

Bridging Generation and Parsing: Collaborative Self-Distillation for Zero-shot Multi-Intent Spoken Language Understanding

Aziguli Wulamu¹, Ao He¹, Dezheng Zhang¹, Jun Li¹, Xinyuan Qian¹,
Zhihong Zhu², Libo Qin³ and Bowen Xing^{1*}

¹School of Computer and Communication Engineering,
University of Science and Technology Beijing

²Peking University

³Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen

Abstract

Multi-intent spoken language understanding remains challenging due to the scarcity of labeled data and the complexity of modeling intent-slot correlations. In this paper, we propose Collaborative Self-Distillation (Colla-SD), a novel framework that leverages a pair of open-weight large language models (LLMs) to synthesize high-quality labeled samples, which are used for fine-tuning the LLMs to achieve self-distillation. Our approach consists of six stages: (a) intent-slot sampling, (b) reverse progressive skeleton generation, (c) deduplication and quality filtering, (d) cross-model consensus flagging, (e) adjudication of disagreements, and (f) model training. On MixATIS, Colla-SD improves the Overall Accuracy of Llama-3.1-8B from 1.09% to 14.37%, corresponding to an approximately 13 $\times$ improvement over its training-free baseline. Across MixATIS and MixSNIPS, Colla-SD consistently improves the evaluated 7B/8B models over their training-free counterparts and remains competitive with much larger training-free LLMs on several metrics. Ablation studies, fine-grained human evaluation, and additional analyses further demonstrate the effectiveness of schema grounding and collaborative verification. Our work shows that Colla-SD¹ can effectively bridge LLM generation and zero-shot structured semantic parsing.

1 Introduction

Spoken language understanding (SLU) is a core component of task-oriented dialogue systems, typically involving intent detection and slot filling (Tur and De Mori, 2011; Young et al., 2013). While traditional SLU assumes a single intent per utterance, real-world user queries often express *multiple intents* simultaneously—e.g., “Play music and check the weather.” This multi-intent setting (Gangadharaiyah and Narayanaswamy, 2019) better reflects

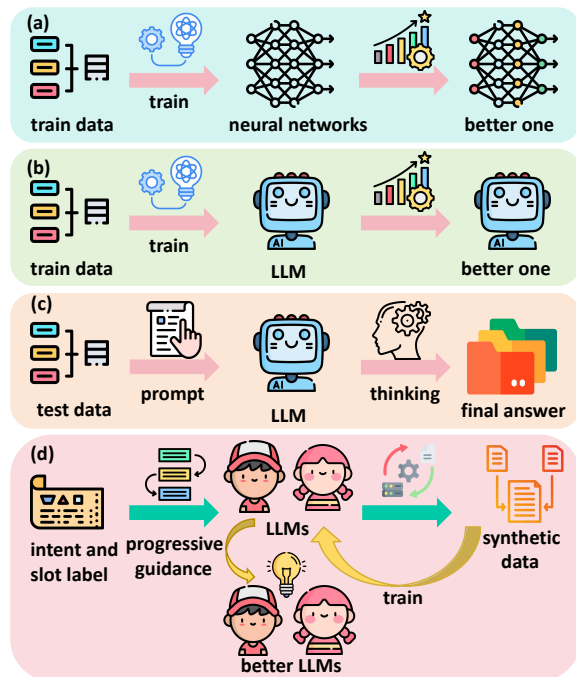


Figure 1: Illustration of different paradigms of multi-intent SLU. (a) Traditional neural networks trained on labeled data; (b) Supervised fine-tuning (SFT) of large language models (LLMs) on labeled data; (c) Zero-shot inference using LLMs; (d) Our proposed collaborative self-distillation framework that leverages LLMs to generate synthetic data and refine themselves.

practical usage but poses significant challenges: modeling complex intent-slot correlations requires rich compositional supervision.

Multi-intent SLU has long been approached with neural architectures that jointly model intent classification and slot filling, while learning their correlations (Qin et al., 2021; Xing and Tsang, 2022a; Xing et al., 2026; Xing and Tsang, 2024; Xia et al., 2025; Zhang et al., 2025). These methods, as shown in Fig. 1(a), require labeled training data to learn the alignment between user intents and structured slot-value outputs. SFT-based multi-intent SLU systems (Wu et al., 2022; Song

*Corresponding author, bwxing714@gmail.com

¹<https://github.com/myxruia/Colla-SD>.

et al., 2024; Xing et al., 2024), as shown in Fig. 1(b)—especially those handling multi-intent and complex slot structures—typically require extensive training on high-quality, human-annotated data that precisely aligns intents with slot-value spans. However, creating such datasets is labor-intensive, time-consuming, and expensive, as it demands expert annotators to consistently label both semantic intents and fine-grained slot boundaries across diverse utterances. Consequently, the scarcity of labeled data remains a major bottleneck in building effective and robust multi-intent SLU systems.

To address this data problem, recent work has explored powerful closed-source LLMs for zero-shot multi-intent SLU (Qin et al., 2025), as shown in Fig. 1(c). However, the complex domain-specific correlations between intents and slots remain challenging to model, even with powerful commercial LLMs. Open-weight LLMs offer an economical alternative, but their predictions often contain hallucination and misalignment, rendering them unsuitable for zero-shot multi-intent SLU. Furthermore, there is a data-level flaw in current methods: widely used benchmarks like MixATIS (Hemphill et al., 1990; Qin et al., 2020) and MixSNIPS (Coucke et al., 2018; Qin et al., 2020) are constructed by simply concatenating single-intent samples.

To address these limitations, we introduce Collaborative Self-Distillation (Colla-SD), as illustrated in Fig. 1(d). At a high level, Colla-SD transforms a task schema into reliable synthetic supervision through a *generate–verify–refine–train* loop. Rather than relying on a single LLM to generate and annotate its own training samples, the framework separates open-ended utterance generation from structured semantic verification and uses multiple LLMs to collaboratively improve the reliability of the resulting supervision.

Specifically, Colla-SD first samples intent and slot specifications from the task schema and grounds the sampled slots with concrete values. It then progressively constructs a semantic overview and generates a natural-language utterance under these constraints. After linguistic quality filtering, two LLMs independently parse each generated utterance. Samples with consistent predictions are retained directly, whereas disagreement cases are further examined through adjudication, which may correct or reconstruct the structured parse when necessary. The resulting synthetic corpus is finally used to fine-tune the downstream LLMs.

Compared to previous works relying on the high-

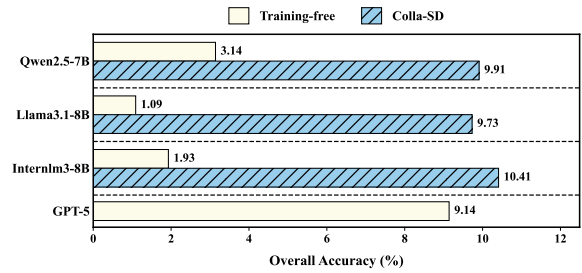


Figure 2: Overall accuracy comparison of 7B/8B-class LLMs on the MixSNIPS dataset. Models equipped with Colla-SD show up to a 10 times improvement compared to the training-free baselines.

quality manually labeled data, our approach breaks the dependency by enabling open-weight LLMs to collaboratively generate schema-consistent, high-fidelity training data without human annotation. By combining the multiple stages of reverse progressive skeleton generation, deduplication and quality filtering, cross-model consensus flagging, adjudication of disagreements, and model training, we create a positive loop that distills multi-intent SLU knowledge from multiple LLMs into themselves, obtaining a more robust and reliable SLU system. These creative steps ensure high semantic fidelity and effectively filter out the hallucination and misalignment typical of open-weight models.

We evaluate our approach on public multi-intent SLU benchmarks. As illustrated in Figure 2, the experimental results demonstrate that 7B/8B-class LLMs equipped with Colla-SD can achieve up to a 10 times performance improvement.

2 Related Work

Multi-Intent SLU Traditional SLU systems (Wu et al., 2020; Qin et al., 2019; Arora et al., 2022; Peng et al., 2022) assume a single intent per utterance, but real-world user queries often express multiple intents simultaneously. Therefore, most recent works focus on multi-intent SLU.

The dual-task interactions have been widely recognized and modeled by existing neural network-based models (Qin et al., 2021; Xing and Tsang, 2022a, 2023a,b; Zhu et al., 2024; Zhang et al., 2025). Rela-Net (Xing and Tsang, 2022b) and LCLR (Zhu et al., 2023) further leverage the dual-task correlation to constrain the decoding process. Some works leverage open-weight LLMs to conduct SFT on the labeled data by designing various question-answering-formatted prompt templates (Song et al., 2024; Wu et al., 2022;

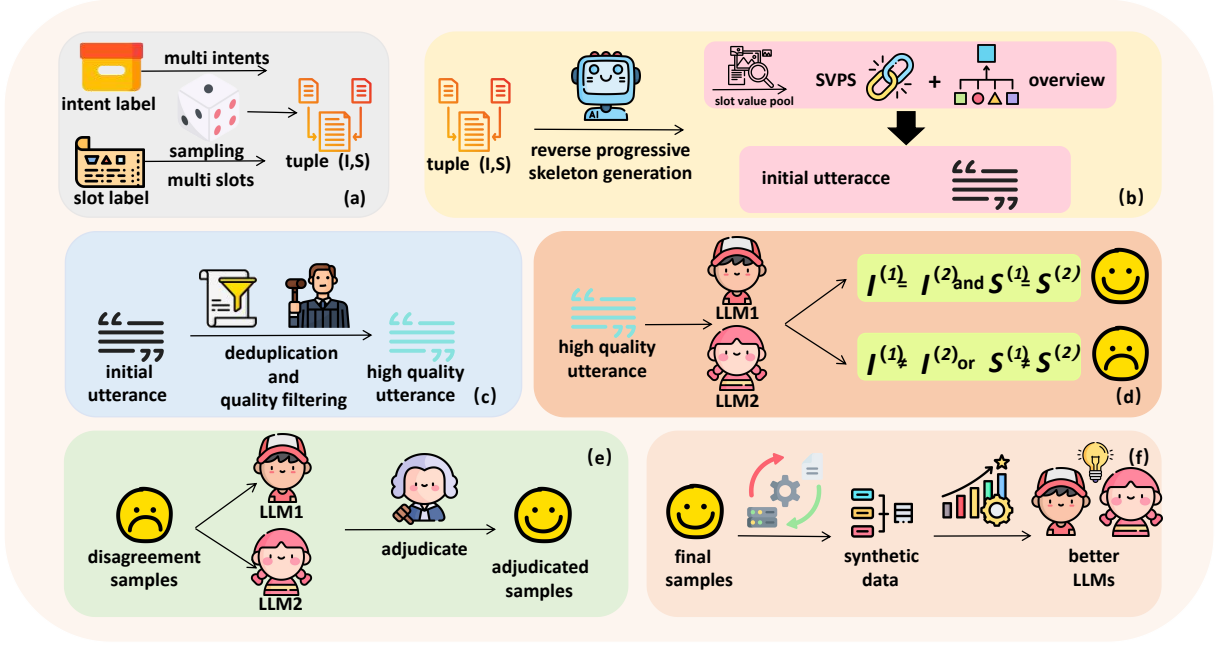


Figure 3: Illustration of our proposed Colla-SD framework, which consists of six key stages: (a) intent-slot sampling, (b) reverse progressive skeleton generation, (c) deduplication and quality filtering, (d) cross-model consensus flagging, (e) adjudication of disagreements, and (f) model training.

Xing et al., 2024; Yin et al., 2025). However, these methods remain heavily dependent on scarce human-annotated data, limiting their scalability to new domains or complex compositional structures. More recently, various advanced in-context learning strategies (Pan et al., 2023; Zhou et al., 2023; Wang et al., 2023; Kojima et al., 2022; Qin et al., 2025) are adopted to guide LLMs to perform few-shot multi-intent SLU, while the performance remains suboptimal. In this paper, we propose Colla-SD to leverage LLMs to synthesize high-quality data without human annotation and train downstream models on the resulting synthetic data.

Data Synthesis with LLMs Recent research has explored the use of LLM to create synthetic data for tasks or to augment existing datasets in order to improve task performance (Yu et al., 2023; Honovich et al., 2023; Shumailov et al., 2023; Patel et al., 2024). Some existing works have explored the impact of certain prompt design choices on model performance, such as the language of the prompt (Gonen et al., 2023) and few-shot prompt ordering (Lu et al., 2022). While prompting strategies improve reasoning, they typically depend on proprietary models and lack self-correction mechanisms. Recent work leverages LLMs for synthetic data generation via instruction tuning (Ding et al., 2023), such as syntheT2C (Zhong et al., 2025)

and SQuAD (Puri et al., 2020). Although quality metrics and filtering heuristics have been studied (Manakul et al., 2023), most approaches do not highlight rigorous consistency during synthesis. To address these bottlenecks, very recent studies have begun exploring advanced frameworks for high-quality data synthesis, such as unified LLM generation pipelines (Huang et al., 2025) and multi-agent collaborative simulations (Tang et al., 2025; Wang et al., 2025). In Colla-SD, paired LLMs play complementary roles during data synthesis and adjudication, allowing their respective strengths to contribute to the resulting synthetic supervision.

3 Preliminary

Multi-intent SLU is a generalization of single-intent SLU, where a user utterance may express multiple intents simultaneously. Formally, given an input utterance $x = (x_1, \dots, x_n)$ of n words, the intent label set $\mathcal{I}$ and the slot label set $\mathcal{S}$, the goal is to jointly predict: (1) a set of intents $\{i_1, \dots, i_k\}$, where $k \geq 1$ and $i_k \in \mathcal{I}$; (2) a sequence of slot value-label pairs $([v_1, l_1], \dots, [v_j, l_j])$, with $l_j \in \mathcal{S}$ indicating the slot label.

The challenge of zero-shot multi-intent SLU lies in accurately capturing intent-slot correlations and understanding the task-specific semantics with only the intent and slot label sets.

4 Methodology

As illustrated in Fig. 3, the overall pipeline of collaborative Self-Distillation (Colla-SD) consists of six key stages: (a) intent-slot sampling, (b) reverse progressive skeleton generation, (c) deduplication and quality filtering, (d) cross-model consensus flagging, (e) adjudication of disagreements, and (f) model training. The core idea of Colla-SD is to bridge open-ended utterance generation and structured semantic parsing through a collaborative multi-model refinement process. Instead of directly trusting a single LLM to both generate and annotate synthetic samples, we explicitly decompose the process into controllable generation, quality filtering, agreement-based verification, and disagreement adjudication. This design reduces hallucination, improves schema consistency, and makes the final synthetic corpus more suitable for downstream supervised fine-tuning.

In this section, we elaborate on the details of each stage. Below, we detail each component with formal definitions to capture the key mechanisms.

4.1 Intent-Slot Sampling

For each synthetic example, we sample a specification tuple (I, S) from the predefined task schema, where I contains 1–3 intent labels and S contains 1–4 slot labels, without accessing the test-set distribution. Since some sampled combinations may be less natural, only samples passing the subsequent generation, filtering, consensus, and adjudication stages are retained.

The tuple (I, S) specifies abstract semantic labels, while the domain-specific Slot Bank $\mathcal{B}$ provides candidate lexical values for each slot type. Each sampled slot is grounded with a value from its corresponding pool, enabling the specification to be realized as a natural utterance.

4.2 Reverse Progressive Skeleton Generation

We introduce reverse progressive skeleton generation to align intents, slot value-label pairs, and utterances. “Reverse” denotes generation from semantic labels to utterances, while “progressive” refers to step-wise generation from slot-value grounding to semantic overview and utterance realization. We further use **Multi-Value Sampling** (MVS) and a **Slot Bank** to improve data complexity and fidelity.

MVS allows a slot label $l \in S$ to take multiple values within one utterance, increasing structural complexity. The paired LLMs then independently

generate utterances in two steps.

Overview and Slot Value-Label Pairs Generation Given the intents I , slot labels S , and a predefined slot bank $\mathcal{B}$, the model selects or assigns plausible, domain-consistent values to each slot and produces a concise *utterance overview*. This step can be denoted as:

$$(\text{Overview}, \{[v_i, l_i]\}_{i=1}^k) = G_{\theta_a}(I, S, \mathcal{B}) \quad (1)$$

where $\mathcal{B}$ represents the available slot bank, G_{θ_a} denotes the generation process of LLM a with parameters θ_a , and $\{[v_i, l_i]\}_{i=1}^k$ denotes the determined Slot value-label pairs. We design the following template to prompt the LLM:

[Task Instruction.]

Your tasks are:

1. Assign a realistic slot value to each given slot label by selecting from the provided [Slot Bank] when available.
2. You may assign multiple different values to the same slot label to increase complexity.
3. Create a semantic DRAFT for synthesizing a user query that reflects all given intents and includes all slot value-label pairs.

[Schema & Slot Bank.]

Given intents

Given slot labels

Available Slot Bank: { "slot_label": ["value1", "value2", ...], ... }

[Format required.]

intents: ["intent1", "intent2", ...]

slot value-label pairs: [{"slot value 1", "slot label 1"}, ...]

utterance_overview: [your concise draft with all slot values]

Fluent Utterance Synthesis Taking the sampled intent, the generated overview and slot-value pairs as guidance, the LLM is prompted to generate a user utterance T that is conversational, grammatical, and strictly adheres to (I, S) :

$$T = G_{\theta_a}(I, \text{Overview}, \{[v_i, l_i]\}_{i=1}^k) \quad (2)$$

The prompt template is designed as:

[Task Instruction.]

Your task is to generate a fluent, natural-sounding ****user query**** (e.g., a question or request) based on the given draft and Slot value-label pairs.

[Overview]

Draft: "utterance_overview"

[Schema.]

Given intents

Given slot value-label pairs

[Format required.]

Output ONLY the utterance. No explanations. No extra

text.

4.3 Deduplication and Quality Filtering

To ensure data diversity and quality, we apply two steps on the raw generated samples:

Deduplication We use Sentence-BERT (Reimers and Gurevych, 2019) to encode each candidate utterance T into an embedding vector $\mathbf{e}_T$. For each newly generated candidate, we calculate its maximum cosine similarity against the set of samples already accepted into the final pool, denoted as $\mathcal{P}$. A candidate is retained only if it satisfies:

$$\max_{T' \in \mathcal{P}} \left(\frac{\mathbf{e}_T \cdot \mathbf{e}_{T'}}{\|\mathbf{e}_T\| \|\mathbf{e}_{T'}\|} \right) \leq \theta \quad (3)$$

where T' represents the sample already present in the pool; θ denotes the threshold.

Quality Filtering We utilize a larger judge model to score utterances based on four dimensions (D): fluency, naturalness, semantic richness, and logical coherence. Each dimension is scored on a scale from 1 to 5. The quality score for a sample T is defined as the average of the four scores:

$$\text{Score}(T) = \frac{1}{4} \sum_{d \in D} \text{score}_d(T) \quad (4)$$

Samples with scores at least 3.0 are retained.

In summary, only samples that successfully pass both the semantic deduplication and the quality filtering threshold are eligible to proceed to the next stage of our pipeline.

4.4 Cross-Model Consensus Flagging

After deduplication and quality filtering, we have to guarantee that the generated samples are assigned precise intent and slot annotations. However, this is challenging since existing LLMs hardly obtain promising zero-shot multi-intent SLU performance. In this case, we propose cross-model consensus flagging, inspired by multi-model debating (Liang et al., 2024). We utilize both LLMs to independently parse every generated utterance from Stage 2 into predicted intent set $\hat{I}$ and slot value-label pairs $\hat{S}$.

Let $(\hat{I}^{(1)}, \hat{S}^{(1)})$ and $(\hat{I}^{(2)}, \hat{S}^{(2)})$ denote the predictions from the two models. The sample is directly retained if: $\hat{I}^{(1)} = \hat{I}^{(2)}$ and $\hat{S}^{(1)} = \hat{S}^{(2)}$. Otherwise, it is flagged for adjudication.

4.5 Adjudication of Disagreements

For disagreement cases, we evenly distribute the samples between the LLMs and randomly designate one as the adjudicator. Each adjudicator follows the instructions below:

[Task Instruction.]

Your task is to produce the best possible semantic parse for the following user utterance.

[Additional Context.]

INTENT_DEFINITIONS_BLOCK
SLOT_DEFINITIONS_BLOCK

[Schema.]

User utterance:
Initial reference: parse1
Model A prediction: parse2
Model B prediction: parse3

[Format required.]

intents: ["intent1", ...]
slot value-label pairs: [{"slot value 1", "slot label 1"}, ...]

The adjudicator receives the following diverse information to give the final decision:

- The initial utterance T ,
- The initial specification (I, S) from Stage 1,
- The two conflicting parses $(\hat{I}^{(1)}, \hat{S}^{(1)})$ and $(\hat{I}^{(2)}, \hat{S}^{(2)})$,
- Full definitions of all valid intents and slots.

The adjudicator generates a parse $(\hat{I}^{\text{adj}}, \hat{S}^{\text{adj}})$ that best aligns with T . $(\hat{I}^{\text{adj}}, \hat{S}^{\text{adj}})$ is assigned as the final annotations of T .

This stage is crucial as it facilitates a complementary synergy between the paired LLMs. This mechanism allows the system to leverage the strengths of both models: if Model A provides a more accurate intent but Model B identifies a complex slot more precisely, the adjudicator can synthesize these "best parts" into a superior final parse. This collaborative refinement effectively mitigates individual model weaknesses and ensures that the final synthesized data is more robust and accurate than the output of any single model acting in isolation.

4.6 Model Training

The final training set for each model combination comprises consensus samples that satisfy the agreement criteria, and adjudicated samples from the adjudication stage. We first use the final synthetic corpus to derive multi-intent detection training set and slot filling training set. After that we perform SFT with LoRA (Hu et al., 2022) on the paired open-weight LLMs.

Methods		MixATIS			MixSNIPS		
		Intent Acc	Slot F1	Overall acc	Intent Acc	Slot F1	Overall acc
Training-free	Qwen2.5-7B (Hui et al., 2024)	32.85	37.80	3.14	74.62	29.44	1.86
	Llama3.1-8B	12.68	37.88	1.09	62.39	29.46	0.86
	Internlm3-8B	22.22	31.43	1.93	73.08	32.41	1.36
	Qwen2.5-32B (Hui et al., 2024)	43.36	69.18	9.54	86.54	45.70	4.55
	Qwen2.5-72B (Hui et al., 2024)	56.04	70.30	10.57	87.22	48.93	4.32
	Llama3.3-70B	12.80	67.49	2.17	85.45	41.75	3.14
	deepseek-r1 (671B) (Guo et al., 2025)	39.13	76.34	10.87	89.77	50.90	5.59
	deepseek-v3 (671B) (Liu et al., 2024)	38.53	74.51	7.49	83.86	50.82	6.05
	deepseek-v4 (284B)	43.12	78.49	10.75	92.09	46.93	3.87
	GPT5	74.51	79.58	27.54	92.91	63.75	9.14
	ChatGPT (gpt-3.5-turbo) (Xing et al., 2024)	22.10	34.50	1.90	67.50	30.00	1.40
	Qwen2.5-7B (Q+L data)	56.16	65.22	12.20	88.09	62.87	9.23
Colla-SD	Llama3.1-8B (Q+L data)	60.02	66.13	14.37	87.40	62.03	9.19
	Qwen2.5-7B (Q+I data)	61.96	59.46	12.92	83.49	63.44	9.91
	Internlm3-8B (Q+I data)	62.68	61.62	11.96	83.58	63.90	9.73
	Llama3.1-8B (L+I data)	48.07	59.85	9.78	85.04	63.69	9.73
	Internlm3-8B (L+I data)	47.95	61.14	10.27	85.31	62.26	10.41

Table 1: Main results on MixATIS and MixSNIPS.

Dataset	MixATIS	MixSNIPS
Intent labels.Num	18	7
Slot labels.Num	78	39
Test samples.Num	828	2199

Table 2: Statistics of the MixATIS and MixSNIPS datasets.

5 Experiments

5.1 Settings

Benchmarks and Evaluation Metrics We conduct experiments on two widely adopted multi-intent SLU benchmarks: MixATIS and MixSNIPS (Hemphill et al., 1990; Coucke et al., 2018; Qin et al., 2020). The evaluation is performed on the official test sets of MixATIS and MixSNIPS. Details of the statistics of MixATIS and MixSNIPS are shown in Table 2. Following the convention, we report three metrics:

- Intent Accuracy (Intent Acc): Proportion of the utterances with correctly predicted intents.
- Slot F1: Macro-averaged F1 score over all slots.
- Overall acc: Overall accuracy (sentence accuracy of joint intent and slot prediction).

Implementation Details We use model combinations consisting of three small models: Qwen2.5-7B, Llama3.1-8B, and InternLM-3-8B. Qwen-2.5-72B (int4-quant version) is adopted as the judge model for quality filtering. We finally obtain

10,000 and 13,000 synthesized samples for MixATIS and MixSNIPS, respectively. For SFT, we adopt LoRA with rank=8 and dropout=0.1.

5.2 Main Results

Performance comparisons are shown in Table 1.

Improvements over Training-Free Baselines. SFT on Colla-SD’s synthetic data yields multi-time improvements over training-free settings. Notably, Llama3.1-8B’s Overall Accuracy jumps from 1.09% to 14.37% on MixATIS and increases 11 times on MixSNIPS. Qwen2.5-7B and InternLM3-8B exhibit similar massive gains, demonstrating Colla-SD’s efficacy in handling multi-intent SLU.

Comparison with Larger Training-Free LLMs. Equipped with Colla-SD, our 7B-8B models significantly outperform massive LLMs (e.g., DeepSeek-R1, Qwen2.5-72B). Without explicit schema alignment, large LLMs struggle with slot boundary extraction in open-ended domains. On MixATIS, GPT-5 remains stronger across all three metrics, whereas on MixSNIPS the best Colla-SD configuration achieves 10.41% Overall Accuracy, compared with 9.14% for GPT-5. The remaining component metrics exhibit mixed trends. Therefore, we treat the large-model results primarily as training-free reference points rather than as evidence of uniform superiority over larger LLMs.

Adjudicator Model	MixATIS			MixSNIPS		
	Intent Acc	Slot F1	Overall Acc	Intent Acc	Slot F1	Overall Acc
Llama-3.1-8B	44.93	59.69	8.21	83.90	64.53	9.14
Qwen-2.5-7B	47.83	63.08	11.47	83.27	63.54	9.28
InternLM-3-8B	58.21	58.21	11.96	82.45	63.69	10.37

Table 3: Impact of different adjudicator models on the final performance. In this setting, Llama-3.1-8B is fixed as the base generation model across all experiments.

Models	MixATIS			MixSNIPS		
	Intent Acc	Slot F1	Overall Acc	Intent Acc	Slot F1	Overall Acc
Llama3.1-8B (w/Colla-SD)	60.02	66.13	14.37	87.40	62.03	9.19
Llama3.1-8B (self-generated)	43.60	23.39	2.54	76.81	27.13	1.64
Qwen2.5-7B (w/Colla-SD)	61.96	59.46	12.92	83.49	63.44	9.91
Qwen2.5-7B (self-generated)	52.17	32.48	4.71	80.67	27.25	1.09
Internlm3-8B (w/Colla-SD)	47.95	61.14	10.27	85.31	62.26	10.41
Internlm3-8B (self-generated)	47.83	24.11	4.71	79.40	26.67	1.14

Table 4: Ablation study results using only self-generated data.

Model-Agnostic Synergy. Consistent improvements across all tested LLM pairings (Q+L, Q+I, and L+I) highlight our framework’s model-agnostic nature. By leveraging the complementary parsing behaviors of different LLMs, Colla-SD stably synthesizes high-quality training data, proving its flexible applicability across various architectures without human annotation.

5.3 Impact of Adjudicator Types

We fix Llama3.1-8B as the generator and compare Llama, Qwen, and InternLM as adjudicators. As shown in Table 3, cross-model adjudication does not improve every component metric, but consistently improves Overall Accuracy over Llama self-adjudication on both datasets.

Different adjudicators exhibit complementary strengths: InternLM achieves the highest Intent and Overall Accuracy on MixATIS, whereas Qwen obtains the highest Slot F1. This suggests that cross-model adjudication mainly benefits joint intent-slot consistency rather than uniformly improving individual metrics. Additional analysis of adjudicator self-preference is provided in Appendix C.

5.4 Effect of the Collaborative Framework

To evaluate the necessity of our consensus mechanism, we fine-tune three models (Qwen, Llama, and Internlm) strictly on their raw, self-generated outputs. As Table 4 shows, removing Colla-SD leads to a severe performance degradation across all models and datasets.

Most notably, **Slot F1 collapses universally**: across all three models on both datasets, Slot F1 plummets from the 59%–66% range down to 23%–32%. **Intent accuracy** also suffers significantly, with Llama experiencing the steepest drop on MixATIS (from 60.02% to 43.60%). Consequently, the stringent **overall accuracy** plunges across the board; for instance, on MixSNIPS, it drops from ~10% to below 2% for all three models.

These findings confirm that raw LLM generations suffer from severe annotation noise. The Colla-SD framework effectively filters this noise through model synergy, providing strictly aligned and robust synthetic data for SFT.

5.5 Effect of MVS and Slot Bank

Table 5 isolates the effects of generative constraints on MixSNIPS, revealing that structural complexity (*Multi-Value*) and domain grounding (*Slot Bank*) are indispensable and highly synergistic.

First, removing Multi-Value Sampling (MVS) significantly degrades performance (e.g., Qwen2.5-7B’s Slot F1 drops from 62.87% to 50.13%). Without multi-value constraints, the model overfits to simplistic alignments and struggles with the dense semantics of real-world queries.

Second, removing Slot Bank breaks lexical grounding. Without a predefined value bank, the generator hallucinates schema-violating entities, injecting severe label noise into the training data.

Crucially, a counter-intuitive phenomenon highlights their synergy: removing both constraints

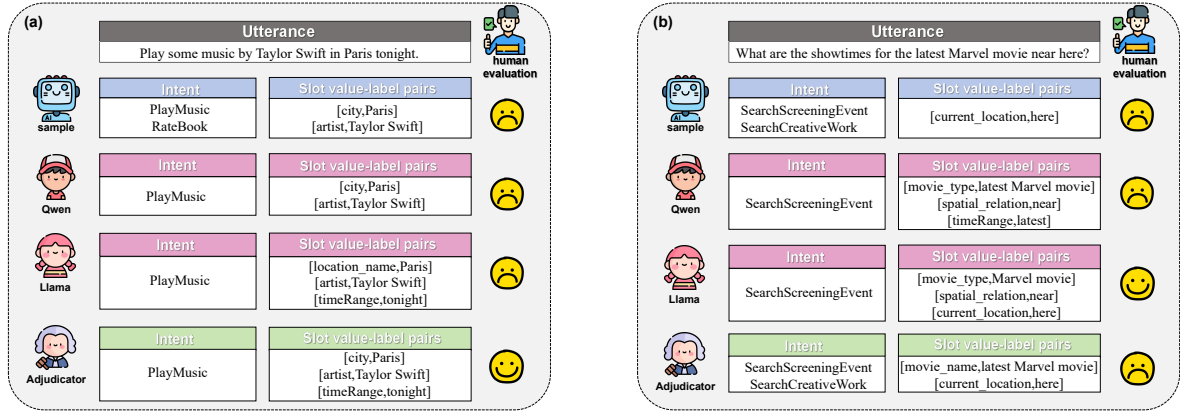


Figure 4: Comparison of human evaluation results.

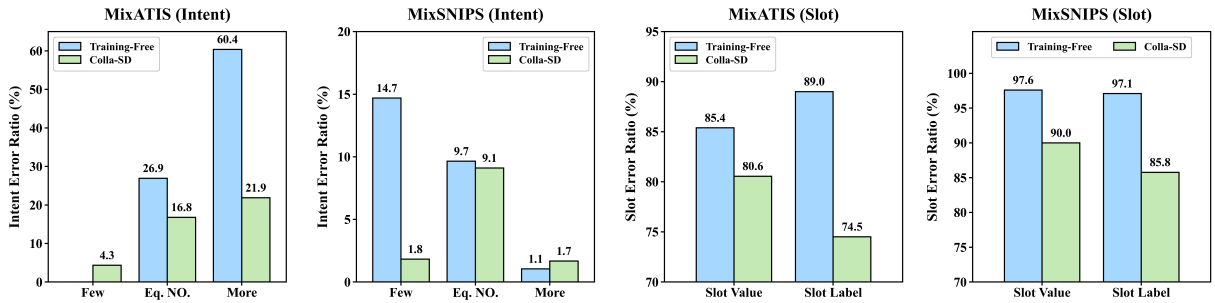


Figure 5: Comprehensive error analysis on MID and SF. Error rates across different error categories: *Few* (under-prediction), *Eq. No.* (correct count but wrong intents), and *More* (over-prediction) for multiple intent detection, alongside *Slot Value* and *Slot Label* errors for slot filling. Lower values indicate better performance.

MixSNIPS	Qwen2.5-7B		
	Intent Acc	Slot F1	Overall Acc
Colla-SD (Full)	88.09	62.87	9.23
w/o Slot Bank	85.72	41.15	3.41
w/o MVS	85.63	50.13	5.87
w/o Both	86.81	44.60	3.73

MixSNIPS	Llama3.1-8B		
	Intent Acc	Slot F1	Overall Acc
Colla-SD (Full)	87.40	62.03	9.19
w/o Slot Bank	84.49	40.73	4.09
w/o MVS	85.72	52.61	7.55
w/o Both	86.72	44.43	4.41

Table 5: Ablation study on the generation constraints (Slot Bank and MVS) using Qwen2.5-7B and Llama3.1-8B on MixSNIPS.

(w/o Both) actually outperforms removing only the Slot Bank (w/o Slot Bank). Forcing the LLM to generate complex, multi-value utterances without strict vocabulary boundaries compounds hallucinations, producing highly toxic data that destroys (I, S) alignment. In contrast, while w/o Both also lacks boundaries, its fallback to simpler

single-value structures limits the “blast radius” of hallucinations per sentence, reducing overall error accumulation.

5.6 Effect of Generator Scale

To examine whether data-construction and downstream models can be decoupled, we use INT4-quantized Qwen2.5-72B and Llama-3.1-70B to construct 2K synthetic samples and fine-tune Qwen2.5-7B and Llama-3.1-8B. The smaller corpus is used because 70B-scale inference incurs substantially higher computational cost.

As shown in Table 6, the resulting data effectively supervise smaller downstream models, with Overall Accuracy reaching 18.48% and 15.58% on MixATIS. These results support a teacher–student configuration, although larger generators do not consistently improve all metrics.

5.7 Reasoning Error Analysis

Fig. 5 presents a fine-grained error analysis comparing the training-free Llama baseline and our Colla-SD SFT model. Notably, on MixATIS, the baseline suffers a high More intent error rate of

Dataset	Qwen2.5-7B		
	Intent Acc	Slot F1	Overall Acc
MixSNIPS	86.45	52.89	5.87
MixATIS	64.86	63.57	18.48

Dataset	Llama3.1-8B		
	Intent Acc	Slot F1	Overall Acc
MixSNIPS	86.63	54.07	7.55
MixATIS	61.96	62.92	15.58

Table 6: Performance of 7B/8B downstream models trained on 2K synthetic samples constructed by INT4-quantized Qwen2.5-72B and Llama3.1-70B.

Evaluation Dimension	Avg. Score
Utterance Fluency	4.64
Semantic Naturalness	3.90
Intent Correctness	4.62
Slot-Label Correctness	3.78
Slot-Value Correctness	4.86
Full-Parse Correctness	3.65

Table 7: Fine-grained human evaluation on 100 randomly sampled synthetic examples. Each dimension is rated on a five-point scale, with utterance-level quality and parse-level correctness evaluated separately.

60.4%, likely because the large intent space induces hallucinations.

In contrast, Colla-SD substantially reduces primary errors across both datasets. On MixATIS, More intent errors drop to 21.9%, while slot label and value errors decrease from 89.0%/85.4% to 74.5%/80.6%. Similarly, MixSNIPS slot label and value errors fall significantly from 97.1%/97.6% to 85.8%/90.0%.

These reductions confirm that Colla-SD effectively distills task knowledge, mitigating hallucinations to yield more faithful predictions.

5.8 Human Evaluation

We conduct a fine-grained human evaluation on 100 randomly sampled synthetic examples using six five-point criteria covering utterance quality and parse correctness (Table 7).

The results show strong fluency, intent correctness, and slot-value correctness, while slot-label and full-parse correctness remain more challenging. This suggests that residual errors mainly arise from structured label assignment rather than surface realization. Under the stricter binary evaluation, 65% of the samples satisfy all correctness criteria.

6 Conclusion

In this paper, we propose Collaborative Self-Distillation (Colla-SD), a framework in which paired LLMs collaboratively synthesize and refine high-quality training data for multi-intent SLU without human annotation. The resulting synthetic data are then used to fine-tune downstream LLMs via SFT. Experimental results on public benchmarks show that Colla-SD consistently improves 7B/8B models over their training-free baselines, while comparisons with substantially larger LLMs remain dataset- and metric-dependent. Our work effectively bridges the gap between open-ended LLM generation and structured semantic parsing, offering a practical path toward annotation-free and scalable SLU systems.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62506033) and Fundamental Research Funds for the Central Universities (No. FRF-TP-25-035).

Limitations

While our framework significantly reduces hallucinations, a current limitation lies in our Slot Bank, which relies on manually collected and pre-defined sets of value choices. Ideally, the model should be capable of dynamically generating diverse and appropriate slot values conditioned directly on the given intent, rather than depending on human-curated pools. To address this, future work could explore integrating retrieval-augmented generation and knowledge graphs. These approaches would allow the framework to dynamically retrieve accurate, domain-specific, and up-to-date slot values from external knowledge sources, thereby eliminating the bottleneck of manual data collection and further enhancing the scalability of the system in zero-shot scenarios.

References

- Siddhant Arora, Siddharth Dalmia, Pavel Denisov, Xuankai Chang, Yushi Ueda, Yifan Peng, Yuekai Zhang, Sujay Kumar, Karthik Ganesan, Brian Yan, Ngoc Thang Vu, Alan W. Black, and Shinji Watanabe. 2022. [Espnet-slu: Advancing spoken language understanding through espnet](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 7167–7171. IEEE.

- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. [Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces](#). *ArXiv preprint*, abs/1805.10190.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. [Enhancing chat language models by scaling high-quality instructional conversations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.
- Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. [Joint multiple intent detection and slot labeling for goal-oriented dialog](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hila Gonen, Sridi Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. [Demystifying prompts in language models via perplexity estimation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shitong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. [Unnatural instructions: Tuning language models with \(almost\) no human labor](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.
- 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Yue Huang, Siyuan Wu, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Chaowei Xiao, Jianfeng Gao, Lichao Sun, and Xiangliang Zhang. 2025. [Datagen: Unified synthetic dataset generation via large language models](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17889–17904. Association for Computational Linguistics.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. [Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. [SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.
- Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. 2023. [A preliminary evaluation of chatgpt for zero-shot dialogue understanding](#). *ArXiv preprint*, abs/2304.04256.
- Ajay Patel, Colin Raffel, and Chris Callison-Burch. 2024. [Datadreamer: A tool for synthetic data generation and reproducible LLM workflows](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16,

- 2024, pages 3781–3799. Association for Computational Linguistics.
- Yifan Peng, Siddhant Arora, Yosuke Higuchi, Yushi Ueda, Sujay Kumar, Karthik Ganesan, Siddharth Dalmia, Xuankai Chang, and Shinji Watanabe. 2022. [A study on the integration of pre-trained ssl, asr, LM and SLU models for spoken language understanding](#). In *IEEE Spoken Language Technology Workshop, SLT 2022, Doha, Qatar, January 9-12, 2023*, pages 406–413. IEEE.
- Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. [Training question answering models from synthetic data](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.
- Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. [A stack-propagation framework with token-level intent detection for spoken language understanding](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.
- Libo Qin, Qiguang Chen, Jingxuan Zhou, Jin Wang, Hao Fei, Wanxiang Che, and Min Li. 2025. [Divide-solve-combine: An interpretable and accurate prompting framework for zero-shot multi-intent detection](#). In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 25038–25046. AAAI Press.
- Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. [GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling](#). 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)*, pages 178–188, Online. Association for Computational Linguistics.
- Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Iliia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. Anderson. 2023. [The curse of recursion: Training on generated data makes models forget](#). *ArXiv preprint, abs/2305.17493*.
- Feifan Song, Lianzhe Huang, and Houfeng Wang. 2024. [A unified framework for multi-intent spoken language understanding with prompting](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 9966–9970. IEEE.
- Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. 2025. [Synthesizing post-training data for llms through multi-agent simulation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 23306–23335. Association for Computational Linguistics.
- Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.
- Dong Wang, Yang Li, Ansong Ni, Ching-Feng Yeh, Youssef Emad, Xinjie Lei, Liam Robbins, Karthik Padthe, Hu Xu, Xian Li, Asli Celikyilmaz, Ramya Raghavendra, Lifei Huang, Carole-Jean Wu, and Shang-Wen Li. 2025. [Matrix: Peer-to-peer multi-agent synthetic data generation framework](#). *CoRR*, abs/2511.21686.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.
- Di Wu, Liang Ding, Fan Lu, and Jian Xie. 2020. [SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1932–1937, Online. Association for Computational Linguistics.
- Yangjun Wu, Han Wang, Dongxiang Zhang, Gang Chen, and Hao Zhang. 2022. [Incorporating instructional prompts into a unified generative framework for joint multiple intent detection and slot filling](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7203–7208, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Ying Xia, Zhen Xiong, Kefan Shen, Zhihong Zhu, Shaorong Xie, and Wei Liu. 2025. Rethinking decoding in multi-intent spoken language understanding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

- Bowen Xing, Lizi Liao, Minlie Huang, and Ivor W. Tsang. 2024. [Dc-instruct: An effective framework for generative multi-intent spoken language understanding](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 14520–14534. Association for Computational Linguistics.
- Bowen Xing, Libo Qin, Zhihong Zhu, Zhou Yu, and Ivor W. Tsang. 2026. [Dxa-net: Dual-task cross-lingual alignment network for zero-shot cross-lingual spoken language understanding](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1052–1062.
- Bowen Xing and Ivor Tsang. 2022a. [Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 159–169, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Bowen Xing and Ivor Tsang. 2022b. [Group is better than individual: Exploiting label topologies and label relations for joint multiple intent detection and slot filling](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3964–3975, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Bowen Xing and Ivor W Tsang. 2023a. Co-guiding for multi-intent spoken language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2965–2980.
- Bowen Xing and Ivor W Tsang. 2023b. Relational temporal graph reasoning for dual-task dialogue language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13170–13184.
- Bowen Xing and Ivor W Tsang. 2024. [HC²L: Hybrid and cooperative contrastive learning for cross-lingual spoken language understanding](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8094–8105.
- Shangjian Yin, Peijie Huang, JiaTian Chen, Haojing Huang, and Yuhong Xu. 2025. [ECLM: Entity level language model for spoken language understanding with chain of intent](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21851–21862, Vienna, Austria. Association for Computational Linguistics.
- Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. [Pomdp-based statistical spoken dialog systems: A review](#). *Proc. IEEE*, 101(5):1160–1179.
- Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. [Large language model as attributed training data generator: A tale of diversity and bias](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Shuli Zhang, Zhiqiang You, Xiaoxiang Qi, Peng Liu, Gaode Wu, Kan Xia, and Shenguang Huang. 2025. Lcan: A label-aware contrastive attention network for multi-intent recognition and slot filling in task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25603–25612.
- Zijie Zhong, Linqing Zhong, Zhaoze Sun, Qingyun Jin, Zengchang Qin, and Xiaofan Zhang. 2025. [Synthet2c: Generating synthetic data for fine-tuning large language models on the text2cypher task](#). In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 672–692. Association for Computational Linguistics.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. [Least-to-most prompting enables complex reasoning in large language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Dongsheng Chen, and Yuexian Zou. 2023. [Towards unified spoken language understanding decoding via label-aware compact linguistics representations](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12523–12531, Toronto, Canada. Association for Computational Linguistics.
- Zhihong Zhu, Xuxin Cheng, Hongxiang Li, Yaowei Li, and Yuexian Zou. 2024. Dance with labels: Dual-heterogeneous label graph interaction for multi-intent spoken language understanding. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1022–1031.

A Statistics and Domain-Specific Adaptation of the Synthesized Corpus

Statistic (Per Model Pair)	MixATIS	MixSNIPS
Standard Utterances	10,000	10,000
Multi-Value Utterances	0	3,000
Total Volume	10,000	13,000

Table 8: The composition of the synthesized training data for each model pair.

Table 8 details the quantitative characteristics of the final synthesized corpora generated for each model combination across the two datasets.

As shown in Table 8, we introduce an additional 3,000 multi-value instances specifically for MixSNIPS, while keeping the MixATIS corpus strictly to 10,000 standard instances. This discrepancy is a deliberate design choice driven by the inherent linguistic characteristics of the respective domains.

MixSNIPS simulates a broad-domain personal assistant (e.g., managing playlists, booking restaurants), where users frequently mention multiple distinct values for the same slot type in a single breath (e.g., “Add song A and song B to my playlist”). Consequently, injecting multi-value instances significantly enhances the model’s structural robustness.

Conversely, MixATIS is restricted to the highly constrained airline travel domain. User queries in this scenario are predominantly linear and precise. Fabricating complex multi-value queries (e.g., simultaneously booking flights to three unrelated destinations in one sentence) fundamentally contradicts the natural distribution of the ATIS domain and its underlying schema. Forcing the LLM to generate such utterances would introduce unnatural, out-of-distribution (OOD) noise that misaligns with the target test set. Therefore, we omit multi-value generation for MixATIS to preserve strict semantic fidelity to the aviation domain.

B Detailed Prompt Templates

In this section, we provide the complete prompt templates used across the four stages of the **CollaSD** pipeline.

B.1 Reverse Progressive Generation

The synthetic data generation is decoupled into two steps: Draft Generation (Table 9) and Utterance Synthesis (Table 10).

B.2 LLM-as-a-Judge for Quality Filtering

Table 11 displays the scoring criteria used to evaluate the linguistic quality of the synthesized utterances.

B.3 Cross-Model Parsing and Adjudication

Table 12 and Table 13 illustrate the instructions for individual model parsing and final conflict resolution.

B.4 Training-free Baseline Evaluation

To evaluate the models without any task-specific fine-tuning, we utilize a zero-shot training-free setup. Since intent classification and slot filling

are decoupled in our baseline evaluation, we employ two separate prompts for these tasks. Table 14 and Table 15 detail the exact instructions provided to the models.

C Analysis of Adjudicator Self-Preference

To further examine whether an adjudicator tends to favor its own prior prediction, we analyze 13,982 disagreement cases from the original data-construction pipeline. For each case, we categorize the final adjudicated parse according to whether it exactly matches the adjudicator’s own prediction, the other model’s prediction, the initial generation reference, or none of these candidates.

The results reveal model-dependent self-preference. Qwen matches its own previous prediction in 27.42% of disagreement cases, compared with 10.78% for the other model’s prediction, corresponding to a gap of 16.64 percentage points. In contrast, Llama exhibits a substantially smaller gap between its own prediction (17.65%) and the other model’s prediction (15.18%).

Nevertheless, newly synthesized parses constitute the majority for both adjudicators, accounting for 59.57% of Qwen cases and 63.99% of Llama cases. This indicates that adjudication usually involves semantic verification and reconstruction rather than simple selection among the existing candidate parses. At the same time, the results suggest that self-preference is not entirely absent and can vary across adjudicator models.

Semantic Draft Generation

System/User:

You are an expert in Aviation domain.

Your tasks are:

1. Create a semantic DRAFT for synthesizing a natural user utterance that reflects all given intents and includes all assigned slot value-label pairs.

Given intents and descriptions: {intent_desc_block}

Given slot value-label pairs and slot descriptions:

{pairs_desc}

{slot_desc_block}

Generation Rules:

- The draft must include each slot value exactly and directly as given.
- The draft must clearly reflect **all given intents** through fluent, conversational phrasing.
- The draft should be as simple as possible.
- The draft can be a phrase, question, or command fragment — but **MUST** contain all provided slot value-label pairs and convey all the given intents.
- Do **not introduce** any new intent, slot or slot value beyond those provided.
- Output **ONLY** in the exact format below. No explanations. No extra text.

Required output format:

intents: ["intent1", "intent2", ...]

slot value-label pairs: [{"slot label 1", "slot value 1"}, {"slot label 2", "slot value 2"},...]

utterance_overview: [your concise natural-language draft]

Table 9: The prompt template used for generating the semantic draft and slot value assignments.

Fluent Utterance Synthesis

System/User:

You are an expert in Aviation domain.

Your task is to generate a fluent, natural-sounding **user query** (e.g., a question, request, or command) based on the given draft and slot value-label pairs.

Draft (for phrasing inspiration only): "{utterance_overview}"

Given intents and descriptions: {intent_desc_block}

Given slot value-label pairs and slot descriptions:

{pairs_desc}

{slot_desc_block}

Rules:

1. You must include each slot value exactly as given—no paraphrasing, no omission.
2. **DO NOT** introduce any new intents or slot value-label pairs beyond those provided.
3. Use the draft only for tone and structure—not as a complete guide.
4. Make the utterance sound realistic and conversational, like something a real user would say to a smart speaker.
5. The output must be a **realistic user query**: typically a **question**, **command**, or **statement**.
6. Do NOT use: "Can you...", "Could you...", "I need...", "I am looking for..."
7. **Ensure the utterance is clear and concise**: Avoid redundant phrases.

Output **ONLY** the utterance. No explanations. No extra text.

Table 10: The prompt template used for synthesizing the final natural language utterance.

Utterance Quality Scoring

System/User:

You are an expert evaluator for spoken language understanding (SLU) data in the aviation domain.

Please score the following user utterance on four dimensions from 1 to 5, based **solely on the quality of the natural language**, not on how well it matches predefined slots or intents.

Scoring Criteria:

1. **Fluency**: Is the utterance grammatically correct, smoothly phrased, and free of awkward or unnatural constructions?
2. **Naturalness**: Does it sound like something a real passenger would actually say in a conversational or query context?
3. **Semantic Richness**: Does the utterance convey meaningful, specific, and non-trivial information (e.g., includes concrete details, avoids vagueness)?
4. **Logical Coherence**: Is the content internally consistent and free of contradictions or implausible combinations?

Input:

- Utterance: "{utterance}"

Output exactly in this format (no extra text, no JSON braces, no code blocks):

"fluency": X, "naturalness": Y, "semantic_richness": Z, "logical_coherence": W

- Example: "fluency": 4, "naturalness": 5, "semantic_richness": 4, "logical_coherence": 5

Table 11: The prompt template used by the Judge model to filter out low-quality queries.

Cross-Model Consensus Parsing

System/User:

You are an expert in aviation-domain spoken language understanding. Your task is to analyze the user utterance and output the expressed intents and the included slot value-label pairs:

1. **Intents**: Choose ONLY from the following labeled intents:

{intent_lines}

2. **Slots**: Extract explicit slot-value pairs where slot labels are ONLY from the following list:

{slot_lines}

Rules:

- Only include intents/slots that are clearly expressed in the utterance.
- Slot values must be **exact substrings** from the utterance (case-sensitive, no normalization).
- Do NOT infer or guess missing information.
- **At most 3 intents** may be selected. If more than 3 are expressed, choose the 3 most prominent.
- **At most 4 slot-value pairs** may be extracted. If more than 4 are present, choose the 4 most relevant.

Please Generate strictly according to the following format. No explanations. No extra text.

format:

intents: ["intent1", "intent2", ...]

slot value-label pairs: [{"slot label 1", "slot value 1"}, {"slot label 2", "slot value 2"},...]

Utterance: "{utterance}"

Table 12: The prompt template used for independent SLU parsing by Model A and Model B.

Adjudication and Conflict Resolution

System/User:

You are an expert in aviation-domain spoken language understanding.

Your task is to produce the best possible semantic parse for the following user utterance.

Constraints:

- You MUST ONLY use intents from this predefined set: [{VALID_INTENTS}]
- You MUST ONLY use slot labels from this predefined set: [{valid_slots_str}]
- Slot values MUST be exact, case-sensitive substrings of the utterance.
- Do NOT invent new intents or slots outside the sets above.
- If none of the candidate parses are fully correct, synthesize a new result using only allowed intents and slots that appear in the utterance.

{INTENT_DEFINITIONS_BLOCK}

User utterance:

"{utterance}"

Candidate parses (may contain errors or illegal labels):

- Initial generation reference: {gold_parse}
- Model A prediction: {pred_a}
- Model B prediction: {pred_b}

Instructions:

1. Carefully check each candidate against the constraints.
2. If all candidates violate the intent/slot vocabulary or extract wrong values, IGNORE them and generate a correct parse from scratch.
3. Only include intents/slots that are explicitly supported by words in the utterance.
4. **At most 3 intents** may be selected. If more than 3 are expressed, choose the 3 most prominent.
5. **At most 4 slot-value pairs** may be extracted. If more than 4 are present, choose the 4 most relevant.

Output format (STRICTLY follow this syntax, no explanations, no extra text):

intents: ["intent1", "intent2", ...]

slot value-label pairs: [{"slot label 1", "slot value 1"}, {"slot label 2", "slot value 2"},...]

Table 13: The prompt template used by the Adjudicator model to resolve cross-model conflicts.

Training-free Intent Classification

System/User:

You are the most precise intent classification expert. Your task is to extract up to 3 intents from the user utterance.

Available intent labels (you MUST choose ONLY from these, never invent new ones):

[{INTENT_LABELS}]

Input utterance: "{utterance}"

CRITICAL ZERO-SHOT RULES: Do not output anything except the annotations. No explanations. No extra text.

Please Generate strictly according to the following format:

```
{
  "intents": ["intent1", "intent2"]
}
```

Table 14: The prompt template used for evaluating the training-free baseline on intent classification.

Training-free Slot Filling

System/User:

You are the most precise slot-filling expert. Your task is to extract every slot-value pair explicitly mentioned in the user utterance with zero tolerance for hallucination.

Available slot labels (you MUST choose ONLY from these, never invent new ones):

[{SLOT_LABELS}]

CRITICAL ZERO-SHOT RULES (follow strictly, no examples allowed):

- Do NOT output anything except the annotations.
- Slot values must be exact substrings copied from the input sentence.

Output format:

```
{  
  "slots": [{"slot_label", "exact_value"}, {"slot_label2", "exact_value2"}, ...]  
}
```

No explanation and extra text.

Input utterance: "{utterance}"

Table 15: The prompt template used for evaluating the training-free baseline on slot filling.

Adjudicator	Own	Other	Initial	New
Qwen	27.42	10.78	2.23	59.57
Llama	17.65	15.18	3.18	63.99

Table 16: Distribution (%) of adjudicated outputs over 13,982 disagreement cases. “Own” and “Other” denote exact matches with the adjudicator’s own previous prediction and the other model’s prediction, respectively. “Initial” denotes matches only with the initial generation reference, while “New” denotes a final parse different from all three candidates.