fewer source-domain reuse errors. StDST consistently reduces the NT rate across all evaluated backbones.

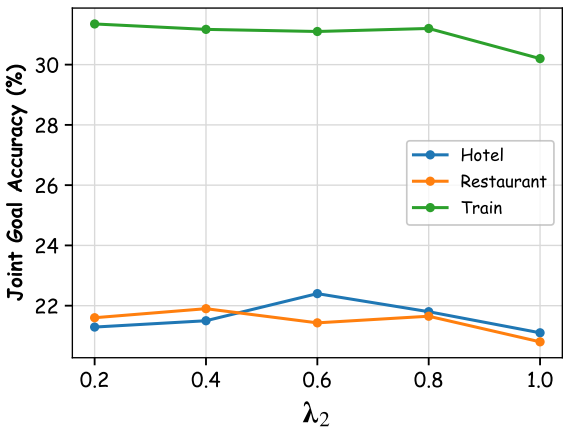
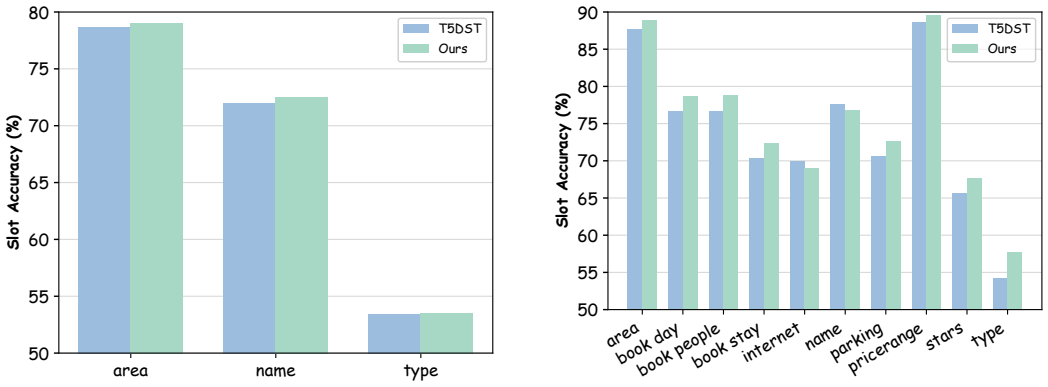


Figure 8: Effects of scaling factor λ_2 on the proposed method.

whether our method remains effective under semantic matching settings, we conducted relevant experiments on SGD, and the results are presented in the Figure 6. Specifically, we integrated the all-MiniLM-L6-v2 model from sentence-transformers to measure the semantic similarity between two slot descriptions, with the threshold set to 0.7. Experimental results on the SGD dataset demonstrate that our method can effectively improve the performance of various models through semantic matching, thereby verifying the flexibility and effectiveness of the proposed approach.

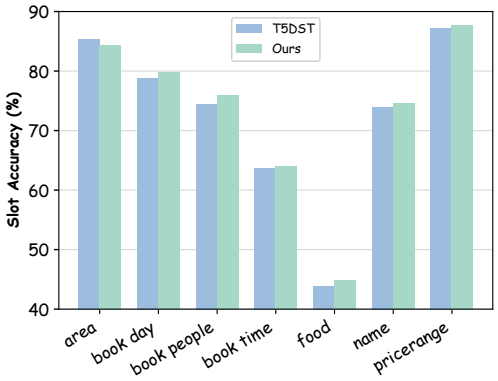
H Effect of Scaling Factor λ_2

To analyze the scaling factor λ_2 for three domains, we experiment with λ_2 at 0.2, 0.4, 0.6, 0.8, and 1. As shown in Figure 8, performance on these three domains does not improve as λ_2 increases, indicating that excessive steering can degrade the model.

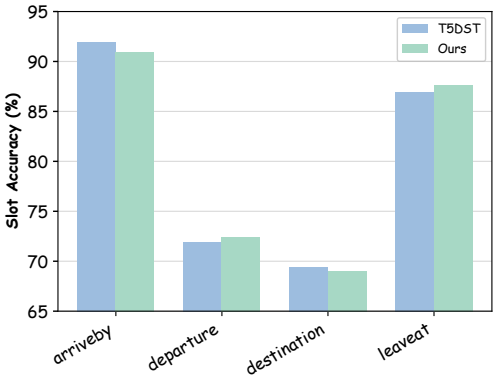


(a) Attraction.

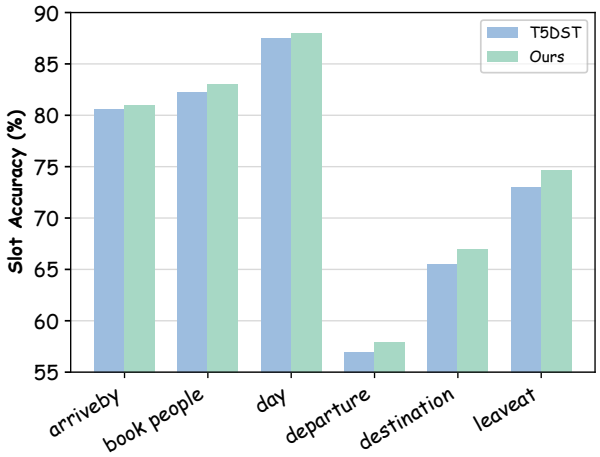
(b) Hotel.



(c) Restaurant.



(d) Taxi.



(e) Train.

Figure 9: Slot-level accuracy across five MultiWOZ domains.

I Slot Accuracy on MultiWOZ

Overall, the proposed method improves on most slots across domains as shown in Figure 9a, Figure 9b, Figure 9c, Figure 9d, and Figure 9e. In the attraction domain, the lift is small but uniform across all three slots. The hotel domain shows the greatest relative improvements, especially on type and booking-related slots. The restaurant domain is broadly positive: occasional dips are offset by gains

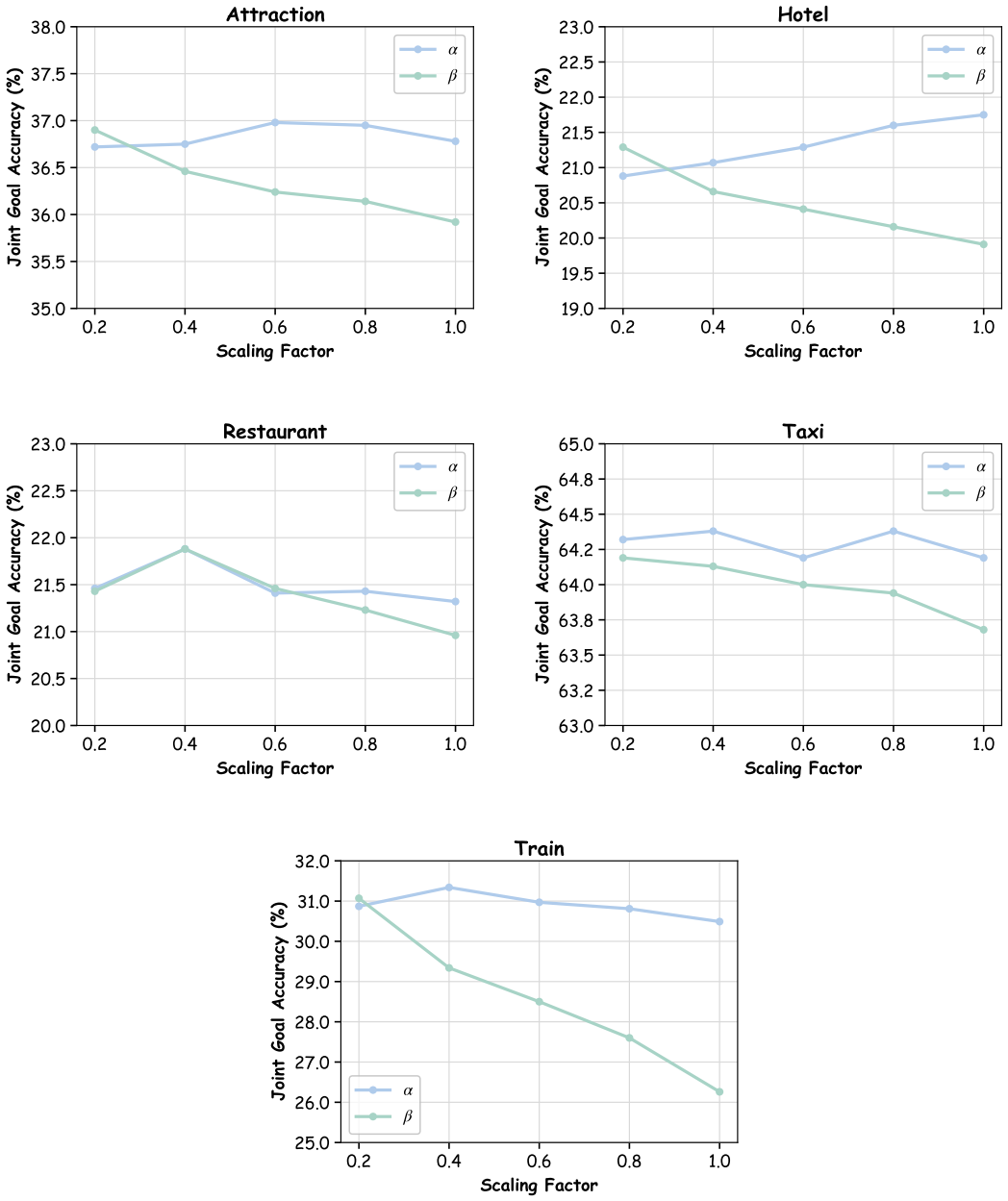


Figure 10: Sensitivity analysis of scaling factors α and β across five MultiWOZ domains.

in booking fields and food. The train domain improves consistently across all of its slots, particularly time and place fields like leaveat and destination. The taxi domain is mixed: departure and leaveat trend upward while arriveby and destination are roughly flat or slightly down, leaving the domain essentially unchanged overall.

Furthermore, to investigate the performance of our proposed method on non-overlapping slots, we conducted additional experiments, as illustrated in Figure 9. For domain-specific slots (e.g., hotel-stars, restaurant-food) that are non-overlapping, our method can mitigate slot-understanding bias and thereby improve model performance. The evaluation metric used for this experiment is slot accuracy (%).

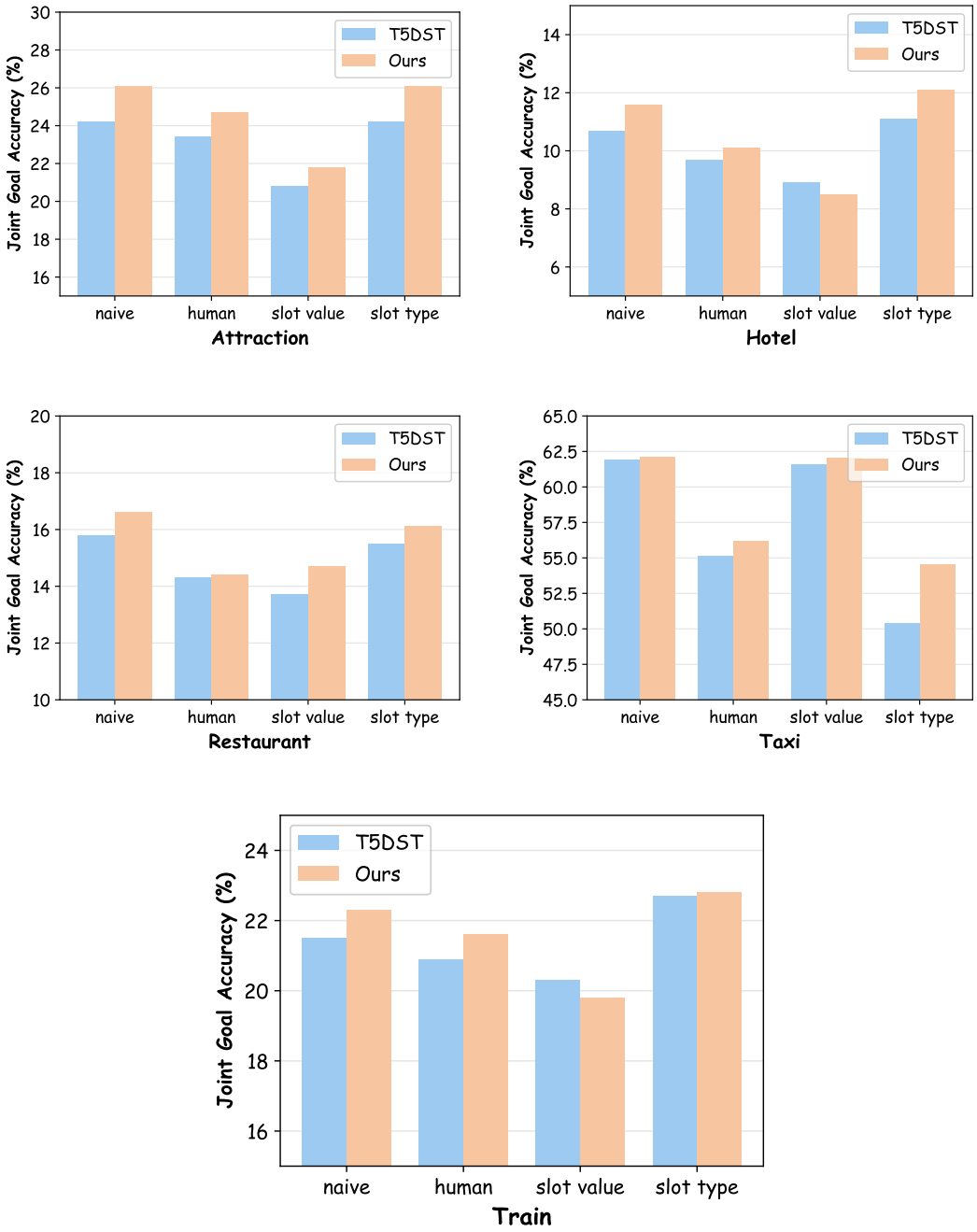


Figure 11: Effect of cross-slot description types across five MultiWOZ domains.

Taken together, these patterns indicate that the proposed method delivers consistent, domain-general benefits without delicate fine-tuning.

J Effect of Scaling Factors α and β

To investigate the scaling factor α and β across these domains, we conducted experiments with α and β at 0.2, 0.4, 0.6, 0.8, and 1. As shown in Figure 10a, Figure 10b, Figure 10c, Figure 10d, and Figure 10e, across domains, increasing α typically yields comparable or slightly improved

performance because moving toward the positive prototype is beneficial, whereas increasing β in most domains leads to a monotonic decline or degradation.

K Effect of Cross Slot Description Type

In this paper, we use "question" slot descriptions, so we examine whether slot steering vectors collected under this setting transfer to other description types. We run experiments across multiple domains with different slot-description forms, and the results are shown in Figure 11a, Figure 11b, Figure 11c, Figure 11d, and Figure 11e. Our slot steering vectors improve performance for the vast majority of domains and description types. The main exceptions are the "slot type" descriptions in the hotel and train domains, where performance drops slightly, which we attribute to the limited detail of those descriptions. In summary, our slot steering vector works not only with the same type of slot description but also with different description types, indicating that it captures the slot's semantic direction rather than being tied to any specific description template.

L Effect of Cross-Layer Construct–Injects

To verify the robustness, we report cross-layer construct–inject grids for Attraction, Hotel, Restaurant, Taxi, and Train, where rows are the source layer used to construct the steering vector, and columns are the target layer used to inject it. As shown in Figure 12, these figures show that model performance is strong and consistent near the diagonal, indicating that high-level semantic subspaces are coherent and transferable and that performance is robust to the choice of layer. With the injection layer fixed, varying the construction layer causes only small fluctuations; differences across injection layers are more pronounced, so the injection layer is more critical than the construction layer, but constructing and injecting at the same layer remains competitive.

In the attraction domain, cross-layer performance is close to the same-layer ones, with optima typically near the diagonal, indicating consistent and transferable higher-layer subspaces. In the hotel domain, after fixing a particular injection layer, most construction layers show similar performance, with combinations near the diagonal occasionally doing better, and constructing and injecting at the same layer still yielding strong results. In the restaurant grid, model performance is almost identical across the board, with negligible differences between diagonal and off-diagonal cells, indicating insensitivity to layer choice. The taxi domain shows overall high and tightly distributed performance, with a slight advantage for models near the diagonal. For the train domain, the optimal region concentrates near the diagonal, and deeper construction layers paired with higher injection layers tend to improve performance, which matches the intuition that higher layers encode more abstract features that can be effectively guided from neighboring layers. Taken together, these patterns justify our default choice in the main text, construct and inject at the final layer, while confirming that cross-layer construct–inject remains stable.

M Case Study

As shown in Figure 13, we present a dialogue from the MultiWOZ dataset to illustrate the states produced by our method and T5DST in zero-shot DST. With the hotel domain as the target domain, both models generate most states correctly. Since T5DST is trained on source domains, it is more likely to predict the hotel name as "Cambridge" from the source domain, whereas our method steers the slot representation and predicts the name correctly. And for the hotel slot type, T5DST also over-predicts the value from the attraction domain, while our method uses steering to track the intended slot value accurately.

N Limitations

The following limitations remain in StDST, and tackling them in future work will enhance its broader applicability.

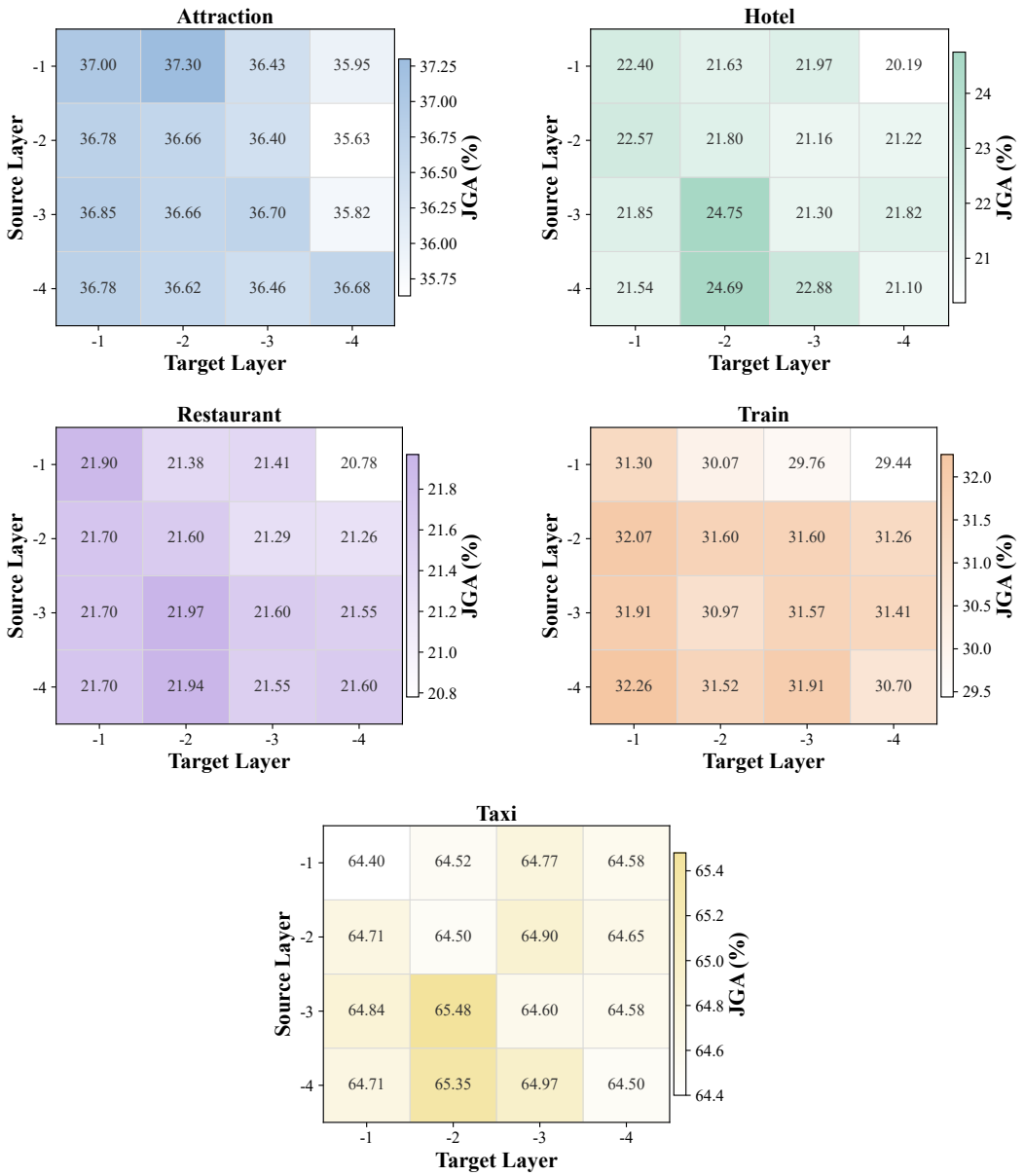


Figure 12: Effect of cross-layer construct–inject across five MultiWOZ domains. Overall, the method is robust to layer choice, with stronger performance typically observed near the diagonal.

- Dependence on source-domain slot information. Because StDST relies on slot information in the source domain to construct positive and negative prototypes, it cannot improve model performance when the source domain lacks such slot information.

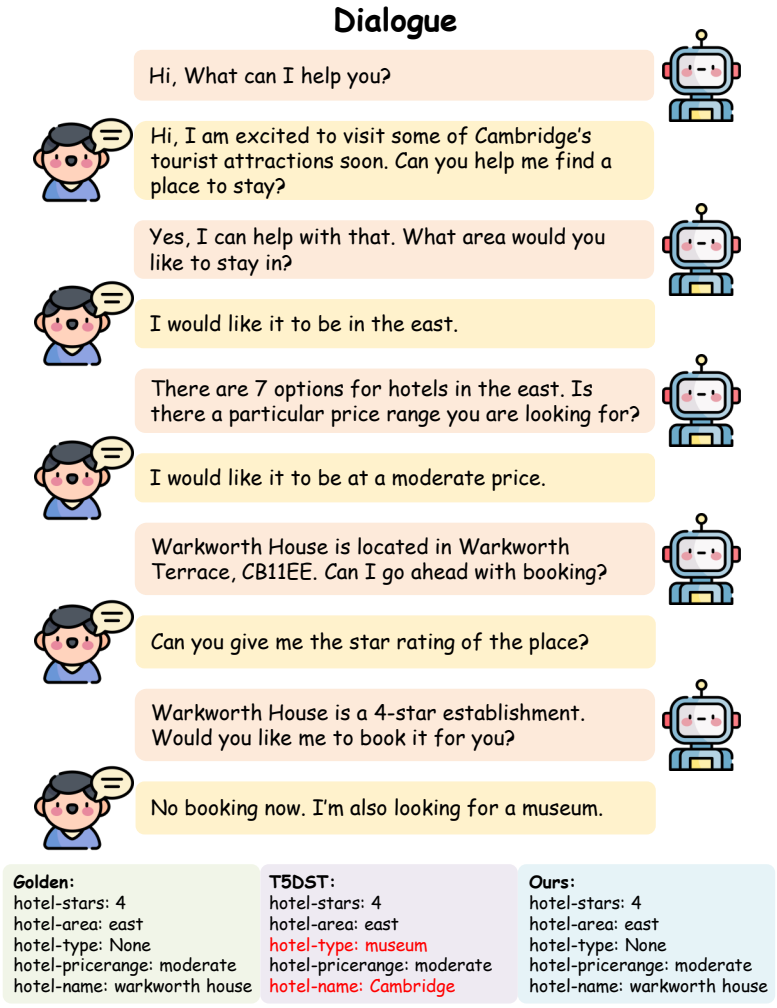


Figure 13: Zero-shot DST based on different models.