DynCSE: Dynamic Contrastive Learning of Sentence Embeddings via Reinforcement Learning from Multiple Teachers

Anonymous ACL submission

Abstract

Unsupervised sentence representation learning has made notable progress in recent years, particularly through contrastive learning and ranking distillation, which leverages ranked semantic knowledge from teacher models to enhance sentence representations. These approachs have reached notable success by extracting ranked knowledge from teacher models to learn semantically discriminative sentence representations. However, existing ranking distillation methods often use static weights for teacher models, which fails to have a varying effect on the student model in different learning stages and lead to potentially limiting performance. In this paper, we propose a novel sentence representation learning framework, DynCSE, which for the first time integrates the Deep Deterministic Policy Gradient (DDPG) into the distillation learning framework. This integration is designed to enhance the efficiency of the agent in dynamically identifying optimal teacher model weights and to mitigate the instability issues associated with network convergence within the agent framework. Experimental results demonstrate that our method outperforms most existing models in the majority of semantic textual similarity (STS) tasks, highlighting its strong capability in both unsupervised and supervised sentence representation learning.

1 Introduction

Sentence embeddings can be applied in various tasks, including information retrieval (Mikolov et al., 2013), text clustering (Ma et al., 2016), and semantic textual similarity comparison (Agirre et al., 2012). One straightforward approach to generating sentence representations is to directly use a designated token embedding or the average token embeddings from the last layer of pre-trained language models (PLMs). However, several studies (Ethayarajh, 2019; Li et al., 2020) have found that native sentence representations derived from PLMs tend to occupy a narrow cone in vector space, significantly limiting their representational capacity, which is known as the anisotropy problem.

Supervised methods like SBERT (Reimers and Gurevych, 2019) usually generate better sentence representations, but they require fine-tuning on large amounts of labeled data. Recently, unsupervised models (Carlsson et al., 2021; Zhang et al., 2021; Giorgi et al., 2021) have adopted a contrastive learning framework, pulling semantically similar sentences closer together and pushing dissimilar ones apart.

These methods usually design different augmentation algorithms to generate positive examples, such as back-translation (Zhang et al., 2021), dropout (Gao et al., 2021) and token shuffling or cutoff (Yan et al., 2021). Despite achieving satisfactory results, these methods treat positive and negative pairs equally without capturing fine-grained semantic ranking information. This leads to a well-knowd issue of feature suppression, making it difficult to distinguish between very similar and less similar sentences.

RankCSE(Liu et al., 2023) introduces a novel approach that integrates ranking distillation and contrastive learning into a unified framework, successfully transferring the performance of the teacher model to the student model. This method performs well on semantic textual similarity (STS) tasks. However, RankCSE's ranking distillation requires assigning static weights to the teacher model during training, which often leads to sampling bias, significantly impacting the performance of the student model.

In cases with multiple teachers, finding appropriate dynamic sampling weights becomes challenging. Additionally, effectively applying these techniques to the field of unsupervised sentence representation is also difficult. Existing methods typically define actions simply as the selection of a

specific teacher, resulting in a discrete action space.

The study that aligns most closely with our work is RLRD (Wang et al., 2024b). RLRD uses PPO as the policy model for the agent to learn the weight adjustments of the teacher model. While the KL-divergence constraint in PPO can prevent the policy from updating too quickly, it may limit the update process when handling continuous action spaces, which can result in slower convergence. Additionally, PPO's stochastic policy requires sampling in continuous action spaces, which can introduce extra variance and lead to decreased policy stability.

We designed a specific reinforcement learning method for unsupervised sentence representation, where we redefine the weights of each teacher model as continuous variables. To handle the continuous environmental variables produced by the teacher models, we employ the DDPG (Lillicrap et al., 2019) algorithm, which is more efficient in continuous action spaces. DDPG outputs concrete actions directly, eliminating the need for sampling, allowing for more precise control over the teacher model's weights. Additionally, its Actor-Critic architecture offers better stability and convergence in continuous action spaces. The Critic provides accurate value estimates, aiding the Actor in improving the policy. Furthermore, DDPG's deterministic policy minimizes the variance during updates, thereby maintaining policy stability.

2 Related Work

2.1 Unsupervised Sentence Representation Learning

Early works typically augmented the concept of word2vec (Mikolov et al., 2013) to learn sentence representations, encompassing approaches such as Skip-Thought (Kiros et al., 2015), FastSent (Ma et al., 2016), and QuickThought (Logeswaran and Lee, 2018). With the widespread adoption of PLMs, various efforts have focused on generating sentence representations by leveraging the embedding of the [CLS] token or applying mean pooling over the last layer of BERT (Reimers and Gurevych, 2019).

However, Ethayarajh (Ethayarajh, 2019) identified the anisotropy issue in language representations, which means that native embeddings learned from pre-trained language models (PLMs) tend to occupy a narrow cone in the vector space. BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) proposed post-processing techniques to address this anisotropy problem. In recent years, contrastive learning has been widely adopted for sentence representation learning to mitigate this issue in language representations. A notable example is SimCSE, which uses dropout as a data augmentation strategy, forming the basis for many subsequent works.

DCLR (Zhou et al., 2022) mitigates the impact of improper negations. ESIMCSE (Wu et al., 2022) employs repeated words as data augmentation to alleviate the tendency of SimCSE to perceive sentences of the same or similar lengths as more semantically similar. SNCSE (Wang et al., 2022) introduces prompt learning, enhancing the model's ability to discern soft negative samples. DiffCSE (Chuang et al., 2022) incorporates equivariant contrastive learning into SimCSE. InfoCSE (Rusak et al., 2024) learns sentence representations by reconstructing original sentence fragments. DenoSent (Wang et al., 2024a) incorporating an intra-sentence perspective, training the model to eliminate noise.

Generative learning techniques (Wang et al., 2021; Wu and Zhao, 2022) have also been proposed to enhance the linguistic interpretability of sentence representations. While these methods have yielded satisfactory results, they have yet to capture fine-grained ranking knowledge between sentences.

2.2 Ranking Distillation

RankEncoder (Seonwoo et al., 2023) shows the semantic meaning of a sentence is also determined by nearest-neighbor sentences that are similar to the input sentence. It argues that sentence representations can be modeled based on the similarity between the current sample and other sentences in the dataset, thus addressing the challenge of distinguishing between similar sentences. Specifically, RankEncoder leverages an external corpus to calculate the similarity between the current sample and sentences in the corpus, and transforms these similarities into rank vectors.

In a similar vein, RankCSE (Liu et al., 2023) proposes two ranking distillation frameworks, ListNet and ListMLE. By using these loss functions, the student model can absorb weak ranking knowledge from the teacher model. RankCSE unifies ranking consistency, ranking distillation, and contrastive learning within a single framework, delivering improved sentence representation performance.

3 Methodology

3.1 Overview

As illustrated in Figure 1, the proposed method is primarily composed of three key components: a student encoder, two teacher models, and a Reinforcement Learning (RL) framework comprising an agent, an environment, and an experience replay pool.

3.2 Contrastive Learning

The contrastive learning framework employed by the student model partitions samples into positive and negative pairs, with the objective of maximizing the similarity between positive samples while minimizing the similarity between negative samples. A common form of contrastive loss utilized in this context is InfoNCE (Rusak et al., 2024).

Consider a mini-batch of sentences $D = \{x_1, x_2, x_3, \ldots, x_n\}$, where each x_i serves as the anchor sample. We employ BERT or RoBERTa models to encode x_i. Due to the stochastic nature of dropout in neural networks, which randomly freezes different nodes within the network, this process yields two distinct representations of the sentence: h_i and h_i^+, where h_i^+ is the positive sample corresponding to the anchor sample h_i. The other sentences h_j within the mini-batch are considered negative samples unrelated to the anchor sample h_i, referred to as soft negative samples. The parameter τ denotes the temperature used in the softmax classification computation.

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{i=1}^{N} \log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i, h_j)/\tau}} \quad (1)$$

To better distinguish semantic and textual similarity, we incorporate the concept from SNCSE (Wang et al., 2022) and enhance it through negative sample learning. By leveraging the Bidirectional Margin Loss (BML) (Wang et al., 2022), our objective is to narrow the gap between positive pairs and soft negative pairs, thereby alleviating the issue of feature suppression.

To better differentiate between semantic and textual similarity, we incorporate the concept of Bidirectional Margin Loss (BML) from SNCSE (Wang et al., 2022) and enhance it through negative sample learning.

Specifically, calculate the cosine similarity difference Δ between positive pairs and soft negative pairs:

$$\Delta = \text{sim}(h_i, h_i^{\#}) - \text{sim}(h_i, h_i^{\#}) \quad (2)$$

and propose bidirectional margin loss (BML) to model the semantic similarity difference:

$$\mathcal{L}_{\text{BML}} = \text{ReLU}(\Delta + \alpha) + \text{ReLU}(-\Delta - \beta) \quad (3)$$

BML loss aims to constraint Δ within this interval:

$$\Delta \in [-\beta, -\alpha] \quad (4)$$

Thereby alleviating the feature suppression problem.

Additionally, we adopt the approach from PromptBERT (Jiang et al., 2022) to preprocess the input sentences, transforming them into template formats and extracting masked fields from the hidden states as representations of the sentence vectors. These sentence vectors are subsequently processed through a Multi-Layer Perceptron (MLP) layer equipped with a non-linear activation function.

3.3 Ranking Distillation

In the context of distillation for ranking, we employ the ListMLE approach to leverage pre-trained teacher models for the optimization of coarse-grained ranking information extraction. These teacher models encode input sentences to generate sentence representations, subsequently computing cosine similarities to derive similarity scores, which serve as soft labels in subsequent stages.

This methodology enables our approach to effectively extract and learn ranking knowledge from the teacher models, thereby enhancing the performance of the student model. The training objective of ListMLE is to minimize the negative log-likelihood between the actual ranking of all samples and the ranking predicted by the model. The form of the objective function is given by

$$\mathcal{L}_{\text{ListMLE}} = -\sum_{i=1}^{N} \log P(\pi_i^T | S(x_i), \tau) \quad (5)$$

where $P(\pi_i^T | S(x_i), \tau)$ denotes the probability of correctly predicting the ranking π_i^T given the model scores $S(x_i)$ for sample x_i and the parameter τ. By maximizing the logarithm of this probability, we can train a model that more accurately predicts the actual ranking.

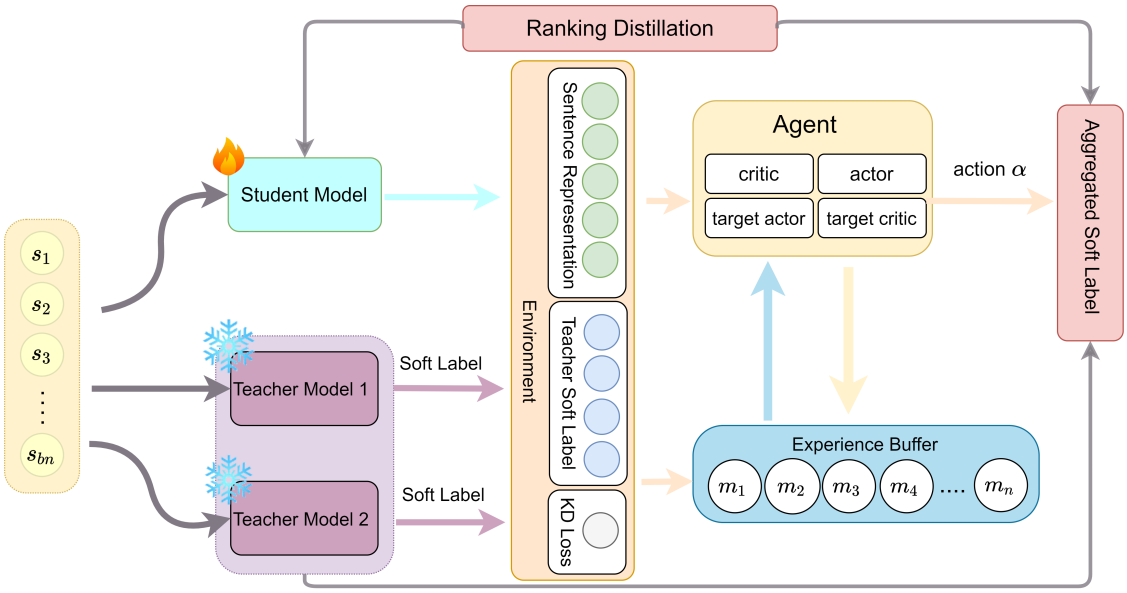

Figure 1: Overview of the proposed approach

3.4 Reinforcement Learning Framework

We adopt a reinforcement learning approach based on Deep Deterministic Policy Gradient (DDPG), as shown in Figure 2. This method adaptively aggregates similarity scores from multiple teacher models into a soft label for training the student model. Each teacher model is associated with an agent, which interacts with the environment, comprising the outputs of the teacher models, the student model, and the distillation loss. The agent's actor network outputs the parameters of a Beta distribution, which is then used to sample from the teacher models and determine their respective weights. After each training iteration, the performance of the student model serves as a reward for policy updates, and the actor network is updated using a soft update. This process continues until the actor and critic networks converge.

3.5 Dynamic Distillation via Reinfrocement Learning

This study employs a reinforcement learning network based on the DDPG (Lillicrap et al., 2019) update strategy, which dynamically selects the weights of teacher models to optimize the learning process. Within the reinforcement learning framework, the fundamental elements encompass state, reward, and action. The following provides a detailed description of these essential components:

State The environmental state is composed of three components. The first part is the vector representations of all sentences in the current batch.

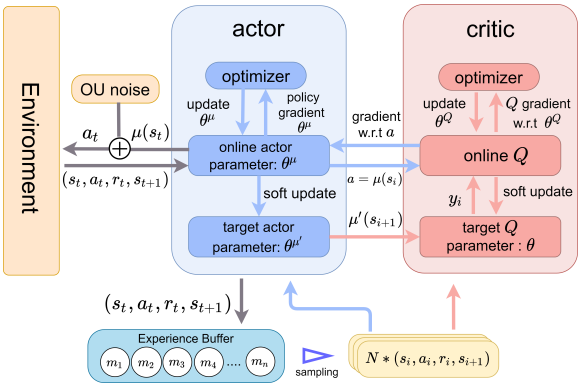

Figure 2: The structure of the DDPG algorithm includes both a actor network and a critic network, each with corresponding target networks. During updates, the target networks are updated through a soft update mechanism using the online networks.

The second is formed by concatenating the soft labels generated by teacher models. A soft label allows the label to be represented as a probability distribution, where each class is assigned a certain probability, rather than a binary, all-or-nothing label.

The third part originates from the distillation losses between the teacher models and the student encoder, serving as a bridge of information between the teacher models and the student encoder and providing a basis for dynamic weight adjustment.

Action

Instead of treating action selection as a discrete choice among teacher models, we adopt a reinforcement learning approach inspired by RLRD (Wang

et al., 2024b) to dynamically assign weights to each teacher model, effectively fusing their guidance signals. However, departing from RLRD, which uses PPO-Clip, we employ the Deep Deterministic Policy Gradient (DDPG) algorithm, better suited for the nuances of natural language processing tasks. Specifically, we define the action as a vector of continuous variables, representing the weight assigned to each teacher model. Each weight is constrained to the interval [0, 1]. Our model parameterizes the action distribution by outputting α and β, the shape parameters of a Beta(α, β) distribution. This allows for a flexible and continuous representation of the weighting strategy.

Within the Beta distribution, α can be interpreted as the number of successes, while β represents the number of failures. Consequently, the action value sampled from Beta(α, β) (ranging between 0 and 1) essentially embodies the "intensity" of a certain behavior or the "likelihood" of executing that behavior.

In this work, a higher action value may indicate that the teacher model tends to select that behavior, due to the higher probability of success for that behavior based on historical data. Specifically, a higher α value may suggest an increased confidence in the success of the action, while a higher β value may indicate an increased confidence in the failure of the action. This parameterized approach, grounded in historical experience, can assist the model in better balancing the selection among different behaviors.

$$\pi_\theta(s_j, a_j) = \text{Beta}(\alpha, \beta) \tag{6}$$

Actor The agent is composed of an actor, a $\text{target}_{\text{actor}}$, a critic, and a $\text{target}_{\text{critic}}$ network. The state s serves as the input to the Actor network, while the outputs α and β are used to parameterize the Beta distribution, thereby determining the probability of the action. The actor network utilizes the input state s, performing linear transformations and ReLU activations through specific weights W_α and W_β, along with biases b_α and b_β, to obtain α and β. This process can be represented by the following equation:

$$\alpha = \text{ReLU}(W_\alpha s + b_\alpha) \tag{7}$$

$$\beta = \text{ReLU}(W_\beta s + b_\beta) \tag{8}$$

Critic In the DDPG algorithm, the Critic network plays a crucial role, with its core task being to evaluate the quality of the policy and provide a basis for policy improvement to the Actor network. Specifically, the Critic network learns the state-action value function (Q-function), which predicts the expected long-term cumulative reward given a specific state and action. This process can be formalized as the following equation:

$$Q(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right] \tag{9}$$

where $Q(s, a)$ represents the expected return for executing action a in state s, γ is the discount factor, and r_t is the immediate reward at time step t. The training process of the Critic network is based on the Bellman Equation, updating the network parameters by minimizing the mean squared error (MSE) between the target Q-value and the predicted Q-value. The calculation of the target Q-value is given by the following formula:

$$\text{target}_Q = r + \gamma Q'(s', \mu'(s')) \tag{10}$$

where r represents the immediate reward, s' is the next state, $\mu'(s')$ is the target action generated by the target Actor network in the next state s', and $Q'(s', \mu'(s'))$ is the predicted Q-value by the target Critic network for the next state s' and the target action $\mu'(s')$. The loss function for the Critic network is defined as:

$$\mathcal{L}(\theta^Q) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(Q(s, a) - target_Q)^2] \tag{11}$$

where θ^Q denotes the parameters of the Critic network, and $\mathcal{D}$ represents the experience replay pool. By minimizing the loss function $\mathcal{L}(\theta^Q)$ through gradient descent, the parameters of the Critic network are updated, thereby progressively approximating the true state-action value function.

target critic The target Critic network ($\text{target}_{\text{critic}}$) in the DDPG algorithm serves to generate stable target Q-values, ensuring a smoother and more stable update process for the Critic network. Specifically, the target Critic network operates through the following two mechanisms. Firstly, the calculation of the target Q-value involves the target Critic network computing the target Q-value (target_{Q}), which is the output given the $\text{state}_{\text{next}}$ and $\text{target}_{\text{action}}$. This target Q-value is utilized to compute the loss function for the Critic network. Secondly, the stabilization of Q-value estimation is achieved by employing the target Q-value

computed by the target Critic network, which reduces fluctuations in the Q-value estimation process, thereby making the updates to the Critic network more stable. This contributes to enhancing the convergence speed and stability of the Critic network.

reward The reward represents the loss function value during the training process, which guides the student model learning by minimizing these losses.

3.6 Optimization

The update strategy of Deep Deterministic Policy Gradient (DDPG) plays a crucial role in addressing the instability of policy updates in continuous action spaces. It utilizes the Actor-Critic algorithm as its foundational framework, employing deep neural networks to approximate the policy network and action-value function. The DDPG architecture incorporates dual neural network structures for both the policy and value functions, with an online network and a target network. The parameters of the target network are updated through a soft update mechanism.

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{aligned} \tag{12}$$

where, τ represents the soft update coefficient, and $\theta^{Q'}$ and $\theta^{\mu'}$ denote the parameters of the target value network and target policy network, respectively. This approach improves the stability of the learning process and accelerates convergence. Additionally, the algorithm incorporates an experience replay mechanism, where the interaction data generated by the Actor and the environment are stored in a replay buffer. Mini-batches are sampled from this buffer for training, similar to the experience replay mechanism in DQN, which reduces the correlation and dependency between samples, making the algorithm more likely to converge.

4 Experiment

4.1 Setup

Datasets evaluate our approach on the Semantic Textual Similarity (STS) dataset, utilizing the SentEval toolkit (Conneau and Kiela, 2018). The dataset comprises seven sub-tasks, including STS2012-STS2016 (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015; Agirre et al., 2016), STSbenchmark (Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014). This dataset encompasses sentence pairs extracted from headlines, news articles, videos, images, and natural language inference tasks, with human-assigned semantic similarity scores ranging from 0 to 5.

Implementation Details: To facilitate a fair comparison, we randomly sampled 1,000,000 sentences from the English Wikipedia corpus provided by SimCSE. In accordance with prior work, we commenced from the pre-trained checkpoints of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), leveraging the embeddings corresponding to the [CLS] token as the representations of the input sentences.

In this work, We first train the RankEncoder model. Using the best-performing RankEncoder and SimCSE-BERT$_{\text{large}}$ as teacher models, we employ BERT$_{\text{base}}$ and BERT$_{\text{large}}$ as the student model. We use SimCSE-RoBERTa$_{\text{base}}$ and SimCSE-RoBERTa$_{\text{large}}$ as teacher models, while RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ as student models. then, We evaluated our models every 125 training steps on the development set of STS-B and preserved the best checkpoints for evaluation on the test sets of all STS tasks.

Baselines: We conducted a comparative analysis of DynCSE against several state-of-the-art methods for unsupervised sentence representation. InferSent (Conneau et al., 2017) uses Glove and is further trained on the NLI dataset. Universal Sentence Encoder (Cer et al., 2018) employs a Transformer to reconstruct surrounding sentences. BERT (CLS, Mean, First-Last Avg.) (Devlin et al., 2019) uses different pooling strategies to generate sentence representations. BERT-Flow (Li et al., 2020) maps BERT outputs to a Gaussian distribution, while BERTWhitening (Su et al., 2021) uses vector whitening. ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021) employ contrastive learning with data augmentation. Other methods include DCLR (Zhou et al., 2022), DiffCSE (Chuang et al., 2022), PromptBERT (Jiang et al., 2022), RLRD (Wang et al., 2024b), SNCSE (Wang et al., 2022), and CMLM (Yang et al., 2021). RankCSE (Liu et al., 2023) uses knowledge distillation, while Densent (Wang et al., 2024a) trains on noisy data. All models were evaluated on the STS dataset.

4.2 main result

Table 1 illustrates the performance of DynCSE on 7 STS tasks compared to previous methods. All experiments are conducted under a self-supervised/unsupervised setting except for non-

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Non-BERT Models								
GloVe embeddings (avg.)	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
InferSent-GloVe	52.86	75.65	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
BERT & Post-Processing Models								
$\text{BERT}_{\text{base}}$ (CLS)	21.54	32.11	21.28	37.89	44.24	20.30	42.42	31.40
$\text{BERT}_{\text{base}}$ (Mean)	30.87	59.89	47.73	60.29	63.73	47.29	58.22	52.57
$\text{BERT}_{\text{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
Generative-based Models								
CMLM	58.20	61.07	61.67	73.32	74.88	76.60	64.80	67.22
PaSeR	70.21	83.88	73.06	83.87	77.60	79.19	65.31	76.16
DenoSent	75.57	83.77	77.24	84.30	79.51	80.81	74.09	79.33
Contrastive Learning-based Models								
ConSERT	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT_{base}	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DCLR	70.81	73.03	75.11	82.56	**84.44**	78.31	71.59	77.22
DiffCSE	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
PromptBERT	71.56	84.58	76.98	84.47	80.60	81.60	69.87	78.54
SNCSE	70.67	84.79	76.99	83.69	80.51	81.35	74.77	78.97
RankCSE	75.66	86.27	77.81	84.74	81.10	81.80	75.13	80.36
RLRD	75.84	86.28	79.22	85.80	81.43	**82.90**	74.79	80.89
DynCSE-$\text{BERT}_{\text{base}}$	**76.59**	**86.37**	**79.27**	**85.86**	81.44	82.85	**75.26**	**81.09**

Table 1: Evaluation performance on 7 STS tasks. The reported metric is spearman correlation(×100) based on cosine similarity following previous works. Bolded results and underlined results correspond to the best and second-best results in the same dataset, respectively.

BERT models. The results reveal that methods that either do not use a PLM or rely solely on post-processing are less effective than those that apply contrastive and generative approaches on a PLM. In the context of single-objective learning, the contrastive objective proves to be more effective for semantic textual similarity tasks than the generative objective since it directly optimizes representation similarities.

Compared to the RLRD model, the DynCSE model demonstrates a 0.2% improvement in performance on the STS task. This enhancement can be attributed to the effectiveness of incorporating a target policy network and a target evaluation network, as the DynCSE model is identical to the RLRD model in terms of dynamic teacher weights.

4.3 Ablation Studies

To validate the impact of different losses on our model, we conducted a series of ablation studies by progressively removing the BML loss, distill loss, and RL loss. We employed the 'bert base' model as our backbone network on the STS-B dataset.

Method	STS avg.
DynCSE w/o (BML, Rank Distill, RL)	77.48
DynCSE w/o (Rank Distill, RL)	78.53
DynCSE w/o RL	80.96
DynCSE	81.09

Table 2: The Spearman correlation coefficient for sentence embedding performance on the STS-B development set is reported. BML refers to the bidirectional marginal loss, rank distillation denotes the ListMLE method we employed, and RL loss represents the DDPG-based knowledge distillation approach used in our work.

Table 2 presents the Spearman correlation coefficients obtained under various configurations. These ablation studies clearly demonstrate the efficacy of each component in enhancing the model's capability for the STS task.

4.4 Comparisons of Different Teachers

In this experiment, we compared the effectiveness of our proposed reinforcement learning strategy with various teacher models. Table 3 demonstrates that our method of dynamically adjusting the weights of teacher models significantly improves model performance compared to the traditional strategy of averaging all teacher weights.

Teacher	DynCSE
RankEncoder	80.26
$\text{SimCSE}_{\text{large}}$	79.81
$\text{SimCSE}_{\text{base}}$ + $\text{SimCSE}_{\text{large}}$	79.27
DiffCSE+ $\text{SimCSE}_{\text{large}}$	79.54
RankEncoder + $\text{SimCSE}_{\text{large}}$	81.09

Table 3: Comparisons of different teachers based on $\text{BERT}_{\text{base}}$.

The impact of different teacher models on DynCSE performance is explored in Table 3, where the student models under DynCSE consistently outperform the teacher models. This indicates that combining ranking distillation with dynamic teacher weighting leads to more semantically distinct sentence representations.

By comparing the performance of DynCSE models using different teachers, we observe that superior teacher models result in better DynCSE performance. This aligns with our expectations, as accurate ranking labels facilitate more effective ranking knowledge transfer.

Teacher	Model	STS avg.
$\text{SimCSE-BERT}_{\text{base}}$ $\text{SimCSE-BERT}_{\text{large}}$	bert-large	82.38
$\text{SimCSE-RoBERTa}_{\text{base}}$ $\text{SimCSE-RoBERTa}_{\text{large}}$	roberta-large	84.42

Table 4: A study on the performance of different student models under a teacher model following supervised training. The results on the STS task indicate that $\text{SimCSE-BERT}_{\text{large}}$ achieves a score of 82.21, while $\text{SimCSE-RoBERTa}_{\text{base}}$ is 82.52, $\text{SimCSE-RoBERTa}_{\text{large}}$ is 83.76

Another noteworthy observation is that training student models with teacher models after supervised training yields better results than directly training with supervised labels. This validates that DynCSE can learn more from the ranking lists generated by teacher models than from the labels alone. As shown in Table 4, student models trained with superivsed $\text{SimCSE-BERT}_{\text{base}}$ and superivsed $\text{SimCSE-BERT}_{\text{large}}$ as teacher models outperform previous supervised models when using a BERT-large student model.

Similarly, we conducted the same verification using $\text{RoBERTa}_{\text{large}}$, and the results also surpass those of all previous supervised training approaches.

5 Conclusion

This paper proposes a reinforcement learning based multi-teacher distillation framework, DynCSE, for unsupervised sentence representation learning. DynCSE dynamically adjusts the weights of teacher models and extracts ranking knowledge through a ranking distillation mechanism. Specifically, treating the teacher model weights as continuous action variables, it employs the Deep Deterministic Policy Gradient (DDPG) algorithm to mitigate the instability commonly encountered in updating continuous action spaces. Experimental results show that DynCSE outperforms state-of-the-art models on all STS benchmark datasets, We hope our work will provide a new perspective on universal sentence representation learning.

6 Limitation

In this section, we discuss the limitations of the current work. First, although satisfactory results have been achieved, our model requires the computation of pseudo-ranking labels from the teacher and additional training of more agents, resulting in extra training time for each epoch. Second, in our implementation and experiments, we directly used RankEncoder and $\text{SimCSE}_{\text{large}}$ as multiple teachers, with DDPG as the algorithm for training agents. However, the selection of the optimal combination of teacher models warrants further exploration. Additionally, knowledge distillation still depends on the performance of the teacher model, which presents a bottleneck. Recent advancements in large models have been widely applied in this area, and we consider incorporating large models into the overall framework as a data augmentation or knowledge distillation method, which could help researchers better understand the potential for improvement. We plan to conduct further research in this direction in the future.

Disinformation and Misinformation: The student model might be exploited to generate misleading content, including disinformation campaigns or fake news, particularly when reinforcement learn-

ing is used to optimize content generation.

Generating Fake Profiles: If the model is applied to social media or digital identities, it may be used to create convincing fake profiles, leading to identity theft or fraud.

Surveillance and Privacy Violations: Reinforcement learning models could potentially be deployed in surveillance systems, unintentionally invading privacy or facilitating unethical monitoring.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65,

Hong Kong, China. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, Online. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. *Preprint*, arXiv:2201.04337.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *Preprint*, arXiv:1506.06726.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2019. Continuous control with deep reinforcement learning. *Preprint*, arXiv:1509.02971.

Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, Kai Chen, and Rui Yan. 2023. RankCSE: Unsupervised sentence representations learning via learning to rank. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13785–13802, Toronto, Canada. Association for Computational Linguistics.

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. *Preprint*, arXiv:1907.11692.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Shutian Ma, Chengzhi Zhang, and Daqing He. 2016. Document representation methods for clustering bilingual documents. In *Proceedings of the 79th ASIS &T Annual Meeting: Creating Knowledge, Enhancing Lives through Information & Technology*, ASIST '16, USA. American Society for Information Science.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Preprint*, arXiv:1310.4546.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. 2024. Infonce: Identifying the gap between theory and practice. *Preprint*, arXiv:2407.00143.

Yeon Seonwoo, Guoyin Wang, Changmin Seo, Sajal Choudhary, Jiwei Li, Xiang Li, Puyang Xu, Sunghyun Park, and Alice Oh. 2023. Ranking-enhanced unsupervised sentence representation learning. *Preprint*, arXiv:2209.04333.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *Preprint*, arXiv:2103.15316.

Hao Wang, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. *Preprint*, arXiv:2201.05979.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning. *Preprint*, arXiv:2104.06979.

Xinghao Wang, Junliang He, Pengyu Wang, Yunhua Zhou, Tianxiang Sun, and Xipeng Qiu. 2024a. Denosent: A denoising objective for self-supervised sentence representation learning. *Preprint*, arXiv:2401.13621.

Xintao Wang, Rize Jin, and Shibo Qi. 2024b. Reinforced multi-teacher knowledge distillation for unsupervised sentence representation. In *International Conference on Artificial Neural Networks*, pages 320–332. Springer.

Bohong Wu and Hai Zhao. 2022. Sentence representation learning with generative objective rather than contrastive objective. In *Proceedings of the 2022*

Conference on Empirical Methods in Natural Language Processing, pages 3356–3368, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2021. Universal sentence representation learning with conditional masked language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6216–6228, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yan Zhang, Ruidan He, Zuozhu Liu, Lidong Bing, and Haizhou Li. 2021. Bootstrapped unsupervised sentence representation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5168–5180, Online. Association for Computational Linguistics.

Kun Zhou, Beichen Zhang, Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6120–6130, Dublin, Ireland. Association for Computational Linguistics.

A Training Detail

We implement all experiments with the deep learning framework PyTorch on a single NVIDIA Tesla A100 GPU (40GB memory). We train our models for 4 epochs, which took 20 hours overall. We evaluate the model every 125 steps on the dev set of STS-B and keep the best checkpoint for the final evaluation on test sets of all STS tasks. We carry out grid-search of learning rate $\in$ {2e-5, 3e-5 } and temperatures $\tau 2, \tau 3 \in$ { 0.0125, 0.025, 0.05 }, while setting batch size to 128

B license of scientific artifacts

BERT-base and BERT-large: BERT models, developed by Google, are released under the Apache License 2.0. This license is highly permissive and widely used in academic and commercial projects. It allows the model to be freely used, modified, and redistributed, both in academic research and in commercial applications, provided that proper attribution is given to the original authors. The license also requires that any modified versions or derivative works clearly indicate the changes made. Additionally, the Apache License disclaims any warranties, so users assume responsibility for any issues arising from the model's use.

RoBERTa-base and RoBERTa-large: The RoBERTa models, created by Facebook AI, are distributed under the MIT License. Like the Apache License, the MIT License is permissive, allowing users to freely use, modify, and redistribute the models in both academic and commercial contexts. The main requirement is attribution to the original authors, but the MIT License does not impose restrictions on how modifications are shared, giving users more flexibility in how they adapt the models.

SimCSE Pre-trained Models: The pre-trained SimCSE models (both unsupervised and supervised) are generally released under the Apache License 2.0, following a similar permissive structure to BERT. This allows for widespread use and modification, making SimCSE models highly adaptable for both academic research and commercial use. Users must ensure that any modifications are properly documented and that appropriate credit is given to the original authors.

RankEncoder Pre-trained Models: Similar to SimCSE, the pre-trained models of RankEncoder are also typically licensed under the Apache License 2.0. This enables free usage, modification, and redistribution in various projects. The flexibility of the Apache License encourages innovation, provided that the original creators are credited, and any alterations are transparently indicated.

SNCSE and SimCSE: Typically, these papers are published under CC BY licenses, permitting both academic and commercial reuse, provided proper credit is given.

RankCSE, DiffCSE, PromptBERT, and ConSERT: These papers also follow CC BY or CC BY-NC (Non-Commercial) licensing. CC BY-NC restricts commercial use but still allows for broad academic sharing and adaptation. Researchers us-

ing these papers must be mindful of whether commercial use is permitted under the specific license.