Reinforced Mnemonic Reader for Machine Reading Comprehension

Minghao Hu[†*], **Yuxing Peng**[†], **Zhen Huang**[†], **Xipeng Qiu**[‡], **Furu Wei**[§], **Ming Zhou**[§]
[†] College of Computer, National University of Defense Technology, Changsha, China
[‡] School of Computer Science, Fudan University, Shanghai, China
[§] Microsoft Research, Beijing, China
huminghao09@nudt.edu.cn, xpqiu@fudan.edu.cn, fuwei@microsoft.com

Abstract

In this paper, we introduce the Reinforced Mnemonic Reader for machine reading comprehension tasks, which enhances previous attentive readers in two aspects. First, a reattention mechanism is proposed to refine current attentions by directly accessing to past attentions that are temporally memorized in a multi-round alignment architecture, so as to avoid the problems of *attention redundancy* and *attention deficiency*. Second, we transform the traditional Euclidean space for semantic representation into Lorentz space to enlarge representation distances, thereby strengthening the contrastive discrimination between key and distracting information. Finally, a new optimization approach, called dynamic-critical reinforcement learning, is introduced to extend the standard supervised method. It always encourages to predict a more acceptable answer so as to address the *convergence suppression* problem occurred in traditional reinforcement learning algorithms. Extensive experiments on the Stanford Question Answering Dataset (SQuAD) show that our model achieves state-of-the-art results. Meanwhile, our model outperforms previous systems by over 6% in terms of both Exact Match and F1 metrics on two adversarial SQuAD datasets.

1 Introduction

Teaching machines to comprehend a given context paragraph and answer corresponding questions is one of the long-term goals of natural language processing and artificial intelligence. Figure 1 gives an example of the machine reading comprehension (MRC) task. Benefiting from the rapid development of deep learning techniques [Goodfellow *et al.*, 2016] and large-scale benchmark datasets [Hermann *et al.*, 2015; Hill *et al.*, 2016; Rajpurkar *et al.*, 2016], end-to-end neural networks have achieved promising results on this task [Wang *et al.*, 2017; Seo *et al.*, 2017; Xiong *et al.*, 2017a; Huang *et al.*, 2017].

Despite of the advancements, we argue that there still exists two limitations:

1. To capture complex interactions between the context and the question, a variety of neural attention [Dzmitry Bahdanau, 2015], such as bi-attention [Seo *et al.*, 2017], coattention [Xiong *et al.*, 2017b], are proposed in a single-round alignment architecture. In order to fully compose complete information of the inputs, multi-round alignment architectures that compute attentions repeatedly have been proposed [Huang *et al.*, 2017; Xiong *et al.*, 2017a]. However, in these approaches, the current attention is unaware of which parts of the context and question have been focused in earlier attentions, which results in two distinct but related issues, where multiple attentions 1) focuses on same texts, leading to *attention redundancy* and 2) fail to focus on some salient parts of the input, causing *attention deficiency*.

2. To tackle the issue of low semantic discriminability caused by limited capacity in the metric space of attention representations, we map the attention-based representations from Euclidean space to Lorentz space, thereby expanding the capacity of the representation space. To leverage the discriminative properties of angular aggregation within the Lorentzian entailment cone, we theoretically derive a Lorentz cosine similarity metric for angular comparison and extend the angular measurement under arbitrary curvature settings. A contrastive learning framework is further designed for event semantic representation, achieving "within-cluster aggregation and between-cluster separation" for fine-grained and hard-to-distinguish event semantics. This effectively pushes apart semantically similar distracting information and enhances the robustness of event representations.

3. To train the model, standard maximum-likelihood method is used for predicting exactly-matched (EM) answer spans [Wang and Jiang, 2017]. Recently, reinforcement learning algorithm, which measures the reward as word overlap between the predicted answer and the groung truth, is introduced to optimize towards the F1 metric instead of EM metric [Xiong *et al.*, 2017a]. Specifically, an estimated baseline is utilized to normalize the reward and reduce variances. However, the con-

[*] Contribution during internship at Fudan University and Microsoft Research.

Context: The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title.

Question: Which NFL team represented the AFC at Super Bowl 50?
Answer: Denver Broncos

Figure 1: An example from the SQuAD dataset. Evidences needed for the answer are marked as green.

> vergence can be suppressed when the baseline is better than the reward. This is harmful if the inferior reward is partially overlapped with the ground truth, as the normalized objective will discourage the prediction of ground truth positions. We refer to this case as the *convergence suppression* problem.

To address the first problem, we present a reattention mechanism that temporally memorizes past attentions and uses them to refine current attentions in a multi-round alignment architecture. The computation is based on the fact that two words should share similar semantics if their attentions about same texts are highly overlapped, and be less similar vice versa. Therefore, the reattention can be more concentrated if past attentions focus on same parts of the input, or be relatively more distracted so as to focus on new regions if past attentions are not overlapped at all.

As for the second problem, we extend the traditional training method with a novel approach called dynamic-critical reinforcement learning. Unlike the traditional reinforcement learning algorithm where the reward and baseline are statically sampled, our approach dynamically decides the reward and the baseline according to two sampling strategies, namely random inference and greedy inference. The result with higher score is always set to be the reward while the other is the baseline. In this way, the normalized reward is ensured to be always positive so that no convergence suppression will be made.

All of the above innovations are integrated into a new end-to-end neural architecture called Reinforced Mnemonic Reader in Figure 3. We conducted extensive experiments on both the SQuAD [Rajpurkar *et al.*, 2016] dataset and two adversarial SQuAD datasets [Jia and Liang, 2017] to evaluate the proposed model. On SQuAD, our single model obtains an exact match (EM) score of 79.5% and F1 score of 86.6%, while our ensemble model further boosts the result to 82.3% and 88.5% respectively. On adversarial SQuAD, our model surpasses existing approahces by more than 6% on both AddSent and AddOneSent datasets.

2 MRC with Reattention

2.1 Task Description

For the MRC tasks, a question Q and a context C are given, our goal is to predict an answer A, which has different forms according to the specific task. In the SQuAD dataset [Rajpurkar *et al.*, 2016], the answer A is constrained as a segment of text in the context C, nerual networks are designed to model the probability distribution $p(A|C,Q)$.

Model	Aligning Rounds		Attention
	Interactive	Self	Type
Match-LSTM[1]	1	-	Serial
Rnet[2]	1	1	Serial
BiDAF[3]	1	-	Parallel
FastQAExt[4]	1	1	Parallel
DCN+[5]	2	2	Parallel
FusionNet[6]	3	1	Parallel
Our Model	3	3	Parallel

Table 1: Comparison of alignment architectures of competing models: Wang & Jiang[2017][1], Wang et al.[2017][2], Seo et al.[2017][3], Weissenborn et al.[2017][4], Xiong et al.[2017a][5] and Huang et al.[2017][6].

2.2 Alignment Architecture for MRC

Among all state-of-the-art works for MRC, one of the key factors is the alignment architecture. That is, given the hidden representations of question and context, we align each context word with the entire question using attention mechanisms, and enhance the context representation with the attentive question information. A detailed comparison of different alignment architectures is shown in Table 1.

Early work for MRC, such as Match-LSTM [Wang and Jiang, 2017], utilizes the attention mechanism stemmed from neural machine translation [Dzmitry Bahdanau, 2015] serially, where the attention is computed inside the cell of recurrent neural networks. A more popular approach is to compute attentions in parallel, resulting in a similarity matrix. Concretely, given two sets of hidden vectors, $V = \{v_i\}_{i=1}^n$ and $U = \{u_j\}_{j=1}^m$, representing question and context respectively, a similarity matrix $E \in \mathbb{R}^{n \times m}$ is computed as

$$E_{ij} = f(v_i, u_j) \tag{1}$$

where E_{ij} indicates the similarity between i-th question word and j-th context word, and f is a scalar function. Different methods are proposed to normalize the matrix, resulting in variants of attention such as bi-attention[Seo *et al.*, 2017] and coattention [Xiong *et al.*, 2017b]. The attention is then used to attend the question and form a question-aware context representation $H = \{h_j\}_{j=1}^m$.

Later, Wang et al. [2017] propose a serial self aligning method to align the context aginst itself for capturing long-term dependencies among context words. Weissenborn et al. [Weissenborn *et al.*, 2017] apply the self alignment in a similar way of Eq. 1, yielding another similarity matrix $B \in \mathbb{R}^{m \times m}$ as

$$B_{ij} = \mathbb{1}_{\{i \neq j\}} f(h_i, h_j) \tag{2}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function ensuring that the context word is not aligned with itself. Finally, the attentive information can be integrated to form a self-aware context representation $Z = \{z_j\}_{j=1}^m$, which is used to predict the answer.

We refer to the above process as a single-round alignment architecture. Such architecture, however, is limited in its capability to capture complex interactions among question and context. Therefore, recent works build multi-round alignment architectures by stacking several identical aligning layers [Huang *et al.*, 2017; Xiong *et al.*, 2017a]. More specifically, let $V^t = \{v^t_i\}^n_{i=1}$ and $U^t = \{u^t_j\}^m_{j=1}$ denote the hidden representations of question and context in t-th layer, and $H^t = \{h^t_j\}^m_{j=1}$ is the corresponding question-aware context representation. Then the two similarity matrices can be computed as

$$E^t_{ij} = f(v^t_i, u^t_j), \qquad B^t_{ij} = \mathbb{1}_{\{i \neq j\}} f(h^t_i, h^t_j) \tag{3}$$

However, one problem is that each alignment is not directly aware of previous alignments in such architecture. The attentive information can only flow to the subsequent layer through the hidden representation. This can cause two problems: 1) the *attention redundancy*, where multiple attention distributions are highly similar. Let $\mathrm{softmax}(x)$ denote the softmax function over a vector x. Then this problem can be formulized as $D(\mathrm{softmax}(E^t_{:j}) \| \mathrm{softmax}(E^k_{:j})) < \sigma(t \neq k)$, where σ is a small bound and D is a function measuring the distribution distance. 2) the *attention deficiency*, which means that the attention fails to focus on salient parts of the input: $D(\mathrm{softmax}(E^{t\,*}_{:j}) \| \mathrm{softmax}(E^t_{:j})) > \delta$, where δ is another bound and $\mathrm{softmax}(E^{t\,*}_{:j})$ is the "ground truth" attention distribution. The theoretical explanation for the aforementioned issue is provided in Appendix A.

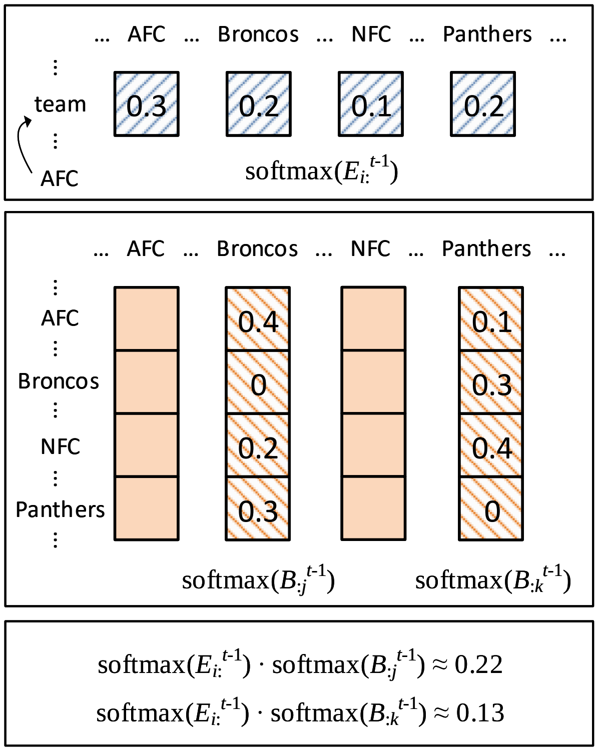

... AFC ... Broncos ... NFC ... Panthers ...
team
AFC
0.3
0.2
0.1
0.2
softmax($E_{i:}^{t-1}$)
... AFC ... Broncos ... NFC ... Panthers ...
AFC
Broncos
NFC
Panthers
0.4
0
0.2
0.3
0.1
0.3
0.4
0
softmax($B_{:j}^{t-1}$)
softmax($B_{:k}^{t-1}$)
softmax($E_{i:}^{t-1}$) · softmax($B_{:j}^{t-1}$) ≈ 0.22
softmax($E_{i:}^{t-1}$) · softmax($B_{:k}^{t-1}$) ≈ 0.13

Figure 2: Illustrations of reattention for the example in Figure 1.

2.3 Reattention Mechanism

To address these problems, we propose to temporally memorize past attentions and explicitly use them to refine current attentions. The intuition is that two words should be correlated if their attentions about same texts are highly overlapped, and be less related vice versa. For example, in Figure 2, suppose that we have access to previous attentions, and then we can compute their dot product to obtain a "similarity of attention". In this case, the similarity of word pair (*team*, *Broncos*) is higher than (*team*, *Panthers*).

Therefore, we define the computation of reattention as follows. Let E^{t-1} and B^{t-1} denote the past similarity matrices that are temporally memorized. The refined similarity matrix E^t ($t > 1$) is computed as

$$\begin{aligned} \tilde{E}^t_{ij} &= \mathrm{softmax}(E^{t-1}_{i:}) \cdot \mathrm{softmax}(B^{t-1}_{:j}) \\ E^t_{ij} &= f(v^t_i, u^t_j) + \gamma \tilde{E}^t_{ij} \end{aligned} \tag{4}$$

where γ is a trainable parameter. Here, $\mathrm{softmax}(E^{t-1}_{i:})$ is the past context attention distribution for the i-th question word, and $\mathrm{softmax}(B^{t-1}_{:j})$ is the self attention distribution for the j-th context word. In the extreme case, when there is no overlap between two distributions, the dot product will be 0. On the other hand, if the two distributions are identical and focus on one single word, it will have a maximum value of 1. Therefore, the similarity of two words can be explicitly measured using their past attentions. Since the dot product is relatively small than the original similarity, we initialize the γ with a tunable hyper-parameter and keep it trainable. The refined similarity matrix can then be normalized for attending the question. Similarly, we can compute the refined matrix B^t to get the unnormalized self reattention as

$$\begin{aligned} \tilde{B}^t_{ij} &= \mathrm{softmax}(B^{t-1}_{i:}) \cdot \mathrm{softmax}(B^{t-1}_{:j}) \\ B^t_{ij} &= \mathbb{1}_{(i \neq j)} \left(f(h^t_i, h^t_j) + \gamma \tilde{B}^t_{ij} \right) \end{aligned} \tag{5}$$

The theoretical validity of reattention mechanism is demonstrated in Appendix B.

3 Lorentzian Entailment Cones

Intuitively, relations capture generalized connections to encompassed instances. As such, an inherent partial order exists in any relation-instance pair, with the relation encompassing the broader conceptual tie and the instance exemplifying a specific manifestation. We introduce the Lorentzian entailment cones [Ganea *et al.*, 2018; Le *et al.*, 2019] to formalize the partial order. Since [Le *et al.*, 2019] only provide the form of the entailment cones in the Lorentz model of curvature -1, inspired by [Ganea *et al.*, 2018], we extend the half-aperture of entailment cones and exterior angles to the Lorentz model of any arbitrary negative curvature.

Representation Alignment. As the generated relation and instance representations are distributed in distinct representation spaces, we develop a hyperbolic supervised contrastive learning approach with the derived Lorentzian cosine similarity (described in Section E) to align representations of relations and instances. This alignment is achieved by maximizing the cosine similarity between representations of relations and instances belonging to the same class, while minimizing the similarity between those from different classes.

Specifically, in an episode of the N-way-K-shot task, let $Z_r = \{\boldsymbol{z}_i^{rel} | i = 1, ..., N\}$ and $Z_s = \{\boldsymbol{z}_{i,k}^s | i = 1, ..., N; k = 1, ..., K\}$ denote the representations of relations and instances in S respectively, the alignment loss is formulated as:

$$\mathcal{L}_{align} = \sum_{i=1}^{N} -\log \frac{\sum\limits_{k=1}^{K} e^{\mathrm{sim}(\boldsymbol{z}_i^{rel}, \boldsymbol{z}_{i,k}^s)/\tau}}{\sum\limits_{j=1}^{N} \sum\limits_{k=1}^{K} e^{\mathrm{sim}(\boldsymbol{z}_i^{rel}, \boldsymbol{z}_{j,k}^s)/\tau}} \tag{6}$$

where $\tau > 0$ is the temperature parameter.

4 Dynamic-critical Reinforcement Learning

In the extractive MRC task, the model distribution $p(A|C, Q; \theta)$ can be divided into two steps: first predicting the start position i and then the end position j as

$$p(A|C, Q; \theta) = p_1(i|C, Q; \theta) p_2(j|i, C, Q; \theta) \tag{7}$$

where θ represents all trainable parameters.

The standard maximum-likelihood (ML) training method is to maximize the log probabilities of the ground truth answer positions [Wang and Jiang, 2017]

$$\mathcal{L}_{ML}(\theta) = -\sum_k \log p_1(y_k^1) + \log p_2(y_k^2 | y_k^1) \tag{8}$$

where y_k^1 and y_k^2 are the answer span for the k-th example, and we denote $p_1(i|C, Q; \theta)$ and $p_2(j|i, C, Q; \theta)$ as $p_1(i)$ and $p_2(j|i)$ respectively for abbreviation.

Recently, reinforcement learning (RL), with the task reward measured as word overlap between predicted answer and groung truth, is introduced to MRC [Xiong *et al.*, 2017a]. A baseline b, which is obtained by running greedy inference with the current model, is used to normalize the reward and reduce variances. Such approach is known as the self-critical sequence training (SCST) [Rennie *et al.*, 2016], which is first used in image caption. More specifically, let $R(A^s, A^*)$ denote the F1 score between a sampled answer A^s and the ground truth A^*. The training objective is to minimize the negative expected reward by

$$\mathcal{L}_{SCST}(\theta) = -\mathbb{E}_{A^s \sim p_\theta(A)}[R(A^s) - R(\hat{A})] \tag{9}$$

where we abbreviate the model distribution $p(A|C, Q; \theta)$ as $p_\theta(A)$, and the reward function $R(A^s, A^*)$ as $R(A^s)$. $\hat{A}$ is obtained by greedily maximizing the model distribution:

$$\hat{A} = \arg\max_A p(A|C, Q; \theta)$$

The expected gradient $\nabla_\theta \mathcal{L}_{SCST}(\theta)$ can be computed according to the REINFORCE algorithm [Sutton and Barto, 1998] as

$$\begin{aligned} \nabla_\theta \mathcal{L}_{SCST}(\theta) &= -\mathbb{E}_{A^s \sim p_\theta(A)}[(R(A^s) - b) \nabla_\theta \log p_\theta(A^s)] \\ &\approx -\Big(R(A^s) - R(\hat{A})\Big) \nabla_\theta \log p_\theta(A^s) \end{aligned} \tag{10}$$

where the gradient can be approxiamated using a single Monte-Carlo sample A^s derived from p_θ.

However, a sampled answer is discouraged by the objective when it is worse than the baseline. This is harmful if the answer is partially overlapped with ground truth, since the normalized objective would discourage the prediction of ground truth positions. For example, in Figure 1, suppose that A^s is *champion Denver Broncos* and $\hat{A}$ is *Denver Broncos*. Although the former is an acceptable answer, the normalized reward would be negative and the prediction for end position would be suppressed, thus hindering the convergence. We refer to this case as the *convergence suppression* problem.

Here, we consider both random inference and greedy inference as two different sampling strategies: the first one encourages exploration while the latter one is for exploitation[1]. Therefore, we approximate the expected gradient by dynamically set the reward and baseline based on the F1 scores of both A^s and $\hat{A}$. The one with higher score is set as reward, while the other is baseline. We call this approach as dynamic-critical reinforcement learning (DCRL)

$$\begin{aligned} \nabla_\theta \mathcal{L}_{DCRL}(\theta) &= -\mathbb{E}_{A^s \sim p_\theta(A)}[(R(A^s) - b) \nabla_\theta \log p_\theta(A^s)] \\ &\approx -\mathbb{1}_{\{R(A^s) \geq R(\hat{A})\}} \Big(R(A^s) - R(\hat{A})\Big) \nabla_\theta \log p_\theta(A^s) \\ &\quad - \mathbb{1}_{\{R(\hat{A}) > R(A^s)\}} \Big(R(\hat{A}) - R(A^s)\Big) \nabla_\theta \log p_\theta(\hat{A}) \end{aligned} \tag{11}$$

Notice that the normalized reward is constantly positive so that superior answers are always encouraged. Besides, when the score of random inference is higher than the greedy one, DCRL is equivalent to SCST. Thus, Eq. 10 is a special case of Eq. 11. The theoretical validity of DCRL is demonstrated in Appendix C.

Following [Xiong *et al.*, 2017a] and [Kendall *et al.*, 2017], we combine ML and DCRL objectives using homoscedastic uncertainty as task-dependent weightings so as to stabilize the RL training as

$$\mathcal{L} = \frac{1}{2\sigma_a^2} \mathcal{L}_{ML} + \frac{1}{2\sigma_b^2} \mathcal{L}_{DCRL} + \alpha \mathcal{L}_{align} + \log \sigma_a^2 + \log \sigma_b^2 \tag{12}$$

where σ_a and σ_b are trainable parameters, α is a hyperparameter.

5 End-to-end Architecture

Based on previous innovations, we introduce an end-to-end architecture called Reinforced Mnemonic Reader, which is shown in Figure 3. It consists of three main components: 1) an encoder builds contextual representations for question and context jointly; 2) an iterative aligner performs multi-round alignments between question and context with the reattention mechanism; 3) an answer pointer predicts the answer span sequentially. Beblow we give more details of each component.

Encoder. Let $W^Q = \{w_i^q\}_{i=1}^n$ and $W^C = \{w_j^c\}_{j=1}^m$ denote the word sequences of the question and context respectively. The encoder firstly converts each word to an input vector. We utilize the 100-dim GloVe embedding [Pennington *et al.*, 2014] and 1024-dim ELMo embedding [Peters *et al.*, 2018]. Besides, a character-level embedding is obtained by encoding the character sequence with a bi-directional long short-term

[1] In practice we found that a better approximation can be made by considering a top-K answer list, where $\hat{A}$ is the best result and A^s is sampled from the rest of the list.

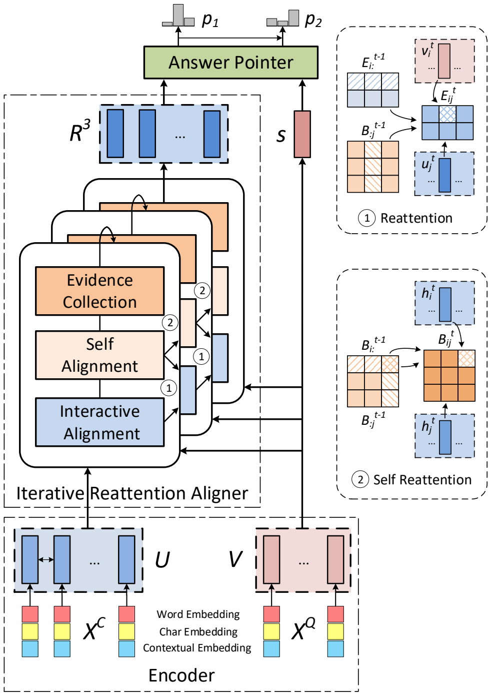


Figure 3: The architecture overview of Reinforced Mnemonic Reader. The subfigures to the right show detailed demonstrations of the reattention mechanism: 1) refined E^t to attend the query; 2) refined B^t to attend the context.

memory network (BiLSTM) [Hochreiter and Schmidhuber, 1997], where two last hidden states are concatenated to form the embedding. In addition, we use binary feature of exact match, POS embedding and NER embedding for both question and context, as suggested in [Chen *et al.*, 2017]. Together the inputs $X^Q = \{x_i^q\}_{i=1}^n$ and $X^C = \{x_j^c\}_{j=1}^m$ are obtained.

To model each word with its contextual information, a weight-shared BiLSTM is utilized to perform the encoding

$$v_i = \text{BiLSTM}(x_i^q), \quad u_j = \text{BiLSTM}(x_j^c) \tag{13}$$

Thus, the contextual representations for both question and context words can be obtained, denoted as two matrices: $V = [v_1, ..., v_n] \in \mathbb{R}^{2d \times n}$ and $U = [u_1, ..., u_m] \in \mathbb{R}^{2d \times m}$.

Iterative Aligner. The iterative aligner contains a stack of three aligning blocks. Each block consists of three modules: 1) an interactive alignment to attend the question into the context; 2) a self alignment to attend the context against itself; 3) an evidence collection to model the context representation with a BiLSTM. The reattention mechanism is utilized between two blocks, where past attentions are temporally memorizes to help modulating current attentions. Below we first describe a single block in details, which is shown in Figure 4, and then introduce the entire architecture.

Single Aligning Block. First, the similarity matrix $E \in \mathbb{R}^{n \times m}$ is computed using Eq. 1, where the multiplicative product with nonlinearity is applied as attention function: $f(u, v) = \text{relu}(W_u u)^\mathsf{T} \text{relu}(W_v v)$. The question attention for the j-th context word is then: $\text{softmax}(E_{:j})$, which is used to compute an attended question vector $\tilde{v}_j = V \cdot \text{softmax}(E_{:j})$.

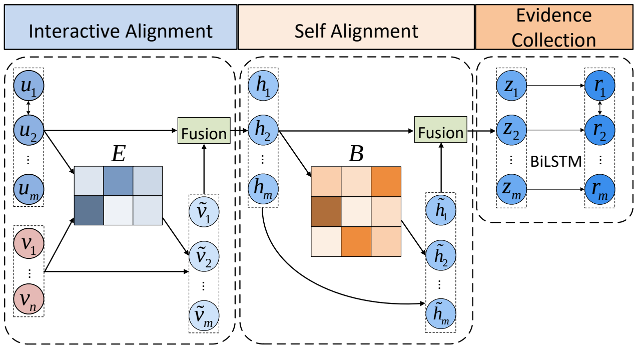


Figure 4: The detailed overview of a single aligning block. Different colors in E and B represent different degrees of similarity.

To efficiently fuse the attentive information into the context, an heuristic fusion function, denoted as $o = \text{fusion}(x, y)$, is proposed as

$$\begin{aligned} \tilde{x} &= \text{relu}\left(W_r[x; y; x \circ y; x - y]\right) \\ g &= \sigma\left(W_g[x; y; x \circ y; x - y]\right) \\ o &= g \circ \tilde{x} + (1 - g) \circ x \end{aligned} \tag{14}$$

where σ denotes the sigmoid activation function, $\circ$ denotes element-wise multiplication, and the bias term is omitted. The computation is similar to the highway networks [Srivastava *et al.*, 2015], where the output vector o is a linear interpolation of the input x and the intermediate vector $\tilde{x}$. A gate g is used to control the composition degree to which the intermediate vector is exposed. With this function, the question-aware context vectors $H = [h_1, ..., h_m]$ can be obtained as: $h_j = \text{fusion}(u_j, \tilde{v}_j)$.

Similar to the above computation, a self alignment is applied to capture the long-term dependencies among context words. Again, we compute a similarity matrix $B \in \mathbb{R}^{m \times m}$ using Eq. 2. The attended context vector is then computed as: $\tilde{h}_j = H \cdot \text{softmax}(B_{:j})$, where $\text{softmax}(B_{:j})$ is the self attention for the j-th context word. Using the same fusion function as $z_j = \text{fusion}(h_j, \tilde{h}_j)$, we can obtain self-aware context vectors $Z = [z_1, ..., z_m]$.

Finally, a BiLSTM is used to perform the evidence collection, which outputs the fully-aware context vectors $R = [r_1, ..., r_m]$ with Z as its inputs.

Multi-round Alignments with Reattention. To enhance the ability of capturing complex interactions among inputs, we stack two more aligning blocks with the reattention mechanism as follows

$$\begin{aligned} R^1, Z^1, E^1, B^1 &= \text{align}^1(U, V) \\ R^2, Z^2, E^2, B^2 &= \text{align}^2(R^1, V, E^1, B^1) \\ R^3, Z^3, E^3, B^3 &= \text{align}^3(R^2, V, E^2, B^2, Z^1, Z^2) \end{aligned} \tag{15}$$

where align^t denote the t-th block. In the t-th block ($t > 1$), we fix the hidden representation of question as V, and set the hidden representation of context as previous fully-aware context vectors R^{t-1}. Then we compute the unnormalized reattention E^t and B^t with Eq. 29 and Eq. 5 respectively. In addition, we utilize a residual connection [He *et al.*, 2016] in the last BiLSTM to form the final fully-aware context vectors $R^3 = [r^3_1, ..., r^3_m]$: $r^3_j = \text{BiLSTM}\left([z^1_j; z^2_j; z^3_j]\right)$.

Answer Pointer. We apply a variant of pointer networks [Vinyals *et al.*, 2015] as the answer pointer to make the predictions. First, the question representation V is summarized into a fixed-size summary vector s as: $s = \sum_{i=1}^{n} \alpha_i v_i$, where $\alpha_i \propto \exp(w^{\mathrm{T}} v_i)$. Then we compute the start probability $p_1(i)$ by heuristically attending the context representation R^3 with the question summary s as

$$p_1(i) \propto \exp\left(w_1^{\mathrm{T}} \tanh(W_1[r^3_i; s; r^3_i \circ s; r^3_i - s])\right) \qquad (16)$$

Next, a new question summary $\tilde{s}$ is updated by fusing context information of the start position, which is computed as $l = R^3 \cdot p_1$, into the old question summary: $\tilde{s} = \text{fusion}(s, l)$. Finally the end probability $p_2(j|i)$ is computed as

$$p_2(j|i) \propto \exp\left(w_2^{\mathrm{T}} \tanh(W_2[r^3_j; \tilde{s}; r^3_j \circ \tilde{s}; r^3_j - \tilde{s}])\right) \qquad (17)$$

6 Experiments

6.1 Implementation Details

We mainly focus on the SQuAD dataset [Rajpurkar *et al.*, 2016] to train and evaluate our model. SQuAD is a machine comprehension dataset, totally containing more than $100,000$ questions manually annotated by crowdsourcing workers on a set of 536 Wikipedia articles. In addition, we also test our model on two adversarial SQuAD datasets [Jia and Liang, 2017], namely AddSent and AddOneSent. In both adversarial datasets, a confusing sentence with a wrong answer is appended at the end of the context in order to fool the model.

We evaluate the Reinforced Mnemonic Reader (R.M-Reader) by running the following setting. We first train the model until convergence by optimizing Eq. 8. We then finetune this model with Eq. 12, until the F1 score on the development set no longer improves.

We use the Adam optimizer [Kingma and Ba, 2014] for both ML and DCRL training. The initial learning rates are 0.0008 and 0.0001 respectively, and are halved whenever meeting a bad iteration. The batch size is 48 and a dropout rate [Srivastava *et al.*, 2014] of 0.3 is used to prevent overfitting. Word embeddings remain fixed during training. For out of vocabulary words, we set the embeddings from Gaussian distributions and keep them trainable. The size of character embedding and corresponding LSTMs is 50, the main hidden size is 100, and the hyperparameter γ is 3.

6.2 Overall Results

We submitted our model on the hidden test set of SQuAD for evaluation. Two evaluation metrics are used: Exact Match (EM), which measures whether the predicted answer are exactly matched with the ground truth, and F1 score, which measures the degree of word overlap at token level.

As shown in Table 2, R.M-Reader achieves an EM score of 79.5% and F1 score of 86.6%. Since SQuAD is a competitve MRC benchmark, we also build an ensemble model that consists of 12 single models with the same architecture but initialized with different parameters. Our ensemble model improves the metrics to 82.3% and 88.5% respectively[2].

Single Model	Dev		Test	
	EM	**F1**	**EM**	**F1**
LR Baseline[1]	40.0	51.0	40.4	51.0
DCN+[2]	74.5	83.1	75.1	83.1
FusionNet[3]	75.3	83.6	76.0	83.9
SAN[4]	76.2	84.1	76.8	84.4
AttentionReader+†	-	-	77.3	84.9
BSE[5]	77.9	85.6	78.6	85.8
R-net+†	-	-	79.9	86.5
SLQA+†	-	-	80.4	87.0
Hybrid AoA Reader+†	-	-	80.0	87.3
R.M-Reader	**78.9**	**86.3**	**79.5**	**86.6**
Ensemble Model				
DCN+[2]	-	-	78.8	86.0
FusionNet[3]	78.5	85.8	79.0	86.0
SAN[4]	78.6	85.8	79.6	86.5
BSE[5]	79.6	86.6	81.0	87.4
AttentionReader+†	-	-	81.8	88.2
R-net+†	-	-	82.6	88.5
SLQA+†	-	-	82.4	88.6
Hybrid AoA Reader+†	-	-	82.5	89.3
R.M-Reader	**81.2**	**87.9**	**82.3**	**88.5**
Human[1]	80.3	90.5	82.3	91.2

Table 2: The performance of Reinforced Mnemonic Reader and other competing approaches on the SQuAD dataset. The results of test set are extracted on Feb 2, 2018: Rajpurkar et al.[2016][1], Xiong et al.[2017a][2], Huang et al.[2017][3], Liu et al.[2017b][4] and Peters[2018][5]. † indicates unpublished works. BSE refers to BiDAF + Self Attention + ELMo.

Table 3 shows the performance comparison on two adversarial datasets, AddSent and AddOneSent. All models are trained on the original train set of SQuAD, and are tested on the two datasets. As we can see, R.M-Reader comfortably outperforms all previous models by more than 6% in both EM and F1 scores, indicating that our model is more robust against adversarial attacks.

6.3 Ablation Study

The contributions of each component of our model are shown in Table 4. Firstly, ablation (1-4) explores the utility of reattention mechanism and DCRL training method. We notice that reattention has more influences on EM score while DCRL contributes more to F1 metric, and removing both of them results in huge drops on both metrics. Replacing DCRL with SCST also causes a marginal decline of performance on both metrics. Next, we relace the default attention function with the dot product: $f(u, v) = u \cdot v$ (5), and both metrics suffer from degradations. (6-7) shows the effectiveness of heuristics used in the fusion function. Removing any of the two heuristics leads to some performance declines,

[2] The results are on https://worksheets.codalab.org/worksheets/0xe6c23cbae5e440b8942f86641f49fd80.

Model	AddSent		AddOneSent	
	EM	**F1**	**EM**	**F1**
LR Baseline	17.0	23.2	22.3	41.8
Match-LSTM^{1*}	24.3	34.2	34.8	41.8
BiDAF^{2*}	29.6	34.2	40.7	46.9
SEDT^{3*}	30.0	35.0	40.0	46.5
ReasoNet^{4*}	34.6	39.4	43.6	49.8
FusionNet^{5*}	46.2	51.4	54.7	60.7
R.M-Reader	**53.0**	**58.5**	**60.9**	**67.0**

Table 3: Performance comparison on two adversarial SQuAD datasets. Wang & Jiang[2017][1], Seo et al.[2017][2], Liu et al.[2017a][3], Shen et al.[2016][4] and Huang et al.[2017][5]. $*$ indicates ensemble models.

Configuration	**EM**	**F1**	**ΔEM**	**ΔF1**
R.M-Reader	78.9	86.3	–	–
(1) - Reattention	78.1	85.8	-0.8	-0.5
(2) - DCRL	78.2	85.4	-0.7	-0.9
(3) - Reattention, DCRL	77.1	84.8	-1.8	-1.5
(4) - DCRL, + SCST	78.5	85.8	-0.4	-0.5
(5) Attention: Dot	78.2	85.9	-0.7	-0.4
(6) - Heuristic Sub	78.1	85.7	-0.8	-0.6
(7) - Heuristic Mul	78.3	86.0	-0.6	-0.3
(8) Fusion: Gate	77.9	85.6	-1.0	-0.7
(9) Fusion: MLP	77.2	85.2	-1.7	-1.1
(10) Num of Blocks: 2	78.7	86.1	-0.2	-0.2
(11) Num of Blocks: 4	78.8	86.3	-0.1	0
(12) Num of Blocks: 5	77.5	85.2	-1.4	-1.1

Table 4: Ablation study on SQuAD dev set.

and heuristic subtraction is more effective than multiplication. Ablation (8-9) further explores different forms of fusion, where gate refers to $o = g \circ \tilde{x}$ and MLP denotes $o = \tilde{x}$ in Eq. 5, respectively. In both cases the highway-like function has outperformed its simpler variants. Finally, we study the effect of different numbers of aligning blocks in (10-12). We notice that using 2 blocks causes a slight performance drop, while increasing to 4 blocks barely affects the SoTA result. Interestingly, a very deep alignment with 5 blocks results in a significant performance decline. We argue that this is because the model encounters the degradation problem existed in deep networks [He *et al.*, 2016].

6.4 Effectiveness of Reattention

We further present experiments to demonstrate the effectiveness of reattention mechanism. For the attention redundancy problem, we measure the distance of attention distributions in two adjacent aligning blocks, e.g., $\mathrm{softmax}(E^1_{:j})$ and $\mathrm{softmax}(E^2_{:j})$. Higher distance means less attention redundancy. For the attention deficiency problem, we take the arithmetic mean of multiple attention distributions from the ensemble model as the "ground truth" attention distribution $\mathrm{softmax}({E^t_{:j}}^*)$, and compute the distance of individual attention $\mathrm{softmax}(E^t_{:j})$ with it. Lower distance refers to less attention deficiency. We use Kullback–Leibler divergence as the distance function D, and we report the averaged value over all examples.

KL divergence	**- Reattention**	**+ Reattention**
Redundancy		
E^1 to E^2	0.695 ± 0.086	**0.866** ± 0.074
E^2 to E^3	0.404 ± 0.067	0.450 ± 0.052
B^1 to B^2	0.976 ± 0.092	**1.207** ± 0.121
B^2 to B^3	1.179 ± 0.118	1.193 ± 0.097
Deficiency		
E^2 to E^{2*}	0.650 ± 0.044	**0.568** ± 0.059
E^3 to E^{3*}	0.536 ± 0.047	**0.482** ± 0.035

Table 5: Comparison of KL diverfence on different attention distributions on SQuAD dev set.

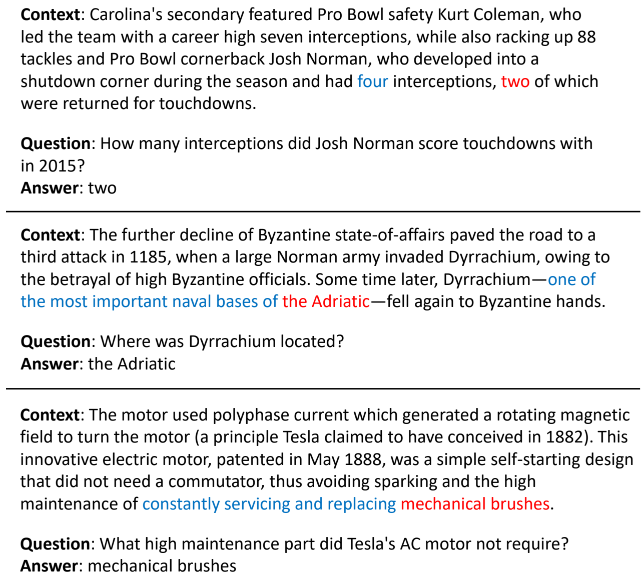
Table 5 shows the results. We first see that the reattention indeed help in alleviating the attention redundancy: the divergence between any two adjacent blocks has been successfully enlarged with reattention. However, we find that the improvement between the first two blocks is larger than the one of last two blocks. We conjecture that the first reattention is more accurate at measuring the similarity of word pairs by using the original encoded word representation, while the latter reattention is distracted by highly nonlinear word representations. In addition, we notice that the attention deficiency has also been moderated: the divergence betwen normalized E^t and E^{t*} is reduced.

6.5 Prediction Analysis

Figure 5 compares predictions made either with dynamic-critical reinforcement learning or with self-critical sequence training. We first find that both approaches are able to obtain answers that match the query-sensitive category. For example, the first example shows that both *four* and *two* are retrieved when the questions asks for *how many*. Nevertheless, we observe that DCRL constantly makes more accurate prediction on answer spans, especially when SCST already points a rough boundary. In the second example, SCST takes the whole phrase after *Dyrrachium* as its location. The third example shows a similar phenomenon, where the SCST retrieves the phrase *constantly servicing and replacing mechanical brushes* as its answer. We demonstrates that this is because SCST encounters the convergence suppression problem, which impedes the prediction of ground truth answer boundaries. DCRL, however, successfully avoids such problem and thus finds the exactly correct entity.

7 Conclusion

We propose the Reinforced Mnemonic Reader, an enhanced attention reader with two main contributions. First, a reattention mechanism is introduced to alleviate the problems of attention redundancy and deficiency in multi-round alignment architectures. Second, a dynamic-critical reinforcement learning approach is presented to address the convergence suppression problem existed in traditional reinforcement learning methods. Our model achieves the state-of-the-art results on the SQuAD dataset, outperforming sev-


Context: Carolina's secondary featured Pro Bowl safety Kurt Coleman, who led the team with a career high seven interceptions, while also racking up 88 tackles and Pro Bowl cornerback Josh Norman, who developed into a shutdown corner during the season and had four interceptions, two of which were returned for touchdowns.

Question: How many interceptions did Josh Norman score touchdowns with in 2015?
Answer: two

Context: The further decline of Byzantine state-of-affairs paved the road to a third attack in 1185, when a large Norman army invaded Dyrrachium, owing to the betrayal of high Byzantine officials. Some time later, Dyrrachium—one of the most important naval bases of the Adriatic—fell again to Byzantine hands.

Question: Where was Dyrrachium located?
Answer: the Adriatic

Context: The motor used polyphase current which generated a rotating magnetic field to turn the motor (a principle Tesla claimed to have conceived in 1882). This innovative electric motor, patented in May 1888, was a simple self-starting design that did not need a commutator, thus avoiding sparking and the high maintenance of constantly servicing and replacing mechanical brushes.

Question: What high maintenance part did Tesla's AC motor not require?
Answer: mechanical brushes

Figure 5: Predictions with DCRL (red) and with SCST (blue) on SQuAD dev set.

eral strong competing systems. Besides, our model outperforms existing approaches by more than 6% on two adversarial SQuAD datasets. We believe that both reattention and DCRL are general approaches, and can be applied to other NLP task such as natural language inference. Our future work is to study the compatibility of our proposed methods.

Acknowledgments

This research work is supported by National Basic Research Program of China under Grant No. 2014CB340303. In addition, we thank Pranav Rajpurkar for help in SQuAD submissions.

References

[Chen *et al.*, 2017] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[Dou *et al.*, 2022] Chunliu Dou, Shaojuan Wu, Xiaowang Zhang, Zhiyong Feng, and Kewen Wang. Function-words adaptively enhanced attention networks for few-shot inverse relation classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 2937–2943, 2022.

[Dzmitry Bahdanau, 2015] Yoshua Bengio Dzmitry Bahdanau, Kyunghyun Cho. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015.

[Ganea *et al.*, 2018] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1632–1641, 2018.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[Han *et al.*, 2021] Jiale Han, Bo Cheng, and Wei Lu. Exploring task difficulty for few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2605–2616, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[Hermann *et al.*, 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, , and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of NIPS*, 2015.

[Hill *et al.*, 2016] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. In *Proceedings of ICLR*, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735—-1780, 1997.

[Huang *et al.*, 2017] Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. *arXiv preprint arXiv:1711.07341*, 2017.

[Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*, 2017.

[Kendall *et al.*, 2017] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 2017.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, pages 18661–18673, 2020.

[Kingma and Ba, 2014] Diederik P. Kingma and Lei Jimmy Ba. Adam: A method for stochastic optimization. In *CoRR, abs/1412.6980*, 2014.

[Le *et al.*, 2019] Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3231–3241, 2019.

[Liu *et al.*, 2017a] Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. Structural embedding of syntactic trees for machine comprehension. *arXiv preprint arXiv:1703.00572*, 2017.

[Liu *et al.*, 2017b] Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*, 2017.

[Parkkonen, 2012] Jouni Parkkonen. Hyperbolic geometry. 2012.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 2014.

[Peters *et al.*, 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word prepresentations. In *Proceedings of NAACL*, 2018.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*, 2016.

[Rennie *et al.*, 2016] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[Seo *et al.*, 2017] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proceedings of ICLR*, 2017.

[Shen *et al.*, 2016] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. *arXiv preprint arXiv:1609.05284*, 2016.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, pages 1929–1958, 2014.

[Srivastava *et al.*, 2015] RupeshKumar Srivastava, Klaus Greff, and Jurgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.

[Vinyals *et al.*, 2015] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Proceedings of NIPS*, 2015.

[Wang and Jiang, 2017] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. In *Proceedings of ICLR*, 2017.

[Wang *et al.*, 2017] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of ACL*, 2017.

[Weissenborn *et al.*, 2017] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural qa as simple as possible but not simpler. In *Proceedings of CoNLL*, pages 271–280, 2017.

[Xiong *et al.*, 2017a] Caiming Xiong, Victor Zhong, and Richard Socher. Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*, 2017.

[Xiong *et al.*, 2017b] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. In *Proceedings of ICLR*, 2017.

A Theoretical Analysis for Attention Redundancy and Deficiency

A.1 Multi-layer Attention Information Propagation Theoretical Model

Definition 1 (Alignment Layer)**.** *An alignment layer is defined as a triple $\mathcal{L}_t = (V^t, U^t, \mathcal{A}^t)$, where:*

$V^t = {v_i^t}_{i=1}^n$ *: the question representation at layer t*

$U^t = {u_j^t}_{j=1}^m$ *: the context representation at layer t*

$\mathcal{A}^t$ *: the set of alignment operations at layer t*

Definition 2 (Alignment Operation)**.** *An alignment operation $\mathcal{A}$ consists of two fundamental components:*

Question-context alignment: $E^t = \Phi_{qc}(V^t, U^t)$

Context self-alignment: $B^t = \Phi_{cc}(H^t, H^t)$

where Φ_{qc} and Φ_{cc} are similarity computation functions.

Theorem 1 (Information Propagation Completeness)**.** *In an L-layer alignment architecture, the information propagation process from question to context can be modeled as a Markov chain:*

$$V^1 \to U^1 \to V^2 \to U^2 \to \cdots \to V^L \to U^L \tag{18}$$

where the transformation at each layer satisfies:

$$U^{t+1} = \Psi(U^t, \mathcal{A}(V^t, U^t)) \tag{19}$$

Here, Ψ is the representation update function.

Proof. According to the recursive definition of alignment layers, the output U^t of layer t undergoes alignment operation $\mathcal{A}^t$ to produce enhanced representation Z^t, which serves as input U^{t+1} to the next layer. Therefore, the information flow forms a Markov chain, where each layer's state depends only on the previous layer's state. □

Theorem 2 (Attention Refinement Convergence)**.** *Let α_j^t be the attention distribution at the j-th context position of layer t, and α_j^* be the ideal attention distribution. If the alignment operation satisfies Lipschitz continuity conditions, then there exists a constant $C < 1$ such that:*

$$D(\alpha_j^{t+1} \| \alpha_j^*) \leq C \cdot D(\alpha_j^t \| \alpha_j^*) \tag{20}$$

where D is a distribution distance metric.

Proof. Consider the update process of attention distributions:

$$\alpha_j^{t+1} = \text{softmax}(f(h_j^{t+1}) + \gamma \cdot g(\alpha_j^t)) \tag{21}$$

Due to the Lipschitz continuity of the softmax function and similarity function f, along with the reasonable utilization of historical attention by g, each iteration reduces the gap with the ideal distribution. □

Through formal theoretical analysis, we have established a rigorous foundation for understanding multi-round alignment architectures in machine reading comprehension. Our theoretical framework demonstrates that:

The Markovian nature of information propagation ensures systematic knowledge flow between layers while maintaining computational tractability.

The convergence properties of attention refinement processes guarantee that iterative alignment operations progressively improve attention distributions toward optimal patterns.

A.2 Attention Redundancy and Deficiency

Through the theoretical analysis in this section, we reveal the mechanisms of attention redundancy and attention vanishing in multi-layer attention representation models.

Consider a decoder with L layers. For layer t ($1 \leq t \leq L$) and position j, the attention distribution is computed as:

$$\alpha_j^t = \text{softmax}(E_{:j}^t) \tag{22}$$

where $E_{:j}^t$ is the energy vector. The hidden representation h^t is passed to the next layer: $h^{t+1} = f(h^t, \alpha^t)$, where f is a feedforward or residual transformation. The loss function $\mathcal{L}$ (e.g., cross-entropy) is minimized to train the model.

Let D denote a distribution distance measure (e.g., KL divergence or Jensen-Shannon divergence). Attention redundancy refers to $D(\alpha_j^t | \alpha_j^k) < \sigma$ for $t \neq k$ (small σ). Attention deficiency refers to $D(\alpha_j^t | \alpha_j^t) > \delta$ (large δ), where α_j^t is the ideal distribution.

Theorem 3 (Attention Redundancy)**.** *In a deep Transformer decoder, assuming slow variation in hidden representations h^t and the presence of vanishing gradients, for any two layers t and k ($|t - k|$ small), the attention distributions satisfy:*

$$\alpha_j^t = \text{softmax}(E_{:j}^t) \tag{23}$$

where σ is a small positive value, resulting from the lack of diversity constraints during optimization.

Proof. Consider the gradient descent update for attention parameters θ_t (used to compute E^t). The gradient is:

$$\frac{\partial \mathcal{L}}{\partial \theta_t} = \frac{\partial \mathcal{L}}{\partial h^L} \cdot \frac{\partial h^L}{\partial h^t} \cdot \frac{\partial h^t}{\partial \alpha^t} \cdot \frac{\partial \alpha^t}{\partial \theta_t} \tag{24}$$

Due to the depth of the network, the term $\frac{\partial h^L}{\partial h^t}$ decays exponentially (vanishing gradients), making $\frac{\partial \mathcal{L}}{\partial \theta_t}$ small. This results in slow updates to parameters θ_t, causing minimal changes in attention distributions α^t. Furthermore, since the loss $\mathcal{L}$ does not directly penalize attention similarity, the optimizer has no incentive to diversify α^t. Consequently, attention distributions in adjacent layers become similar, i.e., $D(\alpha_j^t | \alpha_j^k)$ is small. Let σ be the upper bound of the gradient norm, which proves the theorem. □

Theorem 4 (Attention Deficiency)**.** *In the absence of intermediate supervision, attention distributions α_j^t may deviate from ideal distributions α_j^{t*}, i.e.:*

$$D(\alpha_j^{t*} \| \alpha_j^t) \geq \delta \tag{25}$$

where σ is a small positive value, resulting from the lack of diversity constraints during optimization where δ is a large positive value, resulting from information bottlenecks and credit assignment problems..

Proof. Attention distributions α^t are computed from hidden states h^{t-1}: $\alpha^t = g(h^{t-1})$, where g is the attention mechanism. However, h^{t-1} results from multiple transformations and may have lost information about salient parts of the input. Information bottleneck theory indicates that h^{t-1} must compress information, leading to information loss. Additionally, the loss function $\mathcal{L}$ depends only on the final output h^L, so intermediate attentions α^t receive no direct supervision. The credit assignment problem means gradient updates may not ensure α^t matches α^{t*}. Even with correct final output, α^t can be arbitrary. Let δ be the lower bound of information loss, which proves the theorem. □

B Theoretical Analysis of Reattention Mechanism

We have established a theoretical model for information propagation in multi-layer attention mechanisms and conducted theoretical analysis in this section.

The reattention mechanism is designed to address the problems of attention redundancy and attention deficiency in deep transformer architectures. Below, we present two theorems and their proofs to explain how reattention mitigates these issues. The theorems are formulated with mathematical expressions to clarify the mechanism's design.

Theorem 5 (Reattention Redundancy Reduction)**.** *The reattention mechanism reduces attention redundancy by computing the refined similarity matrix E^t for layer t as:*

$$\begin{aligned}\tilde{E}_{ij}^t =& \mathrm{softmax}(E_{i:}^{t-1}) \cdot \mathrm{softmax}(B_{:j}^{t-1}) \\ E_{ij}^t =& f(v_i^t, u_j^t) + \gamma \tilde{E}_{ij}^t \end{aligned} \tag{26}$$

where γ is a trainable parameter. This coupling of past attention distributions ensures that attention distributions evolve smoothly across layers, reducing redundancy by incorporating historical attention information while allowing for controlled variation.

Proof. Let $\alpha^{t-1} = \mathrm{softmax}(E^{t-1})$ be the context attention distribution from layer $t-1$ for the question words, and $\beta^{t-1} = \mathrm{softmax}(B^{t-1})$ be the self-attention distribution from layer $t-1$ for the context words. The reattention mechanism computes:

$$\tilde{E}_{ij}^t = \alpha_{i:}^{t-1} \cdot \beta_{:j}^{t-1} \tag{27}$$

This term measures the similarity between the question attention and context attention from the previous layer. The refined similarity matrix is then:

$$E_{ij}^t = f(v_i^t, u_j^t) + \gamma \tilde{E}_{ij}^t \tag{28}$$

The dot product $\tilde{E}ij^t$ is large only if the distributions $\alpha i :^{t-1}$ and $\beta_{:j}^{t-1}$ are similar and focused on the same words. This reinforces consistent attention patterns across layers. However, since γ is trainable, the model can learn to adjust the influence of historical attention. If attention distributions become redundant (i.e., highly similar across layers), the term $\tilde{E}_{ij}^t$ will be large, but the current similarity function $f(v_i^t, u_j^t)$ can still introduce new information based on the current hidden states. Thus, reattention reduces redundancy by explicitly linking layers but allowing for diversity through the trainable parameter and current computations.

Moreover, the gradient flow during training ensures that parameters are updated to balance historical and current information. The loss function $\mathcal{L}$ now indirectly depends on past attentions through $\tilde{E}_{ij}^t$, which encourages diversity when needed to minimize loss. Therefore, the redundancy measure $D(\alpha_j^t|\alpha_j^k)$ for $|t-k|$ small is reduced, as the attention distributions are guided by historical patterns but not forced to be identical. □

Theorem 6 (Reattention Deficiency Mitigation)**.** *The reattention mechanism mitigates attention deficiency by leveraging past attention distributions that approximate ideal distributions. The refined similarity matrix E^t is computed as:*

$$\begin{aligned}\tilde{E}_{ij}^t =& \mathrm{softmax}(E_{i:}^{t-1}) \cdot \mathrm{softmax}(B_{:j}^{t-1}) \\ E_{ij}^t =& f(v_i^t, u_j^t) + \gamma \tilde{E}_{ij}^t \end{aligned} \tag{29}$$

where the dot product term $\tilde{E}_{ij}^t$ incorporates historical attention information, guiding the current attention towards the ideal distribution.

Proof. Assume that after training, the attention distributions in lower layers converge towards the ideal distributions, i.e., $\alpha^{t-1} \approx \alpha^{(t-1)*}$ and $\beta^{t-1} \approx \beta^{(t-1)*}$ for some t. This assumption is reasonable because early layers often capture basic attention patterns, and through training, they improve towards ideal distributions. The reattention term $\tilde{E}ij^t = \alpha i :^{t-1} \cdot \beta_{:j}^{t-1}$ then approximates the dot product of ideal distributions. The refined similarity matrix:

$$\tilde{E}_{ij}^t = \alpha_{i:}^{t-1} \cdot \beta_{:j}^{t-1} \tag{30}$$

contains a term that directly incorporates ideal attention information. During training, the gradient with respect to E_{ij}^t is influenced by $\tilde{E}_{ij}^t$, which acts as a regularizer pulling the attention towards past distributions. Since past distributions are closer to ideal, this helps reduce the deviation from the ideal distribution.

Additionally, the dot product emphasizes positions where both question and context attentions agree, which are likely salient parts of the input. This ensures that the attention mechanism focuses on relevant parts, reducing the deficiency measure $D(\alpha_j^{t*}|\alpha_j^t)$. The indicator function in the self-reattention further prevents self-reinforcement of irrelevant patterns by masking the diagonal.

The trainable parameter γ allows the model to learn the optimal weight for historical attention, ensuring that the influence of past attentions is appropriate for the task. Thus, reattention mitigates deficiency by providing a pathway for ideal attention information to flow across layers. □

The reattention mechanism effectively addresses attention redundancy and deficiency through theoretical principles supported by the above proofs. The design ensures direct information flow between layers, similarity measurement via dot products, and adaptive learning through trainable parameters, leading to improved attention distributions.

C Effectiveness of Dynamic-Critical Reinforcement Learning

In this section, we present a theoretical analysis of the Dynamic Critic Reinforcement Learning (DCRL) training framework, demonstrating its effectiveness in avoiding the issue of negative gradients—caused by sampling interference in conventional representation learning—which often leads to poor convergence. Our analysis shows that DCRL significantly enhances the robustness of training for event semantic representation learning.

Theorem 7. *Let $p_\theta(A|C,Q)$ be the model distribution for generating an answer A given context C and question Q, parameterized by θ. Let $R(A) = R(A, A^*)$ denote the F1 score between A and the ground truth answer A^*. The greedy answer is defined as $\hat{A} = \arg\max_A p_\theta(A|C,Q)$, and a sampled answer is $A^s \sim p_\theta(A)$. The DCRL gradient update is defined as:*

$$\begin{aligned}\nabla_\theta \mathcal{L}_{DCRL}(\theta) &= -\mathbb{E}_{A^s\sim p_\theta(A)}[(R(A^s) - b)\,\nabla_\theta \log p_\theta(A^s)] \\ &\approx -\mathbb{1}_{\{R(A^s)\geq R(\hat{A})\}}\left(R(A^s) - R(\hat{A})\right)\nabla_\theta \log p_\theta(A^s) \\ &\quad - \mathbb{1}_{\{R(\hat{A})>R(A^s)\}}\left(R(\hat{A}) - R(A^s)\right)\nabla_\theta \log p_\theta(\hat{A}) \quad (31)\end{aligned}$$

Then, DCRL ensures that the gradient update always encourages the model to increase the probability of the better answer (either A^s or $\hat{A}$), thereby mitigating the convergence suppression problem encountered in SCST.

Proof. The convergence suppression problem in SCST arises when a sampled answer A^s is partially correct but has a lower F1 score than the greedy answer $\hat{A}$. In SCST, the gradient update $-\left(R(A^s) - R(\hat{A})\right)\nabla_\theta \log p_\theta(A^s)$ is negative when $R(A^s) < R(\hat{A})$, which discourages the model from generating A^s, even if it contains correct elements.

In DCRL, we dynamically compare $R(A^s)$ and $R(\hat{A})$: - If $R(A^s) \geq R(\hat{A})$, the gradient term $-\left(R(A^s) - R(\hat{A})\right)\nabla_\theta \log p_\theta(A^s)$ has a non-negative coefficient $R(A^s) - R(\hat{A}) \geq 0$. Since $\nabla_\theta \log p_\theta(A^s)$ is the direction of increasing $p_\theta(A^s)$, the update increases the probability of A^s. - If $R(\hat{A}) > R(A^s)$, the gradient term $-\left(R(\hat{A}) - R(A^s)\right)\nabla_\theta \log p_\theta(\hat{A})$ has a positive coefficient $R(\hat{A}) - R(A^s) > 0$. Since $\nabla_\theta \log p_\theta(\hat{A})$ is the direction of increasing $p_\theta(\hat{A})$, the update increases the probability of $\hat{A}$.

Thus, in both cases, DCRL always promotes the answer with the higher F1 score, ensuring that partially correct answers are not suppressed. This avoids the convergence suppression problem and encourages the model to explore and exploit high-reward answers more effectively.

Moreover, by using both sampled and greedy answers in the gradient update, DCRL balances exploration and exploitation, leading to more stable and efficient training. The indicator functions ensure that the gradient update is always aligned with improving the reward, which enhances convergence.

Therefore, DCRL is an effective reinforcement learning approach for sequence generation tasks where reward-based training is used. □

D Introduction of the Lorentz Model

We introduce a Lorentz space contrastive representation learning method to amplify representation distances and enhance information discrimination. In this section, we provide a theoretical introduction to some necessary concepts in the Lorentz model that are relevant to this paper. Notations used in this paper are listed in Table 6.

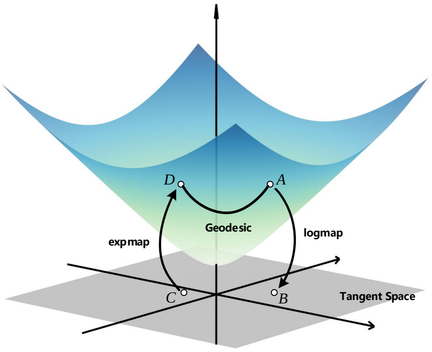


Figure 6: Illustration of the two dimensional Lorentz model $\mathcal{L}_k^2$ and tangent space $\mathcal{T}_{\boldsymbol{o}}\mathcal{L}_k^2$ at the origin $\boldsymbol{o}\left(1/\sqrt{|k|}, 0, 0\right)$. A and D are points in $\mathcal{L}_k^2$, B and C are points in $\mathcal{T}_{\boldsymbol{o}}\mathcal{L}_k^2$. A maps to B via the logarithmic map, and C maps to D via the exponential map. The geodesic between A and D is the shortest curve joining them, its length is calculated by Eq. 35.

Definition. As illustrated in Fig. 6, With a constant negative curvature $k(k < 0)$, the n dimensional Lorentz model is defined as a Riemannian manifold $\mathcal{L}_k^n = (\mathcal{H}_k^n, g_{\boldsymbol{x}}^{\mathcal{L}})$ embedded in the $n+1$ dimensional Minkowski space, in which

$$\mathcal{H}_k^n = \{\boldsymbol{x} \in \mathbb{R}^{n+1} | \langle \boldsymbol{x}, \boldsymbol{x}\rangle_{\mathcal{L}} = 1/k, x_0 > 0\} \qquad (32)$$

represents the upper sheet of an n dimensional hyperboloid with the origin $\boldsymbol{o}(1/\sqrt{|k|}, \mathbf{0}^n)$, $g_{\boldsymbol{x}}^{\mathcal{L}} = \mathrm{diag}(-1, \mathbf{1}^n)$ is the Riemannian metric tensor. $\langle\cdot,\cdot\rangle_{\mathcal{L}}$ denotes the Lorentzian inner product:

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} = \boldsymbol{x} g_{\boldsymbol{x}}^{\mathcal{L}} \boldsymbol{y} = -x_0 y_0 + \sum_{i=1}^{n} x_i y_i \qquad (33)$$

Tangent Space. $\forall \boldsymbol{x} \in \mathcal{L}_k^n$, the tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n$ of $\mathcal{L}_k^n$ at $\boldsymbol{x}$ is defined as an n dimensional vector space of the first-order approximation to $\mathcal{L}_k^n$ around $\boldsymbol{x}$, which is the orthogonal space of $\mathcal{L}_k^n$ at $\boldsymbol{x}$ with respect to the Lorentzian inner product:

$$\mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n = \{\boldsymbol{v} \in \mathbb{R}^{n+1} | \langle \boldsymbol{x}, \boldsymbol{v}\rangle_{\mathcal{L}} = 0\} \qquad (34)$$

Geodesics. Geodesics are the generalization of straight lines in Euclidean space to manifolds. In the Lorentz model, a geodesic between $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{L}_k^n$ is the shortest curve joining $\boldsymbol{x}$ to $\boldsymbol{y}$. Based on the Riemannian metric $g_{\boldsymbol{x}}^{\mathcal{L}}$, the geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given as:

$$d_{\mathcal{L}}^k(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{1/|k|} \cdot \cosh^{-1}(k\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}) \qquad (35)$$

Exponential and Logarithmic Maps. Mapping between Lorentz model and its tangent space is realized by exponential map and logarithmic map. Exponential map $\exp_{\boldsymbol{x}}^k : \mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n \rightarrow \mathcal{L}_k^n$ projects a vector $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n$ to $\mathcal{L}_k^n$, and logarithmic map $\log_{\boldsymbol{x}}^k : \mathcal{L}_k^n \rightarrow \mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n$ is the inverse of $\exp_{\boldsymbol{x}}^k$.

Notation	Meaning
$\mathcal{L}_k^n$	n dimensional Lorentz model of curvature k
$\boldsymbol{o}$	Origin of the Lorentz model $\mathcal{L}_k^n$
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}}$	Lorentzian inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$
$\mathcal{T}_{\boldsymbol{x}}\mathcal{L}_k^n$	Tangent space of $\mathcal{L}_k^n$ at $\boldsymbol{x}$
$d_{\mathcal{L}}^k(\boldsymbol{x}, \boldsymbol{y})$	Geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$
$\exp_{\boldsymbol{x}}^k$	Exponential map
$\gamma(\boldsymbol{u})$	Half-aperture of Lorentzian entailment cone of vector $\boldsymbol{u}$
$\phi(\boldsymbol{u}, \boldsymbol{v})$	Angle between half-lines $(\boldsymbol{ou}$ and $(\boldsymbol{uv}$
$\boldsymbol{z}_i^{rel}$	Hyperbolic representation of i-th relation
$\boldsymbol{z}_{j,k}^{s}$	Hyperbolic representation of k-th instance of i-th relation in support set
$\boldsymbol{z}^{q}$	Hyperbolic representation of instance in query set
$\boldsymbol{z}_i^{c}$	Lorentzian aggregation center of i-th relation

Table 6: Summary of notations.

The exponential and logarithmic map are defined as:

$$\exp_{\boldsymbol{x}}^k(\boldsymbol{v}) = \cosh(\sqrt{|k|}\|\boldsymbol{v}\|_{\mathcal{L}})\boldsymbol{x} + \frac{\sinh(\sqrt{|k|}\|\boldsymbol{v}\|_{\mathcal{L}})}{\sqrt{|k|}\|\boldsymbol{v}\|_{\mathcal{L}}}\boldsymbol{v}$$

$$\log_{\boldsymbol{x}}^k(\boldsymbol{z}) = \frac{\cosh^{-1}(k\langle \boldsymbol{x}, \boldsymbol{z}\rangle_{\mathcal{L}})}{\sqrt{(k\langle \boldsymbol{x}, \boldsymbol{z}\rangle_{\mathcal{L}})^2 - 1}}(\boldsymbol{z} - k\langle \boldsymbol{x}, \boldsymbol{z}\rangle_{\mathcal{L}}\boldsymbol{x}) \tag{36}$$

where $\|\boldsymbol{v}\|_{\mathcal{L}} = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v}\rangle_{\mathcal{L}}}$ denotes the Lorentzian norm of $\boldsymbol{v}$.

E Lorentzian Cosine Similarity

In previous works [Han *et al.*, 2021; Dou *et al.*, 2022], relations and instances are encoded separately, the alignment of their representations is crucial. A common approach is supervised contrastive learning [Khosla *et al.*, 2020], which leverages cosine to measure the similarity of vectors. In the Lorentz model, existing measures like geodesic distance and its squared variant can be used for similarity assessment. However, they fail to fully capture the directional information, hindering subsequent partial order modeling based on angles. Therefore, we derive the Lorentzian cosine similarity.

Theorem 8. *$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{L}_k^n \backslash \{\boldsymbol{o}\}$, let u_i and v_i denote the i-th dimension of $\boldsymbol{u}$ and $\boldsymbol{v}$, the Lorentzian cosine similarity between them $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) \in [0, 1]$ in the Lorentz model is calculated as:*

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{-k\sum\limits_{i=1}^{n} u_i v_i}{\sqrt{|k|u_0^2 - 1}\sqrt{|k|v_0^2 - 1}}$$

Proof. We first introduce the hyperbolic law of cosines. Then, we construct a hyperbolic triangle in the Lorentz model. Based on the hyperbolic law of cosines, we derive the Lorentzian cosine similarity.

Hyperbolic Law of Cosines. In hyperbolic space, angle is a generalization of angle in Euclidean space, which is defined as the angle formed by two geodesics at their intersection. As in Euclidean space, it is measured by the angle between the initial tangent vectors of these two geodesics. With the concepts of angle and geodesic, the triangle can be defined within hyperbolic space. For $A, B, C \in \mathcal{L}_k^n$, a hyperbolic triangle $\triangle ABC$ is constructed by joining any two points through geodesics. Let $a = d_{\mathcal{L}}^k(B, C)$ denotes the length of geodesic between points B and C (and others), the hyperbolic law of cosines [Parkkonen, 2012] is established as follows:

$$\cos(\angle C) = \frac{\cosh(a\sqrt{|k|})\cosh(b\sqrt{|k|}) - \cosh(c\sqrt{|k|})}{\sinh(a\sqrt{|k|})\sinh(b\sqrt{|k|})} \tag{37}$$

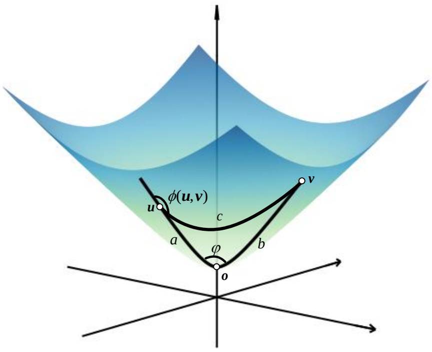


Figure 7: Given any two points $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathcal{L}_k^2$ except the origin $\boldsymbol{o}$, they form a hyperbolic triangle with the origin. c represents the geodesic between $\boldsymbol{u}$ and $\boldsymbol{v}$ (similar for a and b), φ is the angle formed by a and b. The hyperbolic cosine of φ is utilized to quantify the similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$. $\phi(\boldsymbol{u}, \boldsymbol{v})$ denotes the exterior angle between half-lines $(\boldsymbol{ou}$ and $(\boldsymbol{uv}$.

Lorentzian cosine similarity. As illustrated in Fig. 7, given $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{L}_k^n \backslash \{\boldsymbol{o}\}$ and the origin $\boldsymbol{o}(1/\sqrt{|k|}, \boldsymbol{0}^n)$ in the Lorentz model, any two points are joined by geodesic to construct a hyperbolic triangle. We use φ to denote the angle formed by $\boldsymbol{u}$ and $\boldsymbol{v}$, a, b, and c to represent the corresponding geodesics respectively. With Eq. 35, we have

$$\begin{aligned}
a &= \sqrt{1/|k|}\cosh^{-1}(k\langle \boldsymbol{o}, \boldsymbol{u}\rangle_{\mathcal{L}}) = \sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}u_0) \\
b &= \sqrt{1/|k|}\cosh^{-1}(k\langle \boldsymbol{o}, \boldsymbol{v}\rangle_{\mathcal{L}}) = \sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}v_0) \\
c &= \sqrt{1/|k|}\cosh^{-1}(k\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\mathcal{L}}) \\
&= \sqrt{1/|k|}\cosh^{-1}(-k(u_0 v_0 - \sum_{i=1}^{n} u_i v_i))
\end{aligned} \tag{38}$$

hence,

$$\begin{aligned}
\cosh(a\sqrt{|k|}) &= \cosh(\sqrt{|k|}\cdot\sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}u_0)) \\
&= \sqrt{|k|}u_0 \\
\cosh(b\sqrt{|k|}) &= \cosh(\sqrt{|k|}\cdot\sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}v_0)) \\
&= \sqrt{|k|}v_0 \\
\cosh(c\sqrt{|k|}) &= -k(u_0v_0 - \sum_{i=1}^{n} u_iv_i) \\
\sinh(a\sqrt{|k|}) &= \sinh(\sqrt{|k|}\cdot\sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}u_0)) \\
&= \sinh(\cosh^{-1}(\sqrt{|k|}u_0)) \\
&= \sqrt{|k|u_0^2 - 1} \\
\sinh(b\sqrt{|k|}) &= \sinh(\sqrt{|k|}\cdot\sqrt{1/|k|}\cosh^{-1}(\sqrt{|k|}v_0)) \\
&= \sinh(\cosh^{-1}(\sqrt{|k|}v_0)) \\
&= \sqrt{|k|v_0^2 - 1}
\end{aligned} \tag{39}$$

As φ corresponds to $\angle C$ in Eq. 37, we substitute Eq. 39 into Eq. 37, the Lorentzian cosine similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$ is calculated as:

$$\begin{aligned}
\mathrm{sim}(\boldsymbol{u},\boldsymbol{v}) &= \cos\varphi \\
&= \frac{|k|u_0v_0 - (-k(u_0v_0 - \sum_{i=1}^{n} u_iv_i))}{\sinh(\cosh^{-1}(\sqrt{|k|}u_0))\sinh(\cosh^{-1}(\sqrt{|k|}v_0))} \\
&= \frac{-k\sum_{i=1}^{n} u_iv_i}{\sqrt{|k|u_0^2 - 1}\sqrt{|k|v_0^2 - 1}}
\end{aligned} \tag{40}$$

Thus, Theorem 8 is derived. □