Rethinking Label Smoothing on Multi-hop Question Answering

Zhangyue Yin[♢*] **Yuxin Wang**[♢*] **Xiannian Hu**[♢] **Yiguang Wu**[♢] **Hang Yan**[♢]
Xinyu Zhang[♠] **Zhao Cao**[♠] **Xuanjing Huang**[♢] **Xipeng Qiu**[♢†]
[♢]School of Computer Science, Fudan University
[♠]Huawei Poisson Lab
{yinzy21,wangyuxin21,xnhu21}@m.fudan.edu.cn
{ygwu20,hyan19,xjhuang,xpqiu}@fudan.edu.cn
{zhangxinyu35,caozhao1}@huawei.com

Abstract

Multi-Hop Question Answering (MHQA) is a significant area in question answering, requiring multiple reasoning components, including document retrieval, supporting sentence prediction, and answer span extraction. In this work, we present the first application of label smoothing to the MHQA task, aiming to enhance generalization capabilities in MHQA systems while mitigating overfitting of answer spans and reasoning paths in the training set. We introduce a novel label smoothing technique, F1 Smoothing, which incorporates uncertainty into the learning process and is specifically tailored for Machine Reading Comprehension (MRC) tasks. Moreover, we employ a Linear Decay Label Smoothing Algorithm (LDLA) in conjunction with curriculum learning to progressively reduce uncertainty throughout the training process. Experiment on the HotpotQA dataset confirms the effectiveness of our approach in improving generalization and achieving significant improvements, leading to new state-of-the-art performance on the HotpotQA leaderboard.

1 Introduction

Multi-Hop Question Answering (MHQA) is a rapidly evolving research area within question answering that involves answering complex questions by gathering information from multiple sources. This requires a model capable of performing several reasoning steps and handling diverse information structures. In recent years, MHQA has attracted significant interest from researchers due to its potential for addressing real-world problems. The mainstream approach to MHQA typically incorporates several components, including a document retriever, a supporting sentence selector, and a reading comprehension module (Tu et al., 2020; Wu et al., 2021; Li et al., 2022). These components collaborate to accurately retrieve and integrate relevant information from multiple sources, ultimately providing a precise answer to the given question.

Despite the remarkable performance of modern MHQA models in multi-hop reasoning, they continue to face challenges with answer span errors and multi-hop reasoning errors. A study by S2G (Wu et al., 2021) reveals that the primary error source is answer span errors, constituting 74.55%, followed by multi-hop reasoning errors. This issue arises from discrepancies in answer span annotations between the training and validation sets. As illustrated in Figure 1(a), the training set answer includes the quantifier “times”, while the validation set answer does not. Upon examining 200 samples, we found that around 13.7% of answer spans in the HotpotQA validation set deviate from those in the training set.

Concerning multi-hop reasoning, we identified the presence of unannotated, viable multi-hop reasoning paths in the training set. As depicted in Figure 1(b), the non-gold document contains the necessary information to answer the question, similar to gold doc1, yet is labeled as an irrelevant document. During training, the model can only discard this reasoning path and adhere to the annotated reasoning path. Given that current MHQA approaches primarily use cross-entropy loss for training multiple components,

[*]Equal contribution.
[†] Corresponding author.

©2023 China National Conference on Computational Linguistics
Published under Creative Commons Attribution 4.0 International License

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Training set:
Gold Doc1: ***Love or Leave***
"Love or Leave" was the Lithuanian entry in the Eurovision Song Contest 2007, performed in English by 4FUN.
Gold Doc2: ***Lithuania in the Eurovision Song Contest***
Lithuania has participated in the Eurovision Song Contest (known in Lithuania as "Eurovizija") 18 times since its debut in 1994, where Ovidijus Vyšniauskas finished last, receiving nul points.
Question: How many times does the song writer of "Love or Leave" have participated in the Eurovision Song Contest?
Answer: 18 times

Validation set:
Gold Doc1: ***Binocular (horse)***
"Love or Leave" was the Lithuanian entry in the Eurovision Song Contest 2007, performed in English by 4FUN.
Gold Doc2: ***Tony McCoy***
Based in Ireland and the UK, McCoy rode a record 4,358 winners, and was Champion Jockey a record 20 consecutive times, every year he was a professional.
Question: The primary jockey of Binocular was Champion Jockey how many consecutive times?
Answer: 20

(a) Different Answer Span

Non-Gold Doc: ***F. W. Woolworth Building (Watertown, New York)***
(1) The Woolworth Building is an historic building in Watertown, New York.
(2) It is a contributing building in the Public Square Historic District.
(3) Plans for the Woolworth Building were begun in 1916 by Frank W. Woolworth, the founder of the Woolworth's chain of department stores.

Gold Doc1: ***Woolworth Building***
(1) The Woolworth Building, at 233 Broadway, Manhattan, New York City, designed by architect Cass Gilbert and constructed between 1910 and 1912, is an early US skyscraper.

Gold Doc2: ***1 New York Plaza***
(1) 1 New York Plaza is an office building in New York City's Financial District, built in 1969 at the intersection of South and Whitehall Streets.
(2) It is the southernmost of all Manhattan skyscrapers.

Question: Which was built first Woolworth Building or 1 New York Plaza?
Answer: Woolworth Building
Evidence Sentences: ["Woolworth Building", 0], ["1 New York Plaza",0]

(b) Multiple Feasible Reasoning Paths

Figure 1: Causes of errors in answer span and multi-hop reasoning within the HotpotQA dataset. In Figure (a), the answer from the training set contains a quantifier, while the answer from the validation set does not. Figure (b) demonstrates that the correct answer can be inferred using a non-gold document without requiring information from gold doc1.

they tend to overfit annotated answer spans and multi-hop reasoning paths in the training set. Consequently, we naturally pose the research question for this paper: *How can we prevent MHQA models from overfitting answer spans and reasoning paths in the training set?*

Label smoothing is an effective method for preventing overfitting, widely utilized in computer vision (Szegedy et al., 2016). In this study, we introduce label smoothing to multi-hop reasoning tasks for the first time to mitigate overfitting. We propose a simple yet efficient MHQA model, denoted as $\mathbf{R}^3$, comprising **R**etrieval, **R**efinement, and **R**eading Comprehension modules. Inspired by the F1 score, a commonly used metric for evaluating MRC task performance, we develop F1 Smoothing, a novel technique that calculates the significance of each token within the smooth distribution. Moreover, we incorporate curriculum learning (Bengio et al., 2009) and devise the **L**inear **D**ecay **L**abel Smoothing **A**lgorithm (LDLA), which gradually reduces the smoothing weight, allowing the model to focus on more challenging samples during training. Experimental results on the HotpotQA dataset (Yang et al., 2018) demonstrate that incorporating F1 smoothing and LDLA into the $\mathbf{R}^3$ model significantly enhances performance in document retrieval, supporting sentence prediction, and answer span selection, achieving state-of-the-art results among all published works.

Our main contributions are summarized as follows:

- We introduce label smoothing to multi-hop reasoning tasks and propose a baseline model, $\mathbf{R}^3$, with retrieval, refinement, and reading comprehension modules.
- We present F1 smoothing, a novel label smoothing method tailored for MRC tasks, which alleviates errors caused by answer span discrepancies.
- We propose LDLA, a progressive label smoothing algorithm integrating curriculum learning.
- Our experiments on the HotpotQA dataset demonstrate that label smoothing effectively enhances the MHQA model's performance, with our proposed LDLA and F1 smoothing achieving state-of-the-art results.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

2 Related Work

Label Smoothing Label smoothing is a regularization technique first introduced in computer vision to improve classification accuracy on ImageNet (Szegedy et al., 2016). The basic idea of label smoothing is to soften the distribution of true labels by replacing their one-hot encoding with a smoother version. This approach encourages the model to be less confident in its predictions and consider a broader range of possibilities, reducing overfitting and enhancing generalization (Pereyra et al., 2017; Müller et al., 2019; Lukasik et al., 2020a). Label smoothing has been widely adopted across various natural language processing tasks, including speech recognition (Chorowski and Jaitly, 2017), document retrieval (Penha and Hauff, 2021), dialogue generation (Saha et al., 2021), and neural machine translation (Gao et al., 2020; Lukasik et al., 2020b; Graça et al., 2019).

In addition to traditional label smoothing, several alternative techniques have been proposed in recent research. For example, Xu et al. (2020) suggested the **T**wo-**S**tage **LA**bel smoothing (TSLA) algorithm, which employs a smoothing distribution in the first stage and the original distribution in the second stage. Experimental results demonstrated that TSLA effectively promotes model convergence and enhances performance. Penha and Hauff (2021) introduced label smoothing for retrieval tasks and proposed using BM25 to compute the label smoothing distribution, which outperforms the uniform distribution. Zhao et al. (2020) proposed Word Overlapping, which uses maximum likelihood estimation (Su et al., 2020) to optimally estimate the model's training distribution.

Multi-hop Question Answering Multi-hop reading comprehension (MHRC) is a demanding task in the field of machine reading comprehension (MRC) that closely resembles the human thought process in real-world scenarios. Consequently, it has gained significant attention in the field of natural language understanding in recent years. Several datasets have been developed to foster research in this area, including HotpotQA (Yang et al., 2018), WikiHop (Welbl et al., 2018), and NarrativeQA (Kočiský et al., 2018). Among these, HotpotQA (Yang et al., 2018) is particularly representative and challenging, as it requires the model to not only extract the correct answer span from the context but also identify a series of supporting sentences as evidence for MHRC.

Recent advances in MHRC have led to the development of several graph-free models, such as QUARK (Groeneveld et al., 2020), C2FReader (Shao et al., 2020), and S2G (Wu et al., 2021), which have challenged the dominance of previous graph-based approaches like DFGN (Qiu et al., 2019), SAE (Tu et al., 2020), and HGN (Fang et al., 2020). C2FReader (Shao et al., 2020) suggests that the performance difference between graph attention and self-attention is minimal, while S2G's (Wu et al., 2021) strong performance demonstrates the potential of graph-free modeling in MHRC. FE2H (Li et al., 2022), which uses a two-stage selector and a multi-task reader, currently achieves the best performance on HotpotQA, indicating that pre-trained language models alone may be sufficient for modeling multi-hop reasoning. Motivated by the design of S2G (Wu et al., 2021) and FE2H (Li et al., 2022), we introduce a our model $\mathbf{R}^3$.

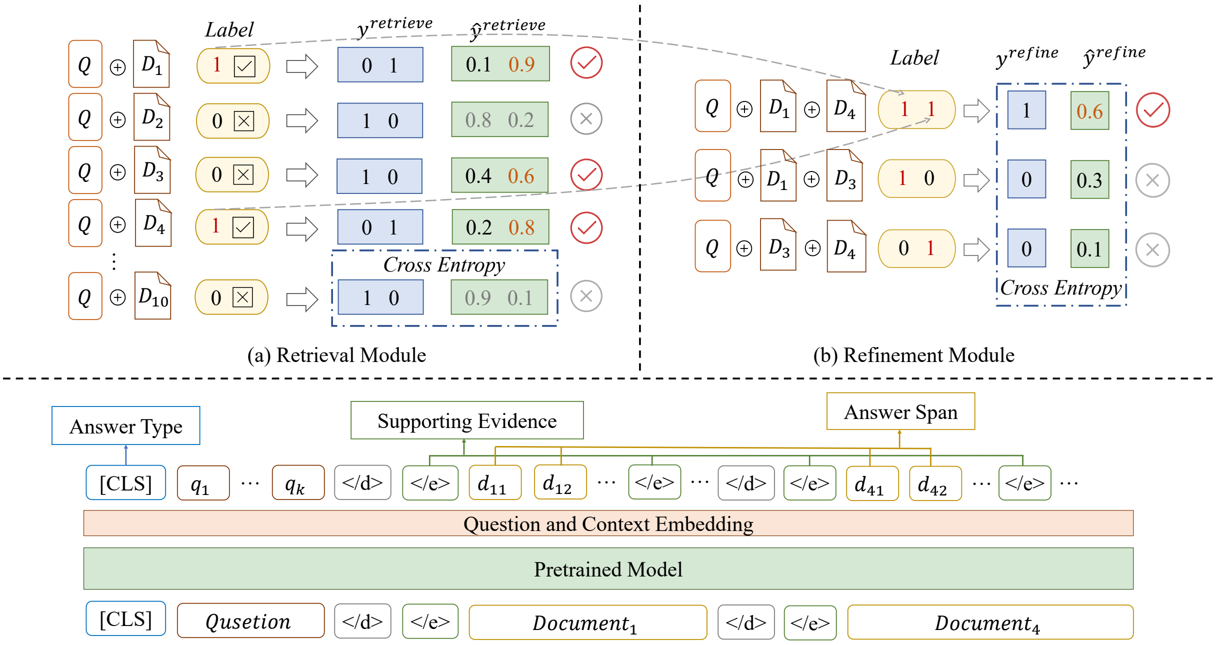
3 Framework

Figure 2 depicts the overall architecture of $\mathbf{R}^3$. The retrieval module serves as the first step, where our system selects the most relevant documents, which is essential for filtering out irrelevant information. In this example, document1, document3, and document4 are chosen due to their higher relevance scores, while other documents are filtered out. Once the question and related documents are given, the refinement module further selects documents based on their combined relevance. In this instance, the refinement module opts for document1 and document4. Following this, the question and document1, document4 are concatenated and used as input for the reading comprehension module. Within the reading comprehension module, we concurrently train supporting sentence prediction, answer span extraction, and answer type selection using a multi-task approach.

3.1 Retrieval Module

In the retrieval module, each question Q is typically accompanied by a set of M documents $D_1, D_2 \ldots, D_M$, but only $C, |C| << M$ (two in HotpotQA) are genuinely relevant to question Q.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China



(a) Retrieval Module

(b) Refinement Module

(c) Reading Comprehension Module

Figure 2: Overview of our $\mathbf{R}^3$ model, which consists of three main modules: **R**etrieval, **R**efinement, and **R**eading Comprehension.

CCL 2023

We model the retrieval process as a binary classification task. Specifically, for each question-document pair, we generate an input by concatenating [CLS], question, [SEP], document, and [SEP] in sequence. We then feed the [CLS] token output from the model into a linear classifier. $\mathcal{L}_{\text{retrieve}}$ represents the cross-entropy between the predicted probability and the gold label. In contrast to S2G (Wu et al., 2021), which employs a complex pairwise learning-to-rank loss, we opt for a simple binary cross-entropy loss, as it maintains high performance while being significantly more efficient.

$$\begin{aligned}\mathcal{L}_{\text{retrieve}} = \mathbb{E}[-\frac{1}{M}\sum_{i=1}^{M}(y_i^{\text{retrieve}} \cdot log(\hat{y}_i^{\text{retrieve}}) \\ +(1-y_i^{\text{retrieve}}) \cdot log(1-\hat{y}_i^{\text{retrieve}}))],\end{aligned} \tag{1}$$

where $\hat{y}_i^{\text{retrieve}}$ is the probability predicted by the model and y_i^{retrieve} is the ground-truth label. M is the number of provided documents. $\mathbb{E}$ means the expectation of all samples.

$$y_i^{\text{retrieve}} = \begin{cases} 1 & D_i \text{ is a golden document.} \\ 0 & D_i \text{ is a non-golden document.} \end{cases} \tag{2}$$

3.2 Refinement Module

In the refinement module, we select the top K relevant documents from the previous step and form pairs, resulting in C_K^2 combinations. Emphasizing inter-document interactions crucial for multi-hop reasoning, we concatenate the following sequence: [CLS], question, [SEP], document1, [SEP], document2, [SEP]. Similar to the retrieval module, we extract the [CLS] token output from the model and pass it through a classifier. Pairs containing two gold-standard documents are labeled as 1, while others are labeled as 0. The refinement module thus filters out irrelevant documents, producing a more concise set for further processing.

$$\mathcal{L}_{\text{refine}} = \mathbb{E}[-\sum_{i=1}^{C_K^2} y_i^{\text{refine}} log(\hat{y}_i^{\text{refine}})], \tag{3}$$

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

where $\hat{y}_i^{\text{refine}}$ is predicted document pair probability and y_i^{refine} is the ground-truth label, C_K^2 is number of all combination.

$$y_i^{\text{refine}} = \begin{cases} 1 & \mathcal{C}_i \text{ consists of two gold documents.} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We use a single pretrained language model as the encoder for both the retrieval and refinement module, and the final loss is a weighted sum of $\mathcal{L}_{\text{retrieve}}$ and $\mathcal{L}_{\text{refine}}$. λ_1 and λ_2 are accordingly coefficients of $\mathcal{L}_{\text{retrieve}}$ and $\mathcal{L}_{\text{refine}}$.

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{retrieve}} + \lambda_2 \mathcal{L}_{\text{refine}}. \tag{5}$$

3.3 Reading Comprehension Module

In the reading comprehension module, we use multi-task learning to simultaneously predict supporting sentences and extract answer span. HotpotQA (Yang et al., 2018) contains samples labeled as "yes" or "no". The practice of splicing "yes" and "no" tokens at the beginning of the sequence (Li et al., 2022) could corrupt the original text's semantic information. To avoid the impact of irrelevant information, we introduce an answer type selection header trained with a cross-entropy loss function.

$$\mathcal{L}_{\text{type}} = \mathbb{E}[-\sum_{i=1}^{3} y_i^{\text{type}} log(\hat{y}_i^{\text{type}})], \tag{6}$$

where $\hat{y}_i^{\text{fine}}$ denotes the predicted probability of answer type generated by our model, and y_i^{fine} represents the ground-truth label. answer type includes "yes", "no" and "span".

$$y_i^{\text{type}} = \begin{cases} 0 & \text{Answer is no.} \\ 1 & \text{Answer is yes.} \\ 2 & \text{Answer is a span.} \end{cases} \tag{7}$$

To extract the span of answers, we use a linear layer on the contextual representation to identify the start and end positions of answers, and adopts cross-entropy as the loss function. The corresponding loss terms are denoted as $\mathcal{L}_{\text{start}}$ and $\mathcal{L}_{\text{end}}$ respectively. Similar to previous work S2G (Wu et al., 2021) and FE2H (Li et al., 2022), we also inject a special placeholder token $< /e >$ and use a linear binary classifier on the output of $< /e >$ to determine whether a sentence is a supporting fact. The classification loss of the supporting facts is denoted as $\mathcal{L}_{\text{sup}}$, and we jointly optimize all of these objectives in our model.

$$\mathcal{L}_{\text{reading}} = \lambda_3 \mathcal{L}_{\text{type}} + \lambda_4(\mathcal{L}_{\text{start}} + \mathcal{L}_{\text{end}}) + \lambda_5 \mathcal{L}_{\text{sup}}. \tag{8}$$

4 Label Smoothing

Label smoothing is a regularization technique that aims to improve generalization in a classifier by modifying the ground truth labels of the training data. In the one-hot setting, the probability of the correct category $q(y|x)$ for a training sample (x, y) is typically defined as 1, while the probabilities of all other categories $q(\neg y|x)$ are defined as 0. The cross-entropy loss function used in this setting is typically defined as follows:

$$\mathcal{L} = -\sum_{k=1}^{K} q(k|x)\log(p(k|x)), \tag{9}$$

where $p(k|x)$ is the probability of the model's prediction for the k-th class. Specifically, label smoothing mixes $q(k|x)$ with a uniform distribution $u(k)$, independent of the training samples, to produce a new distribution $q'(k|x)$.

$$q'(k|x) = (1 - \epsilon)q(k|x) + \epsilon u(k), \tag{10}$$

where ϵ is the weight controls the importance of $q(k|x)$ and $u(k)$ in the resulting distribution. $u(k)$ is construed as $\frac{1}{K}$ of the uniform distribution, where K is the total number of categories. Next, we introduce two novel label smoothing methods.

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Algorithm 1 Linear Decay Label Smoothing.

Require: training epochs $n > 0$; smoothing weight $\epsilon \in [0, 1]$; decay rate $\tau \in [0, 1]$; uniform distribution u

Initialize: Model parameter $w_0 \in \mathcal{W}$;
Input: Optimization algorithm $\mathcal{A}$
for $i = 0, 1, \ldots, n$ **do**
    $\epsilon_i \leftarrow \epsilon - i\tau$
    if $\epsilon_i < 0$ **then**
        $\epsilon_i \leftarrow 0$
    end if
    sample(x_t, y_t)
    $y_t^{LS} \leftarrow (1 - \epsilon_i)y_i + \epsilon u$
    $w_{i+1} \leftarrow \mathcal{A}{-}step(w_i; x_i, y_i^{LS})$
end for

4.1 Linear Decay Label Smoothing

Our proposed Linear Decay Label Smoothing Algorithm (LDLA) addresses the abrupt changes in training distribution caused by the two-stage approach of TSLA, which can negatively impact the training process. In contrast to TSLA, LDLA decays the smoothing weight at a constant rate per epoch, promoting a more gradual learning process.

Given a total of n epochs in the training process and a decay size of τ, the smoothing weight ϵ for the i-th epoch can be calculated as follows:

$$\epsilon_i = \begin{cases} \epsilon - i\tau & \epsilon - i\tau \geq 0 \\ 0 & \epsilon - i\tau < 0 \end{cases} \tag{11}$$

Algorithm 1 outlines the specific steps of the LDLA algorithm. LDLA employs the concept of curriculum learning by gradually transitioning the model's learning target from a smoothed distribution to the original distribution throughout the training process. This approach incrementally reduces uncertainty during training, enabling the model to progressively concentrate on more challenging samples and transition from learning with uncertainty to certainty. Consequently, LDLA fosters more robust and effective learning.

4.2 F1 Smoothing

Unlike traditional classification tasks, MRC requires identifying both the start and end positions of a span. To address the specific nature of this task, a specialized smoothing method is required to achieve optimal results. In this section, we introduce F1 Smoothing, a technique that calculates the significance of a span based on its F1 score.

Consider a sample x that contains a context S and an answer a_{gold}. The total length of the context is denoted by L. We use $q_s(t|x)$ to denote the F1 score between a span of arbitrary length starting at position t in S and the ground truth answer a_{gold}. Similarly, $q_e(t|x)$ denotes the F1 score between a_{gold} and a span of arbitrary length ending at position t in S .

$$q_s(t|x) = \sum_{\xi=t}^{L-1} F1\left((t, \xi), a_{\text{gold}}\right). \tag{12}$$

$$q_e(t|x) = \sum_{\xi=0}^{t} F1\left((\xi, t), a_{\text{gold}}\right). \tag{13}$$

The normalized distributions are noted as $q_s^{'}(t|x)$ and $q_e^{'}(t|x)$, respectively.

$$q_s^{'}(t|x) = \frac{exp(q_s(t|x))}{\sum_{i=0}^{L-1} exp(q_s(i|x))}. \tag{14}$$

CCL 2023

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

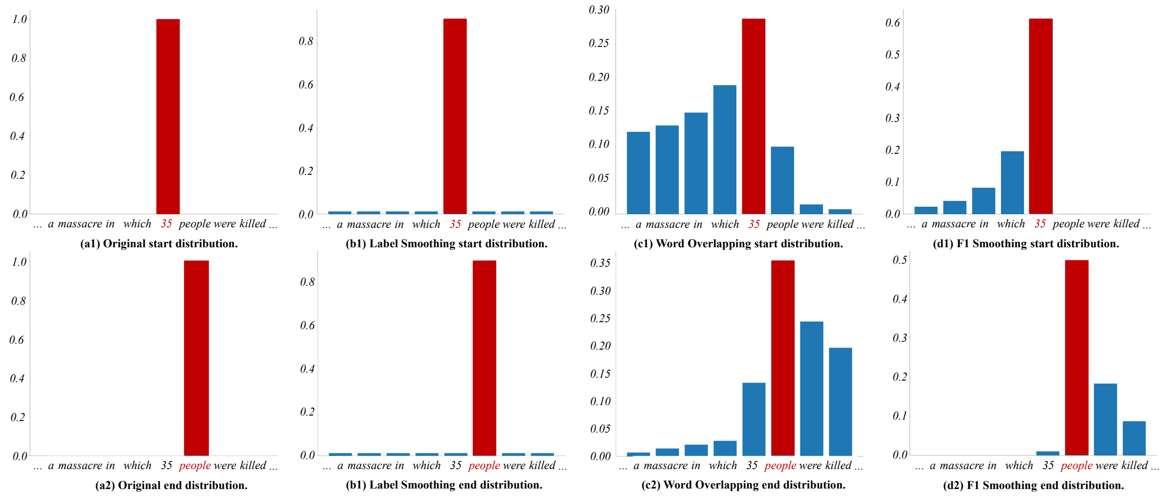


Figure 3: Visualization of original distribution and different label smoothing distributions, including Label Smoothing, Word Overlapping, and F1 Smoothing. The first row shows the distribution of the start token, and the second row shows the distribution of the end token. The gold start and end tokens are highlighted in red.

$$q_e^{'}(t|x) = \frac{exp(q_e(t|x))}{\sum_{i=0}^{L-1} exp(q_e(i|x))}. \tag{15}$$

To decrease the computational complexity of F1 Smoothing, we present a computationally efficient version in Appendix 7. Previous research (Zhao et al., 2020) has investigated various label smoothing methods for MRC, encompassing traditional label smoothing and word overlap smoothing. As illustrated in Figure 3, F1 Smoothing offers a more accurate distribution of token importance in comparison to Word Overlap Smoothing. This method reduces the probability of irrelevant tokens and prevents the model from being misled during training.

5 Experiment

5.1 Dataset

We evaluate our approach on the distractor setting of HotpotQA (Yang et al., 2018), a multi-hop question-answer dataset with 90k training samples, 7.4k validation samples, and 7.4k test samples. Each question in this dataset is provided with several candidate documents, two of which are labeled as gold. In addition to this, HotpotQA also provides supporting sentences for each question, encouraging the model to explain the inference path of the multi-hop question-answer. We use the Exact Match (EM) and F1 score (F1) to evaluate the performance of our approach in terms of document retrieval, supporting sentence prediction, and answer extraction.

5.2 Implementation Details

Our model is built using the Pre-trained language models (PLMs) provided by HuggingFace's Transformers library (Wolf et al., 2020).

Retrieval and Refinement Module We used RoBERTa-large (Liu et al., 2019) and ELECTRA-large (Clark et al., 2020) as our PLMs and conducted an ablation study on RoBERTa-large (Liu et al., 2019). Training on a single RTX3090 GPU, we set the number of epochs to 8 and the batch size to 16. We employed the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 5e-6 and a weight decay of 1e-2.

Reading Comprehension Module We utilized RoBERTa-large (Liu et al., 2019) and DeBERTa-v2-xxlarge (He et al., 2021) as our PLMs, performing ablation studies on RoBERTa-large (Liu et al., 2019). To train RoBERTa-large, we used an RTX3090 GPU, setting the number of epochs to 16 and the batch

CCL 2023

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Model	Answer		Supporting	
	EM	F1	EM	F1
Baseline Model (Yang et al., 2018)	45.60	59.02	20.32	64.49
QFE (Nishida et al., 2019)	53.86	68.06	57.75	84.49
DFGN (Qiu et al., 2019)	56.31	69.69	51.50	81.62
SAE-large (Tu et al., 2020)	66.92	79.62	61.53	86.86
C2F Reader (Shao et al., 2020)	67.98	81.24	60.81	87.63
HGN-large (Fang et al., 2020)	69.22	82.19	62.76	88.47
FE2H on ELECTRA (Li et al., 2022)	69.54	82.69	64.78	88.71
AMGN+ (Li et al., 2021)	70.53	83.37	63.57	88.83
S2G+EGA (Wu et al., 2021)	70.92	83.44	63.86	88.68
FE2H on ALBERT (Li et al., 2022)	71.89	**84.44**	64.98	89.14
$\mathbf{R}^3$ (ours)	71.27	83.57	65.25	88.98
Smoothing $\mathbf{R}^3$ (ours)	**72.07**	84.34	**65.44**	**89.55**

Table 1: In the distractor setting of the HotpotQA test set, our proposed F1 Smoothing and LDLA has led to significant improvements in the performance of the Smoothing $\mathbf{R}^3$ model compared to the $\mathbf{R}^3$ model. Furthermore, the Smoothing $\mathbf{R}^3$ model has outperformed a number of strong baselines and has achieved the highest results.

Model	EM	F1
SAE_{large} (Tu et al., 2020)	91.98	95.76
S2G_{large} (Wu et al., 2021)	95.77	97.82
FE2H_{large} (Li et al., 2022)	96.32	98.02
$\mathbf{R}^3$ (ours)	96.50	98.10
Smoothing $\mathbf{R}^3$	**96.85**	**98.32**

Table 2: Comparison of our retrieval and refinement module with previous baselines on HotpotQA dev set. Label smoothing can further enhance model performance.

CCL 2023

size to 16. For the larger DeBERTa-v2-xxlarge model, we employed an A100 GPU, setting the number of epochs to 8 and the batch size to 16. We used the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 4e-6 for RoBERTa-large and 2e-6 for DeBERTa-v2-xxlarge, along with a weight decay of 1e-2 for optimization.

5.3 Experimental Results

We utilize ELECTRA-large (Clark et al., 2020) as the PLM for the retrieval and refinement modules, and DeBERTa-v2-xxlarge for the reading comprehension module. The $\mathbf{R}^3$ model incorporating F1 Smoothing and LDLA methods is referred to as Smoothing $\mathbf{R}^3$. LDLA is employed for document retrieval and supporting sentence prediction, while F1 Smoothing is applied for answer span extraction. As shown in Table 1, Smoothing $\mathbf{R}^3$ achieves improvements of 0.8% and 0.77% in EM and F1 for answers, and 0.19% and 0.57% in EM and F1 for supporting sentences compared to the $\mathbf{R}^3$ model. Among the tested label smoothing techniques, F1 smoothing and LDLA yield the most significant performance improvement.

We compare the performance of our retrieval and refinement module, which uses ELECTRA-large as a backbone, to three advanced works: SAE (Tu et al., 2020), S2G (Wu et al., 2021), and FE2H (Li et al., 2022). These methods also employ sophisticated selectors for retrieving relevant documents. We evaluate the performance of document retrieval using the EM and F1 metrics. Table 2 demonstrates that our $\mathbf{R}^3$ method outperforms these three strong baselines, with Smoothing $\mathbf{R}^3$ further enhancing performance.

In Table 3, we evaluate the performance of the reading comprehension module, which employs DeBERTa-v2-xxlarge (He et al., 2021) as the backbone, on documents retrieved by the retrieval and

(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Model	Answer		Supporting	
	EM	F1	EM	F1
SAE	67.70	80.75	63.30	87.38
S2G	70.80	-	65.70	-
$\mathbf{R}^3$	71.39	83.84	66.32	89.54
Smoothing $\mathbf{R}^3$	**71.89**	**84.65**	**66.75**	**90.08**

Table 3: Performances of cascade results on the dev set of HotpotQA in the distractor setting.

Setting	EM	F1
Baseline	95.93±.05	97.91±.09
LS	96.06±.11	97.94±.04
TSLA	96.21±.01	98.05±.05
LDLA	**96.57**±.05	**98.18**±.04

Table 4: Various label smoothing methods applied to retrieval modules.

Setting	EM	F1
Baseline	66.94±.05	90.50±.02
LS	66.88±.02	90.53±.02
TSLA	67.42±.05	90.72±.05
LDLA	**67.63**±.04	**90.85**±.03

Table 5: Various label smoothing methods applied to supporting sentence prediction.

refinement module. Our $\mathbf{R}^3$ model outperforms strong baselines SAE and S2G, and further improvements are achieved by incorporating F1 Smoothing and LDLA. These results emphasize the potential for enhancing performance through the application of label smoothing techniques.

5.4 Label Smoothing Analysis

In our study of the importance of label smoothing, we used RoBERTa-large (Liu et al., 2019) as the backbone for our model. To ensure the reliability of our experimental results, we conducted multiple runs with different random number seeds (41, 42, 43, and 44) to ensure stability.

In our experiments, we compared three label smoothing strategies: Label Smoothing (LS), Two-Stage Label smoothing (TSLA), and Linear Decay Label smoothing (LDLA). The initial value of ϵ in our experiments was 0.1, and in the first stage of TSLA, the number of epochs was set to 4. For each epoch in LDLA, ϵ was decreased by 0.01.

Retrieval Module As shown in Table 4, label smoothing effectively enhances the generalization performance of the retrieval module. LDLA outperforms TSLA with a higher EM (0.36%) and F1 score (0.13%), demonstrating superior generalization capabilities.

Supporting Sentence Prediction We assess the impact of label smoothing on the supporting sentence prediction task. The results presented in Table 5 indicate that TSLA exhibits an increase of 0.48% in EM and 0.22% in F1 compared to the baseline. Additionally, LDLA further enhances the performance by 0.21% in EM and 0.13% in F1 when compared to TSLA.

Answer Span Extraction Table 6 highlights the impact of label smoothing methods on answer span extraction in the reading comprehension module. LS, TSLA, and LDLA exhibit slight improvements compared to the baseline. The advanced Word Overlapping technique demonstrates an average improvement of 0.49% in EM and 0.47% in F1, respectively, compared to the baseline. In contrast, our proposed F1 Smoothing technique achieves an average EM improvement of 0.82% and an average F1 score improvement of 0.84%. These results suggest that F1 Smoothing can enhance performance on MRC tasks more effectively than other smoothing techniques.

5.5 Error Analysis

To gain a deeper understanding of how label smoothing effectively enhances model performance, we examined the model's output on the validation set, focusing on answer span errors and multi-hop reasoning errors. First, we define these two types of errors as follows:

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

CCL 2023

Methods	EM	F1
Baseline	69.11±.02	82.21±.03
LS	69.30±.02	82.56±.09
TSLA	69.32±.10	82.66±.09
LDLA	69.39±.12	82.69±.03
Word Overlapping	69.60±.09	82.68±.13
F1 Smoothing	**69.93**±.07	**83.05**±.10

Table 6: Analysis of different label smoothing methods for Answer Span Extraction.

Model	Answer Span Errors	Multi-Hop Reasoning Errors
S2G	1612	550
$\mathbf{R}^3$	1556	562
Smoothing $\mathbf{R}^3$	1536 (↓ 1.3%)	545(↓ 3.0%)

Table 7: Error analysis on Answer Span Errors and Multi-hop Reasoning Errors.

- Answer Span Errors: The predicted answer and the annotated answer have a partial overlap after removing stop words, but are not identical.

- Multi-hop Reasoning Errors: Due to reasoning errors, the predicted answer and the annotated answer are entirely different.

By implementing label smoothing, as shown in Table 7, Smoothing $\mathbf{R}^3$ experienced a 1.3% reduction in answer span errors, decreasing from 1556 to 1536, and a 3.0% decrease in multi-hop reasoning errors, dropping from 562 to 545. Smoothing $\mathbf{R}^3$ shows a significant reduction in both types of errors compared to the S2G model. This finding suggests that incorporating label smoothing during training can effectively prevent the model from overfitting the answer span and reasoning paths in the training set, thereby improving the model's generalization capabilities and overall performance.

6 Conclusion

In this study, we first identify the primary challenges hindering the performance of MHQA systems and propose using label smoothing to mitigate overfitting issues during MHQA training. We introduce F1 smoothing, a novel smoothing method inspired by the widely-used F1 score in MRC tasks. Additionally, we present LDLA, a progressive label smoothing algorithm that incorporates the concept of curriculum learning. Comprehensive experiments on the HotpotQA dataset demonstrate that our proposed model, Smoothing $\mathbf{R}^3$, achieves significant performance improvement when using F1 smoothing and LDLA. Our findings indicate that label smoothing is a valuable technique for MHQA, effectively improving the model's generalization while minimizing overfitting to particular patterns in the training set.

Acknowledgement

We would like to express our heartfelt thanks to the students and teachers of Fudan Natural Language Processing Lab. Their thoughtful suggestions, viewpoints, and enlightening discussions have made significant contributions to this work. We also greatly appreciate the strong support from Huawei Poisson Lab for our work, and their invaluable advice. We are sincerely grateful to the anonymous reviewers and the domain chairs, whose constructive feedback played a crucial role in enhancing the quality of our research. This work was supported by the National Key Research and Development Program of China (No.2022CSJGG0801), National Natural Science Foundation of China (No.62022027) and CAAI-Huawei MindSpore Open Fund.

CCL 2023

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *INTERSPEECH*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

Yingbo Gao, Weiyue Wang, Christian Herold, Zijian Yang, and Hermann Ney. 2020. Towards a better understanding of label smoothing in neural machine translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 212–223, Suzhou, China. Association for Computational Linguistics.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52, Florence, Italy. Association for Computational Linguistics.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. 2020. A simple yet strong pipeline for HotpotQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8839–8845, Online. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *ArXiv preprint*, abs/2111.09543.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Ronghan Li, Lifang Wang, Shengli Wang, and Zejun Jiang. 2021. Asynchronous multi-grained graph network for interpretable multi-hop reading comprehension. In *IJCAI*, pages 3857–3863.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2022. From easy to hard: Two-stage selector and reader for multi-hop question answering. *ArXiv preprint*, abs/2205.11729.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Krishna Menon, and Sanjiv Kumar. 2020a. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Michal Lukasik, Himanshu Jain, Aditya Menon, Seungyeon Kim, Srinadh Bhojanapalli, Felix Yu, and Sanjiv Kumar. 2020b. Semantic label smoothing for sequence to sequence problems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4992–4998, Online. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705.

CCL 2023

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.

Gustavo Penha and Claudia Hauff. 2021. Weakly supervised label smoothing. In *European Conference on Information Retrieval*, pages 334–341. Springer.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.

Sougata Saha, Souvik Das, and Rohini Srihari. 2021. Similarity based label smoothing for dialogue generation. *ArXiv preprint*, abs/2107.11481.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Is Graph Structure Necessary for Multi-hop Question Answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7187–7192, Online. Association for Computational Linguistics.

Lixin Su, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2020. Label distribution augmented maximum likelihood estimation for reading comprehension. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 564–572. ACM.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021. Graph-free multi-hop reading comprehension: A select-to-guide strategy. *ArXiv preprint*, abs/2107.11823.

Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, and Rong Jin. 2020. Towards understanding label smoothing. *ArXiv preprint*, abs/2006.11653.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhenyu Zhao, Shuangzhi Wu, Muyun Yang, Kehai Chen, and Tiejun Zhao. 2020. Robust machine reading comprehension by learning soft labels. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2754–2759, Barcelona, Spain (Online). International Committee on Computational Linguistics.

CCL 2023

Proceedings of the 22nd China National Conference on Computational Linguistics, pages 611-623, Harbin, China, August 3 - 5, 2023.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

7 Appendix A

In order to alleviate the complexity introduced by multiple for loops in the F1 Smoothing method, we have optimized Eq. (12) and Eq. (13). We use $L_a = e^* - s^* + 1$ and $L_p = e - s + 1$ to denote respectively the length of gold answer and predicted answer.

$$q_s(t|x) = \sum_{\xi=t}^{L-1} \text{F1}\left((t,\xi), a_{\text{gold}}\right). \tag{16}$$

If $t < s^*$, the distribution is

$$q_s(t|x) = \sum_{\xi=s^*}^{e^*} \frac{2(\xi - s^* + 1)}{L_p + L_a} + \sum_{\xi=e^*+1}^{L-1} \frac{2L_a}{L_p + L_a}, \tag{17}$$

else if $s^* \le t \le e^*$, we have the following distribution

$$q_s(t|x) = \sum_{\xi=s}^{e^*} \frac{2L_p}{L_p + L_a} + \sum_{\xi=e^*+1}^{L-1} \frac{2(e^* - s + 1)}{L_p + L_a}. \tag{18}$$

In equation 17 and 18, $L_p = e - i + 1$.

We can get $q_e(t|x)$ similarly. If $t > e^*$,

$$q_e(t|x) = \sum_{\xi=s^*}^{e^*} \frac{2(e^* - \xi + 1)}{L_p + L_a} + \sum_{\xi=0}^{s^*-1} \frac{2L_a}{L_p + L_a}, \tag{19}$$

else if $s^* \le t \le e^*$,

$$q_e(t|x) = \sum_{\xi=s^*}^{e} \frac{2L_p}{L_p + L_a} + \sum_{\xi=0}^{s^*-1} \frac{2(e - s^* + 1)}{L_p + L_a}. \tag{20}$$

In equation 19 and 20, $L_p = i - s + 1$.

CCL 2023