PRoDeliberation: Parallel Robust Deliberation for End-to-End Spoken Language Understanding

Trang Le[*]   **Daniel Lazar**[* 1†]   **Suyoun Kim**   **Shan Jiang**   **Duc Le**
Adithya Sagar   **Aleksandr Livshits**   **Ahmed Aly**   **Akshat Shrivastava**[†]
Meta, Coldrays[1]
trangleminh@meta.com, daniel@coldrays.com, akshatsh77@gmail.com

Abstract

Spoken Language Understanding (SLU) is a critical component of voice assistants; it consists of converting speech to semantic parses for task execution. Previous works have explored end-to-end models to improve the quality and robustness of SLU models with Deliberation, however these models have remained autoregressive, resulting in higher latencies. In this work we introduce PRoDeliberation, a novel method leveraging a Connectionist Temporal Classification-based decoding strategy as well as a denoising objective to train robust non-autoregressive deliberation models. We show that PRoDeliberation achieves the latency reduction of parallel decoding (2-10x improvement over autoregressive models) while retaining the ability to correct Automatic Speech Recognition (ASR) mistranscriptions of autoregressive deliberation systems. We further show that the design of the denoising training allows PRoDeliberation to overcome the limitations of small ASR devices, and we provide analysis on the necessity of each component of the system.

1 Introduction

Traditionally, Spoken Language Understanding (SLU) systems are pipelined systems of 1) an Automatic Speech Recognition (ASR) model, which transcribes audio to text, and 2) a semantic parsing model to convert text into structured frames. This results in some inherent limitations, such as cascading errors from ASR mistranscriptions, larger overall systems size from having two models with redundant capabilities, and the inability to use audio cues for semantic parse generation.

End-to-end (E2E) models hold promise of overcoming these limitations by removing the barrier between these two models and allowing audio information to be used in the semantic parse generation.

In particular, *deliberation*-based E2E SLU models (Le et al., 2022; Hu et al., 2020) improve explainability (transcript is produced by first-pass ASR), support of complex compositional structure (via the general decoder), and quality (allows rewriting the ASR transcriptions based on audio information).

However, one downside of the existing deliberation-based E2E SLU models is that they use an autoregressive (AR) decoder for generation, which yields high latency, especially for longer outputs. This precludes them from being used in the on-device setting, which requires low latency. Previous semantic parsing models can yield very low latency with non-autoregressive (NAR) decoders (Shrivastava et al., 2023, 2021; Babu et al., 2021; Zhu et al., 2020), however, these systems use extractive decoders and are therefore unable to correct mistranscriptions from ASR systems.

Our study targets improving Deliberation based SLU systems to adapt them for on-device streaming use cases. In this work we present PRoDeliberation (**P**arallel **Ro**bust **Deliberation**), a novel Connectionist Temporal Classification (CTC) decoder-based E2E SLU model. CTC decoders are commonly used for ASR (Graves et al., 2006) and have successfully been adapted for machine translation (Libovický and Helcl, 2018; Saharia et al., 2020), but to the best of our knowledge, have never been used for semantic parsing. To further improve the rewrite capabilities of this model we introduce a *denoising* training regimen inspired by masked language modeling (Devlin et al., 2019; Lewis et al., 2020) in which we corrupt the transcription text, requiring the model to correct the corruptions by relying on audio. The denoising training applies beyond SLU and can be used with any downstream task which consumes ASR transcriptions.

We study PRoDeliberation and find that PRoDeliberation retains the quality of autoregressive parsers while achieving a **2-10x** latency reduction across different ASR model sizes. We compare
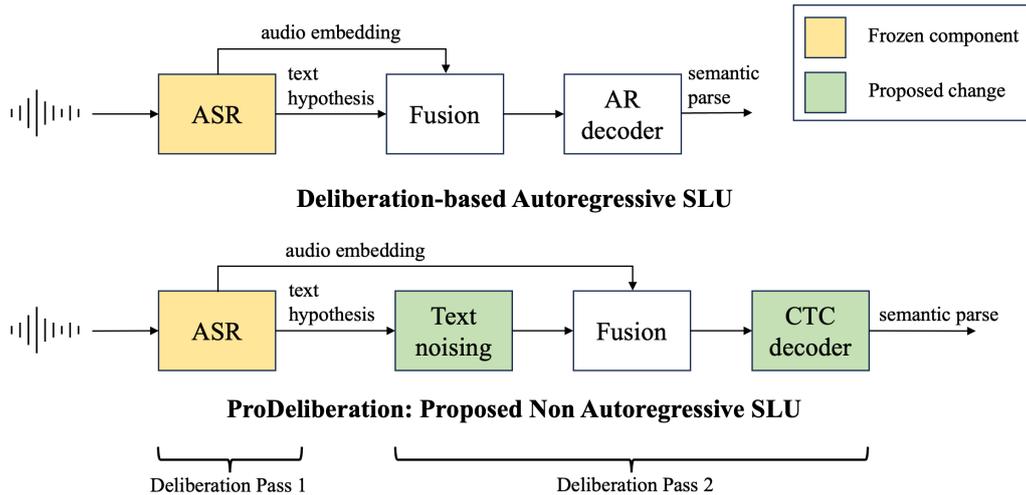
Figure 1: Overview of the proposed method, compared to the previous state of the art for end-to-end Spoken Language Understanding. The proposed method adds two crucial components: a CTC non-autoregressive decoder and a text noising mechanism for training. We label the two passes for the Deliberation architecture (both methods).

ProDeliberation to Mask Predict-based approaches (Ghazvininejad et al., 2019; Babu et al., 2021) and confirm that PRoDeliberation overcomes the quality shortcomings of Mask Predict. Finally, we show that the denoising objective further boosts ASR robustness by approximately 0.3%, thereby exceeding the quality of autoregressive models.

We provide a summary of our approach in Figure 1; concretely, our contributions are:

1. We propose the first non-autoregressive deliberation model, **PRoDeliberation**. Our model leverages CTC decoding and a denoising objective; both of which have not been applied to Spoken Language Understanding before.

2. Our model shows significant latency improvement while preserving quality of autoregressive SLU models, which is a challenging but important task.

3. We provide ablations for PRoDeliberation to understand the impact of each component.

2   Related work

2.1   Spoken Language Understanding

Datasets   Semantic parsing is a critical piece of traditional voice assistants, however text-based semantic parsing benchmarks assume gold text, disregarding errors propagating from speech understanding in the full SLU pipeline. Recently new SLU benchmarks have been introduced such as SLURP (Bastianelli et al., 2020), a multidomain SLU benchmark, and STOP (Tomasello et al., 2023),

a multidomain and compositional SLU benchmark. In this work we study the STOP dataset as it is the largest dataset, includes more speakers, and contains compositionality, making it the most diverse and complex dataset.

Models   Tomasello et al. (2023) propose leveraging fully E2E models pre-trained on the ASR task and fine-tuned on the parsing task to produce a series of models based on HuBERT (Hsu et al., 2021) and Wav2Vec 2.0 (Baevski et al., 2020). However such E2E systems remain too large for on-device processing and lack the modularity and interpretablity often required to deploy SLU systems. FANS (Radfar et al., 2021) explore on-device SLU systems leveraging SLU specific decoders and Kashiwagi et al. (2023) explore distilling E2E models, but both lack interpretability. Neural interfaces (Rao et al., 2020) have been used to retain the modularity of transcription in E2E models, and the Deliberation SLU system (Le et al., 2022) further the neural interfaces into the on-device space. In this work we study further optimizing efficiency and robustness of deliberation SLU systems with non-autoregressive decoding and denoising.

Robustness   Recently Kim et al. (2023) have explored improving the robustness of Deliberation-based SLU by incorporating confidence signals from ASR models and shown promising results (+0.1%-0.4%). Sharma et al. (2023) augmented Deliberation SLU robustness with extra data through generating text data with LLMs and generating audio data with Joint-Audio-Text and Text-to-Speech

techniques. In this work we take a complimentary direction to improve robustness through denoising training. We focus on denoising as we can scale denoising training to directly influence the difficulty of the robustness training. Our method can be combined with Kim et al. (2023) and Sharma et al. (2023)'s strategies to achieve a highly robust deliberation system.

2.2 Non-autoregressive Parsing

The semantic parsing community has shifted towards the seq2seq paradigm (Sutskever et al., 2014) to improve quality and align with the larger NLP community (Lewis et al., 2020; Raffel et al., 2020). With this shift, semantic parsers require low latency due to their real time nature. Non-autoregressive semantic parsing has taken off due to its sub-linear decoding time leading to strong latency improvements. Prior work in this space (Babu et al., 2021) has shown that non-autoregressive algorithms can lead to lower latency systems at similar accuracy gains. To further improve latency and quality, prior works aim to build an inductive bias regarding semantic parsing systems: Zhu et al. (2020) rely on insertion transformers to build a semantic parse around the text, Shrivastava et al. (2021) rely on span based prediction to extract slot text, and Shrivastava et al. (2023) retrieve defined semantic frames and fill in the slots with span prediction.

Our work continues in this direction, critically extending efficient parsing into end-to-end SLU systems which do not take for granted the truth of the ASR transcriptions but rather operate on both audio input and text hypotheses. The shift away from a gold text input space introduces nuances around robustness and text generation that prior work cannot support due to the focus on extractive decoders. We introduce a CTC loss function (Graves et al., 2006) for non-autoregressive decoding to reduce the reliance on length prediction, a key bottleneck even in non-end-to-end systems, but even more so for end-to-end SLU.

3 PRoDeliberation

In this work we extend Deliberation with two critical components: First we introduce **CTC decoding**, which utilizes parallel decoding to optimize latency. However, we notice an increased error rate due to parallel decoding. In order to combat this we add the second key component: **denoising**, which improve model robustness during training.

Figure 2 shows the architecture of PRoDeliberation in comparison to the architecture of Autoregressive Deliberation.

3.1 Deliberation

Le et al. (2022) introduced the deliberation-based E2E SLU system, shown in the upper part of Figure 1. Deliberation was introduced in Hu et al. (2020) as a two-pass ASR system to improve transcription quality. We describe these two passes in detail in the following.

First Pass: Speech Recognition with RNN-T The first pass of the system uses the recurrent neural network transducer (RNN-T) (Graves, 2012), which is the most popular streaming ASR architecture. RNN-T is built on three core components: (1) **Acoustic encoder** which consumes the raw audio and produces an audio encoding $emb_{\text{aud}} \in \mathbb{R}^{A \times D}$ (2) **Predictor** (internal language model) that consumes the text predicted so far and produces a language encoding $emb_{\text{text}} \in \mathbb{R}^{T \times D}$ and (3) **Joiner** which combines these two embeddings emb_{text} and emb_{aud} to produce a probability output for the currently predicted token. Each component of RNN-T is streamable, allowing the full model to decode text as the user speaks.

Second Pass: Deliberation Decoder The deliberation decoder conducts a second pass over the RNN-T hidden states using the language embedding emb_{text} and the audio embedding emb_{aud} that have been cached during the first pass. The deliberation decoder fuses the embedding:

$$emb_{\text{attn}} = \text{MHA}(emb_{\text{text}}, emb_{\text{aud}}, emb_{\text{aud}})$$
$$emb_{\text{stack}} = \text{Stack}(emb_{\text{text}}, emb_{\text{attn}})$$
$$emb_{\text{fused}} = \text{Linear}(emb_{\text{stack}})$$
$$emb_{\text{pool}} = \text{Transformer}(emb_{\text{fused}}),$$

where MHA indicates multi-headed attention. The fused embeddings are then passed to a series of transformer encoder layers to pool the embedding, denoted emb_{pool}. Finally the pooled layers are passed to an autoregressive pointer generator decoder which produces tokens which are either 'pointers' to input tokens or newly 'generated' tokens (Aghajanyan et al., 2020; See et al., 2017).

3.2 Nonautoregressive Decoding with CTC

The latency of autoregressive decoders suffers as sequence length grows due to the linear relationship between latency and sequence length, a result
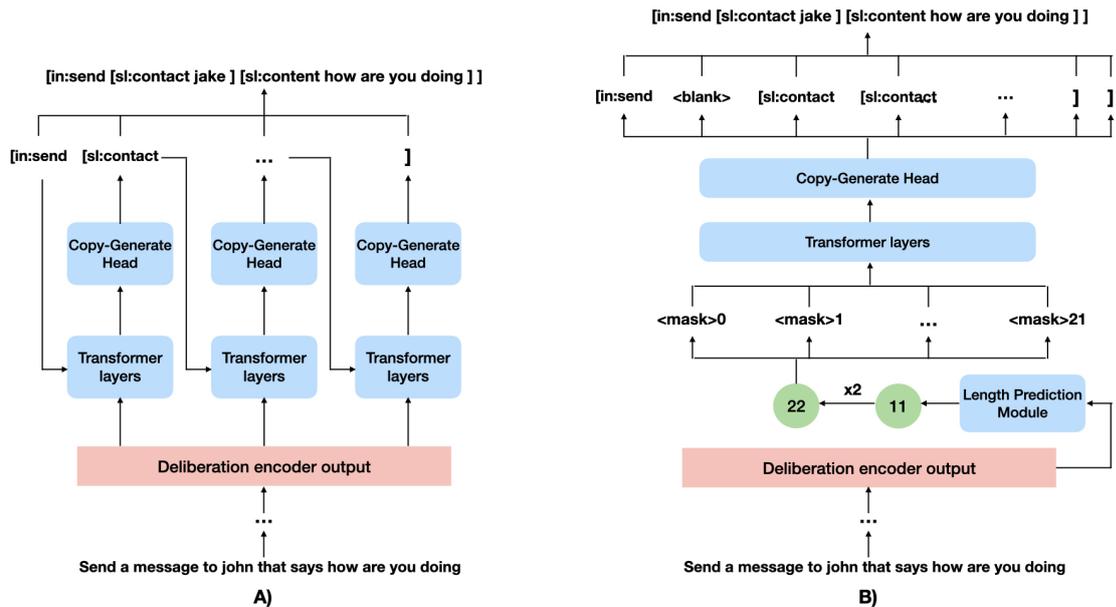
Figure 2: A comparison between (A) Autoregressive Deliberation Decoders and (B) PRoDeliberation architecture. PRoDeliberation extends Deliberation (A) with a fuzzy length prediction module and a CTC based decoder. Both (A) and (B) can correct ASR errors ('john' to 'jake' in the figure).

of the conditional dependence of each token output on the preceding tokens. To alleviate this, and following text-based semantic parsing work, we extend deliberation to leverage non-autoregressive decoders (Zhu et al., 2020; Babu et al., 2021; Shrivastava et al., 2021). However we find that prior work relies on Mask Predict (Ghazvininejad et al., 2019) which introduces a bottleneck in the form of length prediction of the output sequence. In this section we introduce our CTC-based non-autoregressive decoder which overcomes this limitation of length prediction. Our decoder is composed of 2 steps: (1) Length Prediction and (2) Parallel Decoding.

Length Prediction A critical component of non-autoregressive networks is length prediction for output sequence (l). Prior work implements a length prediction module to determine the output length (Babu et al., 2021; Ghazvininejad et al., 2019).

In the extractive setting such as pipelined SLU, where the semantic parse contains either 1) ontology tokens (intent and slot labels) or 2) spans of transcribed text, this does not pose a problem (Shrivastava et al., 2021). However, a major advantage of E2E SLU is the ability to correct ASR errors. This changes the simple extractive function of the decoder into a more complex one, one which may change the length of a span of tokens in a slot from the length of the span predicted by ASR. Because of this, we propose using a CTC decoder which allows for more flexibility in the decoding process

than Mask Predict.

Instead of an equality constraint between the predicted length and the final output length, our use of the CTC decoder enforces the constraint that the predicted target length has to be strictly greater than the true target length. In semantic parsing the relation between input and output length depends on the semantics and ontology[1] unlike Machine Translation (Libovický and Helcl, 2018; Saharia et al., 2020) or ASR (Graves et al., 2006), which rely on heuristics based on input length. Accordingly, we introduce a **fuzzy length predictor**, an extension to the length prediction module that multiplies the output by a fuzzy constant $\alpha > 1$ to account for an upper bound. The number of tokens predicted by the decoder, l, is then

$$l = \alpha \times \texttt{LengthModule}(emb_{\text{pool}}) \ .$$

This constant helps ensure that even if the length prediction is smaller than the true target length, the decoder is able to output enough tokens for the full semantic parse.

Parallel Decoding with CTC The use of CTC loss creates the flexibility in length prediction by enabling the decoder to produce blank tokens, al-

---

[1]An example: "Remind me to call Samantha" -> [IN:CREATE_REMINDER [SL:TODO call Samantha]] vs [IN:CREATE_REMINDER [SL:TODO [IN:CALL [SL:CONTACT Samantha]]]]

lowing it to decode any sequence length that is shorter than the fuzzy predicted length.

Our decoder leverages the deliberation encoder's hidden states and length module's [MASK] tokens to decode a semantic parse non-autoregressively:

$$y_{1...l} = \text{Decoder}([\text{MASK}]_{1...l}, emb_{\text{pool}}) \ ,$$

where the decoder is a randomly-initialized Transformer decoder. As a result of CTC loss, the predicted parse $y_1, ..., y_l$ can have repeating tokens and blank tokens. We then apply the CTC alignment procedure (Graves et al., 2006), which merges repeating tokens and removes blank tokens to produce the final prediction.

3.3 Denoising

While the CTC decoder gives the model the ability to rewrite transcription errors, we wish to specialize training to specifically encourage the model to rely on both audio and text signals. Inspired by the success of Masked Language Modeling (Devlin et al., 2019; Lewis et al., 2020) as well as SpecAugment for ASR training (Park et al., 2019) we add additional text corruptions to the ASR hypotheses to add an implicit *denoising* objective during training. Unlike SpecAugment, the noise impacts the *text* in order to force the model to rely on audio representations in order to correct transcription errors. We use two noising methods, deletion and substitution noise, along with two meta operations: sequential noising and sampling noising, summarized in Figure 3.

Deletion Inspired by the BART MLM Objective (Lewis et al., 2020), we implement a noising function which randomly deletes tokens from the input. Critically, this deletion noise does not insert a MASK token; instead the model must rely on the audio modality to find the corruption and find an appropriate replacement. We choose this approach to mirror the impact of ASR deletions. To implement the noise we employ a binomial distribution where the number of trials is the sequence length, and we introduce a hyper parameter P that controls the probability at each trial to determine the number of tokens (N) corrupted from the input. We then uniformly select N tokens from the input and delete them.

Substitution We also noise input words via word substitution. To find word substitution pairs which emulate true ASR errors we build a dictionary of word confusion pairs from inference of the 10M

ASR model over the STOP train set. Similar to deletion noise, we use a binomial distribution to select words for substitution. A word only has substitution applied if it appears in the substitution dictionary. If it does, we sample a replacement word according to the frequency each replacement has appeared for it in the ASR output.

Meta Operations With the different noising functions we implemented above, we introduce two meta noising functions that allow us to combine them. 1) Sequential Noise: This mechanism applies R different noise types sequentially, each following the parameterization above. In other words, all samples have each noise type applied in series. 2) Sampling Noise: This mechanism uniformly samples amongst the R different noise types following the parameterization above.

3.4 Training

We jointly optimize for training the decoder and length prediction module with a label loss and length loss, respectively. The length loss is negative log likelihood between predicted lengths and target lengths:

$$\mathcal{L}_{length} = -logP(l_{target}|l_{pred}) \ .$$

The label loss is a CTC loss between decoder predictions and target labels:

$$\mathcal{L}_{label} = -logP(Y|X) \ .$$

where Y is the target labels and X is the decoder predictions before removing blanks and repeated tokens. The conditional likelihood $P(Y|X)$ is the sum of probabilities of all possible valid alignments between X and Y. Let π denote an alignment, $\mathcal{B}$ be the set of all valid alignments, and y_k^t denote the probability of observing label k at time t. Then,

$$P(Y|X) = \sum_{\pi \in \mathcal{B}} p(\pi|X) \ ,$$

$$p(\pi|X) = \prod_{t=1}^{T} y_{\pi_t}^t \ .$$

We add label smoothing to both label loss and length loss to control for over-confident predictions. The overall loss is a weighted sum of these two losses:

$$\mathcal{L} = \mathcal{L}_{label} + \lambda \times \mathcal{L}_{length} \ .$$
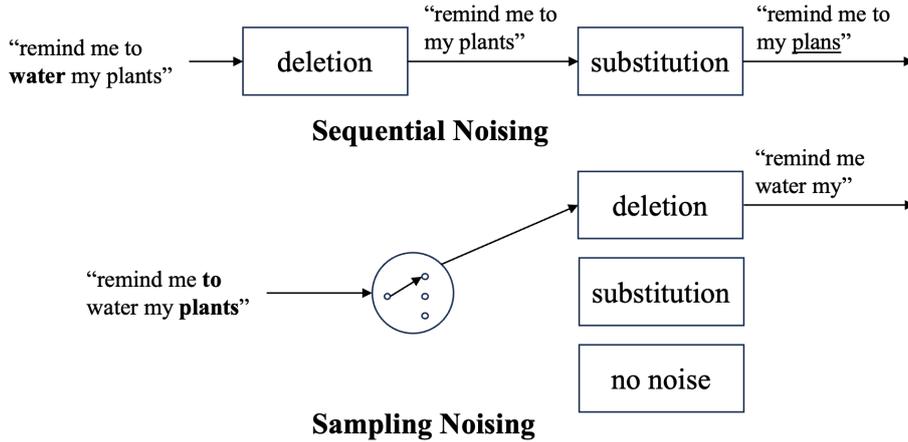
Figure 3: The text noising process. We proposed two noise processes, deletion and substitution noise, as well as two meta-operations, sequential and sampling noising.

4 Experiments

4.1 STOP Dataset

We leverage the STOP dataset (Spoken Task Oriented Parsing) (Tomasello et al., 2023) to evaluate our proposed approach. STOP is based on the TOPv2 dataset (Chen et al., 2020), a compositional multi-domain semantic parsing benchmark that has been well-studied in text parsing. STOP consists of 8 different domains: alarm, messaging, music, navigation, timer, weather, reminder, and event. Each data point contains an audio clip from Amazon Mechanical Turk workers, where their names and information remain anonymous. The dataset is split into three subsets: 120k training samples, 33k validation samples, and 76k test samples.

4.2 Baselines

Our study targets improving Deliberation based SLU systems (Le et al., 2022) for on-device streaming use cases. We align our baselines with Le et al. (2022) to study our contributions.

Base ASR Model   Our studies leverage two on-device-sized ASR models containing 10M and 25M RNNT (Graves, 2012) variants. These models use an acoustic encoder-based conformer encoder (Gulati et al., 2020; Shi et al., 2022) (3L for 10M parameter model and 13L for 25M model), a predictor with a 1-layer LSTM, and a 1-layer Joiner network. We used a 4-stride, 40ms lookahead, 120ms segment size audio input features, and 4000 sentence piece targets (Kudo and Richardson, 2018). We employed the same training recipe as proposed originally for E2E deliberation-based SLU (Le et al., 2022). The Word Error Rates (WER) for each split

of the STOP datasets, using ASRs of two different sizes (10M and 25M), are presented in Table 1.

Table 1: The WER for each split of the STOP datasets, using the frozen RNNT ASR of two different sizes.

	WER		
Frozen ASR (RNNT)	**train**	**valid**	**test**
10M	7.68	6.99	6.54
25M	4.19	3.83	3.54

Autoregressive (AR) Deliberation   Following Le et al. (2022), we constrain all our model architectures to leverage 5M parameters to enable an iso-size comparison. All models also use the same decoding vocabulary, consisting of 4095 unigram WordPieces plus the vocabulary of ontology tokens built from training data. The 5M AR deliberation model leverages 2 pooling (encoder) layers, 1 decoder layers, with hidden dimension of 224. This AR deliberation model has the best quality and latency performance across various encoder and decoder depths (Table 7 in Appendix A).

Mask Predict Deliberation   Our first non-autoregressive architecture extends the AR decoder with the Mask Predict formulation. We introduce a length module after the AR deliberation encoder to predict the output length for non-autoregressive decoding. During training, golden length is used as input, while during testing, predicted length from length module is used. Iterative refinement is not employed in order to focus on efficient one-shot decoding with minimal latency. Our 5M Mask Predict Deliberation model leverages 4 pooling (encoder)

Table 2: Evaluation on the STOP test split. PRoDeliberation outperforms Mask Predict, is on par with AR model for 10M ASR, and outperforms for 25M ASR.

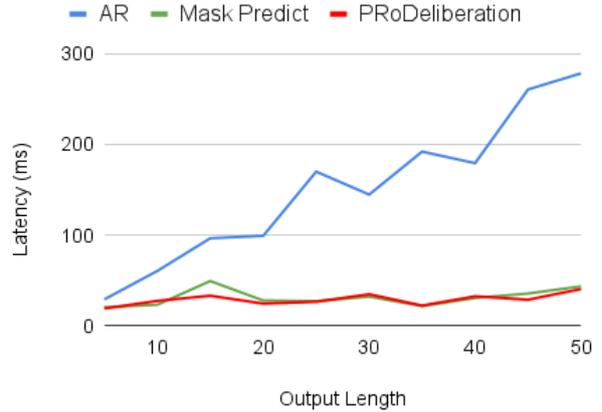| ASR	NLU	EM
10M	Deliberation AR (Le et al., 2022)	67.90
	Deliberation AR (Kim et al., 2023)	**68.37**
	Deliberation AR (Sharma et al., 2023)	67.37
	Mask Predict (Ghazvininejad et al., 2019)	67.23
	ProDeliberation (ours)	68.31
25M	Deliberation AR (Le et al., 2022)	73.87
	Deliberation AR (Kim et al., 2023)	74.05
	Mask Predict (Ghazvininejad et al., 2019)	72.17
	ProDeliberation (ours)	**74.27**

Figure 4: Latency by output length of AR, Mask Predict and ProDeliberation. AR latency increases linearly with output length, while the NAR models (Mask Predict and ProDeliberation) are approximately constant.

layers, 3 decoder layers, with hidden dimension of 200.

PRoDeliberation   We introduce a fuzzy length prediction and replaces the Mask Predict loss with a CTC loss to mitigate the length bottleneck. To optimize for latency reduction, iterative refinement is not utilized. Our 5M ProDeliberation model leverages 4 pooling (encoder) layers, 3 decoder layers, with hidden dimension of 200.

4.3   Training Hyperparameters

All baseline and proposed models are trained with 8 GPUs of 32GB memory each, using AdamW optimizer and tri-stage learning rate scheduler. All training is done with SpecAugment enabled (Park et al., 2019).

Hyperparameters, including warmup steps, hold steps, decay steps, learning rate, encoder and decoder dropouts, scaling factors in loss, and denoising probabilities, are obtained through hyperparameter sweeps using Bayesian optimization to maximize Exact Match (EM) accuracy on the eval split of STOP dataset. These hyperparameter values can be found in Table 8, Appendix A.

5   Results

Quality Analysis   On the 10M ASR model, PRoDeliberation achieves better performance than Mask Predict deliberation model and has similar performance to the AR deliberation model. For 25M ASR model, PRoDeliberation outperforms both Mask Predict and AR (Table 2). All models are trained on the STOP train split, optimized

through hyperparameter sweeps on the STOP validation split, and the best run is evaluated on the STOP test split. The metric used is Exact Match, which is the sample-level accuracy for exactly predicting the target tokens.

Latency Analysis   To measure the latency of the models, we first use Torchscript to export the NLU-equivalent module of the baselines and proposed model, each of which consists of fusion layers and an AR decoder or a CTC decoder. We run the Torchscript modules on an X86-64 CPU with 224 GB RAM with various output sequence lengths ranging from 5 to 50 tokens. For each output sequence length, latency is reported by averaging latency measurements of 50 runs.

The PRoDeliberation model has significantly lower latency than AR deliberation model; as shown in Figure 4, it is ~2x faster for short output sequences (<10 tokens) and ~10x faster for long output sequences (>45 tokens). PRoDeliberation has similar latency to Mask Predict deliberation, which is expected as both models have non-autoregressive decoders. Due to their non-autoregressive decoding nature, their latency is largely unaffected by the output length, while AR deliberation latency scales linearly with the output length.

6   Analysis

6.1   Architecture Ablation

Encoder vs Decoder Size   We study the importance of encoder and decoder depths in ProDeliberation by varying these depths while keeping model

size constant (5M parameters) by adjusting the hidden dimension accordingly. Increasing encoder depth up to 4 encoder layers leads to better performance, but more than 4 encoder layers do not further improve the model performance (Table 3). We observe similar trend for decoder depth, where 3 decoder layers results in the best performance, but deeper decoder than 3 layers does not lead to further improvement (Table 4). Given the same number of parameters, optimal PRoDeliberation model (4 encoder layers, 3 decoder layers, 200 hidden dimension) requires deeper and narrower architecture than optimal AR Deliberation (2 encoder layers, 1 decoder layer, 224 hidden dimension, Table 7).

Table 3: Exact Match of ProDeliberation on STOP test set, with varying encoder depths at 5M parameters.

Enc. layers	Dec. layers	Hidden dim.	Total params	EM
2	3	224	5M	66.45
3	3	208	5M	67.84
4	3	200	5M	**68.31**
5	3	184	5M	67.93
6	3	176	5M	68.12

Table 4: Exact Match of ProDeliberation on STOP test set, with varying decoder depths at 5M parameters.

Enc. layers	Dec. layers	Hidden dim.	Total params	EM
4	1	216	5M	67.18
4	2	208	5M	67.73
4	3	200	5M	**68.31**
4	4	184	5M	68.05
4	5	176	5M	68.06

Impact of Target length scale As mentioned in Section 3.2, we scale the predicted output length by a factor α to ensure the sequence length for the CTC loss is greater than or equal to the true output length. Here we investigate how this scale factor α affects the latency and quality of ProDeliberation.

Figure 5 shows a minimal impact on latency when we vary target length scale α. This is expected as ProDeliberation decodes all output tokens at the same time, so output sequence length does not significantly effect inference latency.

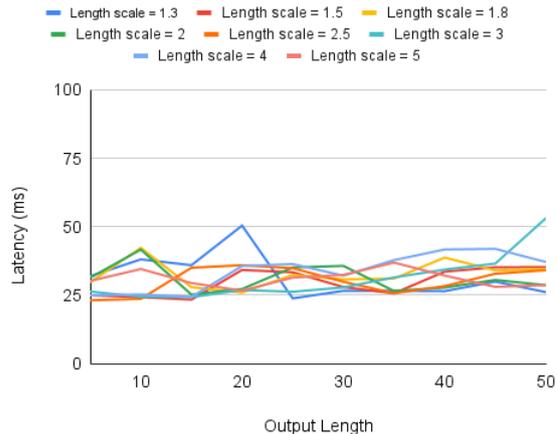We observe that α being too small (<1.8) leads

Figure 5: Latency by output length with different target length scale factor α.

to very poor performance, while large α (>2) does not affect performance nearly as much, and α = 2 results in the best performance (Table 9, Appendix A).

6.2 Denoising ablation

We perform an ablation study on the denoising, investigating the relative importance of substitution and deletion noise, as well as the relative performance of the meta noising strategies. The results are summarized in Table 5. For each setting, we sweep to find the optimal denoising parameter(s).

Sampling between deletion and substitution noise shows the best performance for both 10M and 25M ASR. The sampling denoise strategy requires higher substitution probability than deletion probability in both 10M and 25M models. The substitution-only denoise model also performs better than the deletion-only denoise model with 10M ASR and is on-par with the deletion-only denoise model with 25M ASR. 10M ASR requires lower substitution probability but slightly higher deletion probability than 25M ASR. All models with denoising perform better than models with no noise. Moreover, performance without denoising is on par with the autoregressive model; denoising improves performance further.

From the error breakdown in Table 6 we see that when the ASR model is smaller (and therefore more error prone) a greater share of the improvement from denoising comes from the samples with ASR errors. Moreover, there is greater *overall* improvement with a smaller ASR model. This indicates that the denoising works as expected: by

Table 5: Denoising ablation study. For each ASR model size, we compare denoising strategies.

Denoise strategy	10M ASR			25M ASR		
	Subs prob	Del prob	EM	Subs prob	Del prob	EM
samp(del, subs)	0.088	0.003	**68.31**	0.026	0.007	**74.27**
del	-	0.012	67.8	-	0.024	74.07
subs	0.069	-	68.02	0.04	-	74.02
no noise	-	-	67.95	-	-	73.96

Table 6: Exact Match of E2E models with different ASR model sizes. We break down EM into samples with ASR errors and samples without ASR errors.

	10M ASR		25M ASR	
EM	CTC	CTC+D	CTC	CTC+D
total	67.95	**68.31**	73.96	**74.27**
no ASR error	**85.18**	85.14	82.95	**83.14**
ASR error	25.70	**27.00**	54.36	**54.73**

improving robustness to ASR errors.

7 Conclusion

We presented a novel SLU model, PRoDeliberation, which uses a CTC decoder and noise augmentation to train a parallelized and robust E2E SLU model. This model achieves much lower latency than the state-of-the-art models while maintaining the ability to correct errors in the ASR transcript. Further, we analyzed the system's performance across two ASR model sizes and provided ablations guiding the decisions made when designing PRoDeliberation. We also show that the denoising training for PRoDeliberation allows it to overcome the limitations of very small ASR models. This represents a significant step forward in building E2E SLU systems that are small enough and have low enough latency to run on edge devices.

8 Limitations

Even though PRoDeliberation shows significant latency reduction compared to autoregressive deliberation while retaining the model quality, PRoDeliberation still has its own limitations.

Lack of beam search support Autoregressive deliberation model can use beam search with larger beam size to improve the model's generation quality, at the cost of added latency. PRoDeliberation, on the other hand, has a non-autoregressive decoder and cannot directly use beam search to fur-

ther augment the model performance. We can extend PRoDeliberation to allow for beam search in a couple of ways. Similar to (Xiao et al., 2023), we can use multiple candidate lengths from the length prediction module and decode according to each in parallel. Another possibility is to look into fusion mechanisms with language models similar to (Futami et al., 2022). However, one thing to note is that in the on-device space which deliberation models target (Le et al., 2022), the added latency from larger beam size is rarely tolerated and beam size of 1 is often used.

Data requirements This approach requires access to pairs of audio data with semantic parse labels. Though this is a common requirement for end-to-end SLU models, it does pose a more difficult data requirement than cascaded systems, which require audio samples with text labels and text samples with semantic parse labels, the combination of which is easier to generate than the audio data with semantic parse labels.

While this paper focused on methods which minimize latency, there are other mechanisms for further improving quality at the expense of increased latency. Iterative refinement is an approach that would have this trade-off, which can be used for both ProDeliberation as well as the Mask Predict baseline. Future work may explore this approach to offer approaches with different balances between latency and quality.

References

Armen Aghajanyan, Jean Maillard, Akshat Shrivastava, Keith Diedrick, Michael Haeger, Haoran Li, Yashar Mehdad, Veselin Stoyanov, Anuj Kumar, Mike Lewis, and Sonal Gupta. 2020. Conversational Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Arun Babu, Akshat Shrivastava, Armen Aghajanyan,

Ahmed Aly, Angela Fan, and Marjan Ghazvininej. 2021. Non-Autoregressive Semantic Parsing for Compositional Task-Oriented Dialog. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. *CoRR*, abs/2011.13205.

Xilun Chen, Ashish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Hayato Futami, Hirofumi Inaguma, Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2022. Non-autoregressive error correction for ctc-based asr with phone-conditioned masked lm. *arXiv preprint arXiv:2209.04062*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

A. Graves. 2012. Sequence transduction with recurrent neural networks. In *ICML Representation Learning Workshop*.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *Proceedings of the 23rd international conference on Machine learning*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447.

Kevin Hu, Rohit Prabhavalkar, Ruoming Pang, and Tara Sainath. 2020. Deliberation Model Based Two-Pass End-to-End Speech Recognition. In *Proc. ICASSP*.

Yosuke Kashiwagi, Siddhant Arora, Hayato Futami, Jessica Huynh, Shih-Lun Wu, Yifan Peng, Brian Yan, Emiru Tsunoo, and Shinji Watanabe. 2023. E-branchformer-based e2e slu toward stop on-device challenge. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2.

Suyoun Kim, Akshat Shrivastava, Duc Le, Ju Lin, Ozlem Kalinli, and Michael L. Seltzer. 2023. Modality confidence aware training for robust end-to-end spoken language understanding. *ArXiv*, abs/2307.12134.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Duc Le, Akshat Shrivastava, Paden Tomasello, Suyoun Kim, Aleksandr Livshits, Ozlem Kalinli, and Michael L Seltzer. 2022. Deliberation model for on-device spoken language understanding. *Interspeech*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Conference on Empirical Methods in Natural Language Processing*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Martin H. Radfar, Athanasios Mouchtaris, Siegfried Kunzmann, and Ariya Rastrow. 2021. Fans: Fusing asr and nlu for on-device slu. In *Interspeech*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rastrow. 2020. Speech to semantics: Improve asr and nlu jointly via all-neural interfaces. In *Proc. INTERSPEECH*.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. *ArXiv*, abs/2004.07437.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Roshan Sharma, Suyoun Kim, Daniel Lazar, Trang Le, Akshat Shrivastava, Kwanghoon Ahn, Piyush Kansal, Leda Sari, Ozlem Kalinli, and Michael Seltzer. 2023. Augmenting text for spoken language understanding with large language models. *arXiv preprint arXiv:2309.09390*.

Yangyang Shi, Chunyang Wu, Dilin Wang, Alex Xiao, Jay Mahadeokar, Xiaohui Zhang, Chunxi Liu, Ke Li, Yuan Shangguan, Varun Nagaraja, et al. 2022. Streaming transformer transducer based speech recognition using non-causal convolution. *Proc. ICASSP*.

Akshat Shrivastava, Pierce Chuang, Arun Babu, Shrey Desai, Abhinav Arora, Alexander Zotov, and Ahmed Aly. 2021. Span Pointer Networks for Non-Autoregressive Task-Oriented Semantic Parsing. In *Proceedings of the Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Akshat Shrivastava, Shrey Desai, Anchit Gupta, Ali Elkahky, Aleksandr Livshits, Alexander Zotov, and Ahmed Aly. 2023. Retrieve-and-fill for scenario-based task-oriented semantic parsing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 430–447, Dubrovnik, Croatia. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Paden Tomasello, Akshat Shrivastava, Daniel Lazar, Po-Chun Hsu, Duc Le, Adithya Sagar, Ali Elkahky, Jade Copet, Wei-Ning Hsu, Yossef Mordechay, Robin Algayres, Tu Anh Nguyen, Emmanuel Dupoux, Luke Zettlemoyer, and Abdelrahman Mohamed. 2023. STOP: A dataset for Spoken Task Oriented Semantic Parsing. In *CoRR*.

Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-yan Liu. 2023. A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Qile Zhu, Haidar Khan, Saleh Soltan, Stephen Rawls, and Wael Hamza. 2020. Don't Parse, Insert: Multilingual Semantic Parsing with Insertion Based Decoding. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

A   Appendix

Table 7: Exact Match and Latency at output length = 10 and output length = 50 of AR Deliberation on STOP eval set, with varying encoder and decoder depths at 5M parameters.

Enc. layers	Dec. layers	Hidden dim.	Total params	EM	Latency @ output_len=10	Latency @ output_len=20
2	1	224	5M	68.37	58	267
3	1	200	5M	67.85	62	273
4	1	192	5M	67.86	63	271
5	1	176	5M	68.11	71	301
2	2	200	5M	68.27	98	365
2	3	176	5M	68.39	124	510
2	4	168	5M	68.02	157	771
2	5	152	5M	68.37	294	953

Table 8: Hyperparameter values of PRoDeliberation.

Hyperparameter	10M ASR	25M ASR
Length loss scaling factor (λ)	0.2504	0.1260
Target length scale (α)	2	2
Decoder attention dropout	0.1784	0.4991
Encoder dropout	0.1697	0.2646
Deletion probability	0.0026	0.0070
Substitution probability	0.0882	0.0256
Learning rate	0.00085	0.00053
Warm up epochs	10	10
Hold epochs	40	90
Decay epochs	95	105

Table 9: Exact Match of PRoDeliberation on STOP test set, with varying target length scale.

Target length scale	1.3	1.5	1.8	2	2.5	3	4
Exact Match	49.50	49.95	62.44	**68.31**	66.48	67.33	68.08