DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings

Yung-Sung Chuang[†] **Rumen Dangovski**[†] **Hongyin Luo**[†] **Yang Zhang**[‡] **Shiyu Chang**[*]
Marin Soljačić[†] **Shang-Wen Li**[◇] **Wen-tau Yih**[◇] **Yoon Kim**[†] **James Glass**[†]
Massachusetts Institute of Technology[†] Meta AI[◇]
MIT-IBM Watson AI Lab[‡] UC Santa Barbara[*]
yungsung@mit.edu

Abstract

We propose DiffCSE, an unsupervised contrastive learning framework for learning sentence embeddings. DiffCSE learns sentence embeddings that are sensitive to the difference between the original sentence and an edited sentence, where the edited sentence is obtained by stochastically masking out the original sentence and then sampling from a masked language model. We show that DiffSCE is an instance of equivariant contrastive learning (Dangovski et al., 2021), which generalizes contrastive learning and learns representations that are insensitive to certain types of augmentations and sensitive to other "harmful" types of augmentations. Our experiments show that DiffCSE achieves state-of-the-art results among unsupervised sentence representation learning methods, outperforming unsupervised SimCSE[1] by 2.3 absolute points on semantic textual similarity tasks. [2]

1 Introduction

Learning "universal" sentence representations that capture rich semantic information and are at the same time performant across a wide range of downstream NLP tasks without task-specific finetuning is an important open issue in the field (Conneau et al., 2017; Cer et al., 2018; Kiros et al., 2015; Logeswaran and Lee, 2018; Giorgi et al., 2020; Yan et al., 2021; Gao et al., 2021). Recent work has shown that finetuning pretrained language models with *contrastive learning* makes it possible to learn good sentence embeddings without any labeled data (Giorgi et al., 2020; Yan et al., 2021; Gao et al., 2021). Contrastive learning uses multiple augmentations on a single datum to construct positive pairs whose representations are trained to be more similar to one another than negative pairs. While different data augmentations (random cropping, color jitter, rotations, etc.) have been found to be crucial for pretraining vision models (Chen et al., 2020), such augmentations have generally been unsuccessful when applied to contrastive learning of sentence embeddings. Indeed, Gao et al. (2021) find that constructing positive pairs via a simple dropout-based augmentation works much better than more complex augmentations such as word deletions or replacements based on synonyms or masked language models. This is perhaps unsurprising in hindsight; while the training objective in contrastive learning encourages representations to be *invariant* to augmentation transformations, direct augmentations on the input (e.g., deletion, replacement) often change the meaning of the sentence. That is, ideal sentence embeddings should *not* be invariant to such transformations.

We propose to learn sentence representations that are *aware* of, but not necessarily invariant to, such direct surface-level augmentations. This is an instance of *equivariant* contrastive learning (Dangovski et al., 2021), which improves vision representation learning by using a contrastive loss on *insensitive* image transformations (e.g., grayscale) and a prediction loss on *sensitive* image transformations (e.g., rotations). We operationalize equivariant contrastive learning on sentences by using dropout-based augmentation as the insensitive transformation (as in SimCSE (Gao et al., 2021)) and MLM-based word replacement as the sensitive transformation. This results in an additional cross-entropy loss based on the *difference* between the original and the transformed sentence.

We conduct experiments on 7 semantic textual similarity tasks (STS) and 7 transfer tasks from SentEval (Conneau and Kiela, 2018) and find that this difference-based learning greatly improves over standard contrastive learning. Our DiffCSE approach can achieve around 2.3% absolute improve-

[1] SimCSE has two settings: unsupervised and supervised. In this paper, we focus on the unsupervised setting. Unless otherwise stated, in this paper we use SimCSE to refer to unsupervised SimCSE.

[2] Pretrained models and code are available at https://github.com/voidism/DiffCSE.

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4207 - 4218
July 10-15, 2022 ©2022 Association for Computational Linguistics

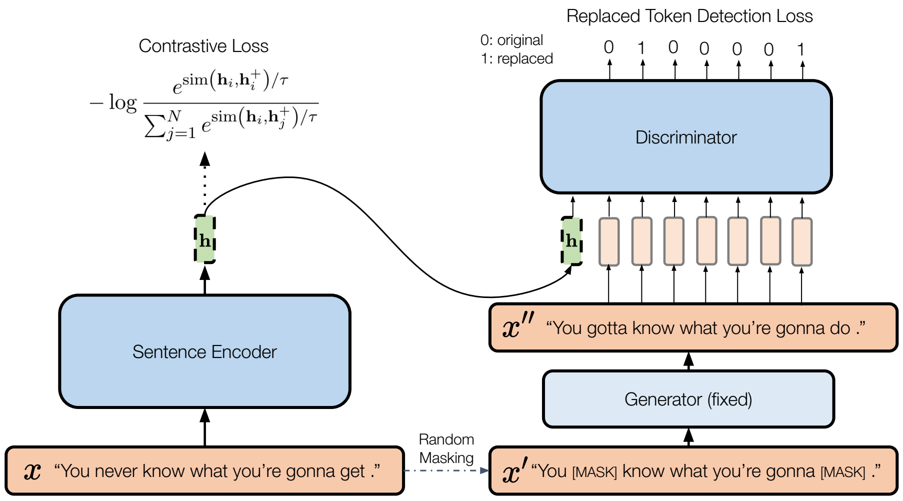

Contrastive Loss
$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^N e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}}$
h
Sentence Encoder
x "You never know what you're gonna get ."
Random Masking
Replaced Token Detection Loss
0: original
1: replaced
0 1 0 0 0 0 1
Discriminator
h
x'' "You gotta know what you're gonna do ."
Generator (fixed)
x' "You [MASK] know what you're gonna [MASK] ."

Figure 1: Illustration of DiffCSE. On the left-hand side is a standard SimCSE model trained with regular contrastive loss on dropout transformations. On the right hand side is a conditional difference prediction model which takes the sentence vector $\mathbf{h}$ as input and predict the difference between x and x''. During testing we discard the discriminator and only use $\mathbf{h}$ as the sentence embedding.

ment on STS datasets over SimCSE, the previous state-of-the-art model. We also conduct a set of ablation studies to justify our designed architecture. Qualitative study and analysis are also included to look into the embedding space of DiffCSE.

2 Background and Related Work

2.1 Learning Sentence Embeddings

Learning universal sentence embeddings has been studied extensively in prior work, including unsupervised approaches such as Skip-Thought (Kiros et al., 2015), Quick-Thought (Logeswaran and Lee, 2018) and FastSent (Hill et al., 2016), or supervised methods such as InferSent (Conneau et al., 2017), Universal Sentence Encoder (Cer et al., 2018) and Sentence-BERT (Reimers and Gurevych, 2019). Recently, researchers have focused on (unsupervised) contrastive learning approaches such as SimCLR (Chen et al., 2020) to learn sentence embeddings. SimCLR (Chen et al., 2020) learns image representations by creating semantically close augmentations for the same images and then pulling these representations to be closer than representations of random negative examples. The same framework can be adapted to learning sentence embeddings by designing good augmentation methods for natural language. ConSERT (Yan et al., 2021) uses a combination of four data augmentation strategies: adversarial attack, token shuffling, cut-off, and dropout. DeCLUTR (Giorgi et al., 2020) uses overlapped spans as positive examples and distant spans as negative examples for learning contrastive span representations. Finally, SimCSE (Gao et al., 2021) proposes an extremely simple augmentation strategy by just switching dropout masks. While simple, sentence embeddings learned in this manner have been shown to be better than other more complicated augmentation methods.

2.2 Equivariant Contrastive Learning

DiffCSE is inspired by a recent generalization of contrastive learning in computer vision (CV) called equivariant contrastive learning (Dangovski et al., 2021). We now explain how this CV technique can be adapted to natural language.

Understanding the role of input transformations is crucial for successful contrastive learning. Past empirical studies have revealed useful transformations for contrastive learning, such as random resized cropping and color jitter for computer vision (Chen et al., 2020) and dropout for NLP (Gao et al., 2021). Contrastive learning encourages representations to be insensitive to these transformations, i.e. the encoder is trained to be invariant to a set of manually chosen transformations. The above studies in CV and NLP have also revealed transformations that are *harmful* for contrastive learning. For example, Chen et al. (2020) showed that making the representations insensitive to rotations decreases the ImageNet linear probe accuracy, and Gao et al. (2021) showed that using an MLM to replace 15% of the words drastically reduces performance on

STS-B. While previous works simply omit these transformations from contrastive pre-training, here we argue that we should still make use of these transformations by learning representations that are *sensitive* (but not necessarily invariant) to such transformations.

The notion of (in)sensitivity can be captured by the more general property of equivariance in mathematics. Let T be a transformation from a group G and let $T(x)$ denote the transformation of a sentence x. Equivariance is the property that there is an induced group transformation T' on the output features (Dangovski et al., 2021):

$$f(T(x)) = T'(f(x)).$$

In the special case of contrastive learning, T''s target is the identity transformation, and we say that f is trained to be "invariant to T." However, invariance is just a trivial case of equivariance, and we can design training objectives where T' is not the identity for some transformations (such as MLM), while it is the identity for others (such as dropout). Dangovski et al. (2021) show that generalizing contrastive learning to equivariance in this way improves the semantic quality of features in CV, and here we show that the complementary nature of invariance and equivariance extends to the NLP domain. The key observation is that the encoder should be equivariant to MLM-based augmentation instead of being invariant. We can operationalize this by using a conditional discriminator that combines the sentence representation with an edited sentence, and then predicts the *difference* between the original and edited sentences. This is essentially a conditional version of the ELECTRA model (Clark et al., 2020), which makes the encoder equivariant to MLM by using a binary discriminator which detects whether a token is from the original sentence or from a generator. We hypothesize that conditioning the ELECTRA model with the representation from our sentence encoder is a useful objective for encouraging f to be "equivariant to MLM."

To the best of our knowledge, we are the first to observe and highlight the above parallel between CV and NLP. In particular, we show that equivariant contrastive learning extends beyond CV, and that it works for transformations even without algebraic structures, such as diff operations on sentences. Further, insofar as the canonical set of useful transformations is less established in NLP than is in CV, DiffCSE can serve as a diagnostic tool for NLP researchers to discover useful transformations.

3 Difference-based Contrastive Learning

Our approach is straightforward and can be seen as combining the standard contrastive learning objective from SimCSE (Figure 1, left) with a *difference prediction* objective which conditions on the sentence embedding (Figure 1, right).

Given an unlabeled input sentence x, SimCSE creates a positive example x^+ for it by applying different dropout masks. By using the BERT$_{\texttt{base}}$ encoder f, we can obtain the sentence embedding $\mathbf{h} = f(x)$ for x (see section 4 for how $\mathbf{h}$ is obtained). The training objective for SimCSE is:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{e^{\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_i^{+}\right)/\tau}}{\sum_{j=1}^{N} e^{\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j^{+}\right)/\tau}},$$

where N is the batch size for the input batch $\{x_i\}_{i=1}^{N}$ as we are using in-batch negative examples, $\operatorname{sim}(\cdot,\cdot)$ is the cosine similarity function, and τ is a temperature hyperparameter.

On the right-hand side of Figure 1 is a conditional version of the difference prediction objective used in ELECTRA (Clark et al., 2020), which contains a generator and a discriminator. Given a sentence of length T, $x = [x_{(1)}, x_{(2)}, ..., x_{(T)}]$, we first apply a random mask $m = [m_{(1)}, m_{(2)}, ..., m_{(T)}], m_{(t)} \in [0, 1]$ on x to obtain $x' = m \cdot x$. We use another pretrained MLM as the generator G to perform masked language modeling to recover randomly masked tokens in x' to obtain the edited sentence $x'' = G(x')$. Then, we use a discriminator D to perform the Replaced Token Detection (RTD) task. For each token in the sentence, the model needs to predict whether it has been replaced or not. The cross-entropy loss for a single sentence x is:

$$\begin{aligned}\mathcal{L}_{\text{RTD}}^{x} = \sum_{t=1}^{T} \Big(& -\mathbb{1}\left(x''_{(t)} = x_{(t)}\right) \log D\left(x'', \mathbf{h}, t\right) \\ & - \mathbb{1}\left(x''_{(t)} \neq x_{(t)}\right) \log\left(1 - D\left(x'', \mathbf{h}, t\right)\right) \Big)\end{aligned}$$

And the training objective for a batch is $\mathcal{L}_{\text{RTD}} = \sum_{i=1}^{N} \mathcal{L}_{\text{RTD}}^{x_i}$. Finally we optimize these two losses together with a weighting coefficient λ:

$$\mathcal{L} = \mathcal{L}_{\text{contrast}} + \lambda \cdot \mathcal{L}_{\text{RTD}}$$

The difference between our model and ELECTRA is that our discriminator D is *conditional*, so it can use the information of x compressed in a fixed-dimension vector $\mathbf{h} = f(x)$. The gradient of D can be backward-propagated into f through $\mathbf{h}$. By doing so, f will be encouraged to make $\mathbf{h}$ informative enough to cover the full meaning of x, so that D can distinguish the tiny difference between x and x''. This approach essentially makes the conditional discriminator perform a "diff operation", hence the name DiffCSE.

When we train our DiffCSE model, we fix the generator G, and only the sentence encoder f and the discriminator D are optimized. After training, we discard D and only use f (which remains fixed) to extract sentence embeddings to evaluate on the downstream tasks.

4 Experiments

4.1 Setup

In our experiment, we follow the setting of unsupervised SimCSE (Gao et al., 2021) and build our model based on their PyTorch implementation.[3] We also use the checkpoints of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as the initialization of our sentence encoder f. We add an MLP layer with Batch Normalization (Ioffe and Szegedy, 2015) (BatchNorm) on top of the `[CLS]` representation as the sentence embedding. We will compare the model with/without BatchNorm in section 5. For the discriminator D, we use the same model as the sentence encoder f (BERT/RoBERTa). For the generator G, we use the smaller DistilBERT and DistilRoBERTa (Sanh et al., 2019) for efficiency. Note that the generator is fixed during training unlike the ELECTRA paper (Clark et al., 2020). We will compare the results of using different size model for the generator in section 5. More training details are shown in Appendix A.

4.2 Data

For unsupervised pretraining, we use the same 10^6 randomly sampled sentences from English Wikipedia that are provided by the source code of SimCSE.[3] We evaluate our model on 7 semantic textual similarity (STS) and 7 transfer tasks in SentEval.[4] STS tasks includes STS 2012–2016 (Agirre et al., 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). All the STS experiments are fully unsupervised, which means no STS training datasets are used and all embeddings are fixed once they are trained. The transfer tasks are various sentence classification tasks, including MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000) and MRPC (Dolan and Brockett, 2005). In these transfer tasks, we will use a logistic regression classifier trained on top of the frozen sentence embeddings, following the standard setup (Conneau and Kiela, 2018).

4.3 Results

Baselines We compare our model with many strong unsupervised baselines including SimCSE (Gao et al., 2021), IS-BERT (Zhang et al., 2020), CMLM (Yang et al., 2020), DeCLUTR (Giorgi et al., 2020), CT-BERT (Carlsson et al., 2021), SG-OPT (Kim et al., 2021) and some post-processing methods like BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) along with some naive baselines like averaged GloVe embeddings (Pennington et al., 2014) and averaged first and last layer BERT embeddings.

Semantic Textual Similarity (STS) We show the results of STS tasks in Table 1 including BERT$_{\texttt{base}}$ (upper part) and RoBERTa$_{\texttt{base}}$ (lower part). We also reproduce the previous state-of-the-art SimCSE (Gao et al., 2021). DiffCSE-BERT$_{\texttt{base}}$ can significantly outperform SimCSE-BERT$_{\texttt{base}}$ and raise the averaged Spearman's correlation from 76.25% to 78.49%. For the RoBERTa model, DiffCSE-RoBERTa$_{\texttt{base}}$ can also improve upon SimCSE-RoBERTa$_{\texttt{base}}$ from 76.57% to 77.80%.

Transfer Tasks We show the results of transfer tasks in Table 2. Compared with SimCSE-BERT$_{\texttt{base}}$, DiffCSE-BERT$_{\texttt{base}}$ can improve the averaged scores from 85.56% to 86.86%. When applying it to the RoBERTa model, DiffCSE-RoBERTa$_{\texttt{base}}$ also improves upon SimCSE-RoBERTa$_{\texttt{base}}$ from 84.84% to 87.04%. Note that the CMLM-BERT$_{\texttt{base}}$ (Yang et al., 2020) can achieve even better performance than DiffCSE. However, they use 1TB of the training data from Common Crawl dumps while our model only use 115MB of the Wikipedia data for pretraining. We put their scores in Table 2 for reference. In SimCSE, the authors propose to use MLM as an auxiliary task for the sentence encoder to further boost the performance of transfer tasks. Compared with

[3] `https://github.com/princeton-nlp/SimCSE`

[4] `https://github.com/facebookresearch/SentEval`

Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT$_{\texttt{base}}$ (first-last avg.)♢	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT$_{\texttt{base}}$-flow♢	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT$_{\texttt{base}}$-whitening♢	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT$_{\texttt{base}}$ ♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CMLM-BERT$_{\texttt{base}}$ ♠ (1TB data)	58.20	61.07	61.67	73.32	74.88	76.60	64.80	67.22
CT-BERT$_{\texttt{base}}$ ♢	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SG-OPT-BERT$_{\texttt{base}}$ †	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
SimCSE-BERT$_{\texttt{base}}$ ♢	68.40	82.41	74.38	80.91	78.56	76.85	**72.23**	76.25
∗ SimCSE-BERT$_{\texttt{base}}$(reproduce)	70.82	82.24	73.25	81.38	77.06	77.24	71.16	76.16
∗ DiffCSE-BERT$_{\texttt{base}}$	**72.28**	**84.43**	**76.47**	**83.90**	**80.54**	**80.59**	71.23	**78.49**
RoBERTa$_{\texttt{base}}$ (first-last avg.)♢	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
RoBERTa$_{\texttt{base}}$-whitening♢	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-RoBERTa$_{\texttt{base}}$ ♢	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
SimCSE-RoBERTa$_{\texttt{base}}$ ♢	**70.16**	81.77	73.24	81.36	80.65	80.22	68.56	76.57
∗ SimCSE-RoBERTa$_{\texttt{base}}$(reproduce)	68.60	81.36	73.16	81.61	80.76	80.58	68.83	76.41
∗ DiffCSE-RoBERTa$_{\texttt{base}}$	70.05	**83.43**	**75.49**	**82.81**	**82.12**	**82.38**	**71.19**	**78.21**

Table 1: The performance on STS tasks (Spearman's correlation) for different sentence embedding models. ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020); ♢: results from Gao et al. (2021); ♠: results from Yang et al. (2020); †: results from Kim et al. (2021); ∗: results from our experiments.

the results of SimCSE with MLM, DiffCSE still can have a little improvement around 0.2%.

5 Ablation Studies

In the following sections, we perform an extensive series of ablation studies that support our model design. We use BERT$_{\texttt{base}}$ model to evaluate on the development set of STS-B and transfer tasks.

Removing Contrastive Loss In our model, both the contrastive loss and the RTD loss are crucial because they maintain what should be sensitive and what should be insensitive respectively. If we remove the RTD loss, the model becomes a SimCSE model; if we remove the contrastive loss, the performance of STS-B drops significantly by 30%, while the average score of transfer tasks also drops by 2% (see Table 3). This result shows that it is important to have insensitive and sensitive attributes that exist together in the representation space.

Next Sentence vs. Same Sentence Some methods for unsupervised sentence embeddings like Quick-Thoughts (Logeswaran and Lee, 2018) and CMLM (Yang et al., 2020) predict the next sentence as the training objective. We also experiment with a variant of DiffCSE by conditioning the ELECTRA loss based on the next sentence. Note that this kind of model is not doing a "diff operation" between two similar sentences, and is not an instance of equivariant contrastive learning. As shown in Table 3 (use next sent. for x'), the score of STS-B decreases significantly compared to DiffCSE while transfer performance remains similar. We also tried using the same sentence and the next sentence at the same time for conditioning the ELECTRA objective (use same+next sent. for x'), and did not observe improvements.

Other Conditional Pretraining Tasks Instead of a conditional binary difference prediction loss, we can also consider other conditional pretraining tasks such as a conditional MLM objective proposed by Yang et al. (2020), or corrective language modeling,[5] proposed by COCO-LM (Meng et al., 2021). We experiment with these objectives instead of the difference prediction objective in Table 3. We observe that conditional MLM on the same sentence does not improve the performance either on STS-B or transfer tasks compared with DiffCSE. Conditional MLM on the next sentence performs even worse for STS-B, but slightly better than using the same sentence on transfer tasks. Using both the same and the next sentence also does not improve the performance compared with DiffCSE. For the corrective LM objective, the performance of STS-B decreases significantly compared with DiffCSE.

Augmentation Methods: Insert/Delete/Replace In DiffCSE, we use MLM token replacement as the equivariant augmentation. It is possible to use other methods like random insertion or deletion instead of replacement.[6] For insertion, we choose to

[5] This task is similar to ELECTRA. However, instead of a binary classifier for replaced token detection, corrective LM uses a vocabulary-size classifier with the copy mechanism to recover the replaced tokens.

[6] Edit distance operators include *insert, delete* and *replace*.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
GloVe embeddings (avg.)♣	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought♡	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings♣	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT-`[CLS]`embedding♣	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT$_{\texttt{base}}$ ♡	81.09	87.18	94.96	88.75	85.96	88.64	74.24	85.83
SimCSE-BERT$_{\texttt{base}}$ ♢	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
w/ MLM	82.92	87.23	95.71	88.73	86.81	87.01	78.07	86.64
* DiffCSE-BERT$_{\texttt{base}}$	**82.69**	**87.23**	**95.23**	**89.28**	**86.60**	90.40	**76.58**	**86.86**
CMLM-BERT$_{\texttt{base}}$(1TB data)	83.60	89.90	96.20	89.30	88.50	91.00	69.70	86.89
SimCSE-RoBERTa$_{\texttt{base}}$ ♢	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
w/ MLM	**83.37**	87.76	**95.05**	87.16	**89.02**	**90.80**	75.13	86.90
* DiffCSE-RoBERTa$_{\texttt{base}}$	82.82	**88.61**	94.32	**87.71**	88.63	90.40	**76.81**	**87.04**

Table 2: Transfer task results of different sentence embedding models (measured as accuracy). ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020); ♢: results from Gao et al. (2021).

	STS-B	Avg. transfer
SimCSE	81.47	83.91
DiffCSE	**84.56**	**85.95**
w/o contrastive loss	54.48	83.46
use next sent. for x'	82.91	85.83
use same+next sent. for x'	83.41	85.82
Conditional MLM		
for same sent.	83.08	84.43
for next sent.	75.82	85.68
for same+next sent.	82.88	84.82
Conditional Corrective LM	79.79	85.30

Table 3: Development set results of STS-B and transfer tasks for DiffCSE model variants, where we vary the objective and the use of same or next sentence.

Augmentation	STS-B	Avg. transfer
MLM 15%	**84.48**	85.95
randomly insert 15%	82.20	85.96
randomly delete 15%	82.59	**85.97**
combining all	82.80	85.92

Table 4: Development set results of STS-B and transfer tasks with different augmentation methods for learning equivariance.

	STS-B	Avg. transfer
DiffCSE		
w/ BatchNorm	**84.56**	**85.95**
w/o BatchNorm	83.23	85.24
SimCSE		
w/ BatchNorm	82.22	85.66
w/o BatchNorm	81.47	83.91

Table 5: Development set results of STS-B and transfer tasks for DiffCSE and SimCSE with and without BatchNorm.

randomly insert mask tokens to the sentence, and then use a generator to convert mask tokens into real tokens. The number of inserted masked tokens is 15% of the sentence length. The task is to predict whether a token is an inserted token or the original token. For deletion, we randomly delete 15% tokens in the sentence, and the task is to predict for each token whether a token preceding it has been deleted or not. The results are shown in Table 4. We can see that using either insertion or deletion achieves a slightly worse STS-B performance than using MLM replacement. For transfer tasks, their results are similar. Finally, we find that combining all three augmentations in the training process does not improve the MLM replacement strategy.

Pooler Choice In SimCSE, the authors use the pooler in BERT's original implementation (one linear layer with tanh activation function) as the final layer to extract features for computing contrastive loss. In our implementation (see details in Appendix A), we find that it is better to use a two-layer pooler with Batch Normalization (BatchNorm) (Ioffe and Szegedy, 2015), which is commonly used in contrastive learning framework in computer vision (Chen et al., 2020; Grill et al., 2020; Chen and He, 2021; Hua et al., 2021). We show the ablation results in Table 5. We can observe that adding BatchNorm is beneficial for either DiffCSE or SimCSE to get better performance on STS-B and transfer tasks.

Size of the Generator In our DiffCSE model, the generator can be in different model size from BERT$_{\texttt{large}}$, BERT$_{\texttt{base}}$ (Devlin et al., 2019), DistilBERT$_{\texttt{base}}$ (Sanh et al., 2019), BERT$_{\texttt{medium}}$, BERT$_{\texttt{small}}$, BERT$_{\texttt{mini}}$, BERT$_{\texttt{tiny}}$ (Turc et al., 2019). Their exact sizes are shown in Table 6 (L: number of layers, H: hidden dimension). Notice that although DistilBERT$_{\texttt{base}}$ has only half the number of layers of BERT, it can retain 97% of

	STS-B	Avg. transfer
SimCSE	81.47	83.91
DiffCSE w/ generator:		
BERT$_{\texttt{large}}$ (L=24, H=1024)	82.93	85.88
BERT$_{\texttt{base}}$ (L=12, H=768)	83.63	85.85
DistilBERT$_{\texttt{base}}$ (L=6, H=768)	**84.56**	**85.95**
BERT$_{\texttt{medium}}$ (L=8, H=512)	82.25	85.80
BERT$_{\texttt{small}}$ (L=4, H=512)	82.64	85.66
BERT$_{\texttt{mini}}$ (L=4, H=256)	82.12	85.90
BERT$_{\texttt{tiny}}$ (L=2, H=128)	81.40	85.23

Table 6: Development set results of STS-B and transfer tasks with different generators.

Ratio	*15%*	*20%*	*25%*	*30%*	*40%*	*50%*
STS-B	84.48	84.04	84.49	**84.56**	84.48	83.91

Table 7: Development set results of STS-B under different masking ratio for augmentations.

BERT's performance due to knowledge distillation.

We show our results in Table 6, we can see the performance of transfer tasks does not change much with different generators. However, the score of STS-B decreases as we switch from BERT-medium to BERT-tiny. This finding is not the same as ELECTRA, which works best with generators 1/4-1/2 the size of the discriminator. Because our discriminator is conditional on sentence vectors, it will be easier for the discriminator to perform the RTD task. As a result, using stronger generators (BERT$_{\texttt{base}}$, DistilBERT$_{\texttt{base}}$) to increase the difficulty of RTD would help the discriminator learn better. However, when using a large model like BERT$_{\texttt{large}}$, it may be a too-challenging task for the discriminator. In our experiment, using DistilBERT$_{\texttt{base}}$, which has the ability close to but slightly worse than BERT$_{\texttt{base}}$, gives us the best performance.

Masking Ratio In our conditional ELECTRA task, we can mask the original sentence in different ratios for the generator to produce MLM-based augmentations. A higher masking ratio will make more perturbations to the sentence. Our empirical result in Table 7 shows that the difference between difference masking ratios is small (in 15%-40%), and a masking ratio of around 30% can give us the best performance.

Coefficient λ In Section 3, we use the λ coefficient to weight the ELECTRA loss and then add it with contrastive loss. Because the contrastive learning objective is a relatively easier task, the scale of contrastive loss will be 100 to 1000 smaller than ELECTRA loss. As a result, we need a smaller λ to balance these two loss terms. In the Table 8 we show the STS-B result under different λ values. Note that when λ goes to zero, the model becomes a SimCSE model. We find that using $\lambda = 0.005$ can give us the best performance.

λ	*0*	*0.0001*	*0.0005*	*0.001*
STS-B	82.22	83.90	84.40	84.24
λ	*0.005*	*0.01*	*0.05*	*0.1*
STS-B	**84.56**	83.44	84.11	83.66

Table 8: Development set results of STS-B under different λ.

6 Analysis

6.1 Qualitative Study

A very common application for sentence embeddings is the retrieval task. Here we show some retrieval examples to qualitatively explain why DiffCSE can perform better than SimCSE. In this study, we use the 2758 sentences from STS-B testing set as the corpus, and then use sentence query to retrieve the nearest neighbors in the sentence embedding space by computing cosine similarities. We show the retrieved top-3 examples in Table 9. The first query sentence is "you can do it, too.". The SimCSE model retrieves a very similar sentence but has a slightly different meaning ("you can use it, too.") as the rank-1 answer. In contrast, DiffCSE can distinguish the tiny difference, so it retrieves the ground truth answer as the rank-1 answer. The second query sentence is "this is not a problem". SimCSE retrieves a sentence with opposite meaning but very similar wording, while DiffCSE can retrieve the correct answer with less similar wording. We also provide a third example where both SimCSE and DiffCSE fail to retrieve the correct answer for a query sentence using double negation.

6.2 Retrieval Task

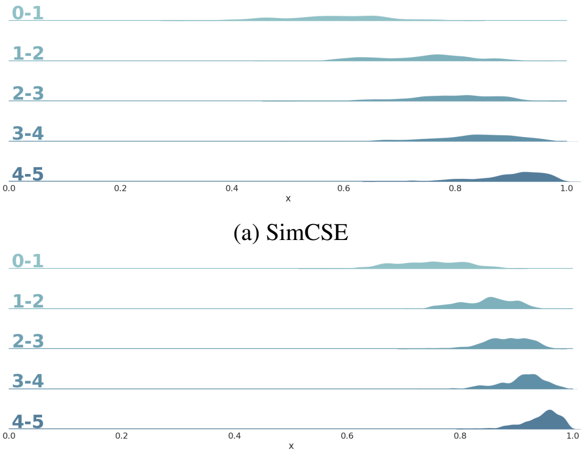
Besides the qualitative study, we also show the quantitative result of the retrieval task. Here we also use all the 2758 sentences in the testing set of STS-B as the corpus. There are 97 positive pairs in this corpus (with 5 out of 5 semantic similarity scores from human annotation). For each positive pair, we use one sentence to retrieve the other one, and see whether the other sentence is in the top-1/5/10 ranking. The recall@1/5/10 of the retrieval task are shown in Table 10. We can observe that DiffCSE can outperform SimCSE for

SimCSE-BERT$_{\text{base}}$	**DiffCSE-BERT$_{\text{base}}$**
Query: you can do it, too.	
1) you can use it, too. 2) can you do it? 3) yes, you can do it.	1) yes, you can do it. 2) you can use it, too. 3) can you do it?
Query: this is not a problem.	
1) this is a big problem. 2) you have a problem. 3) i don 't see why that should be a problem.	1) i don 't see why this could be a problem. 2) i don 't see why that should be a problem. 3) this is a big problem.
Query: i think that is not a bad idea.	
1) i do not think it's a good idea. 2) it's not a good idea . 3) it is not a good idea .	1) i do not think it's a good idea . 2) it is not a good idea. 3) but it is not a good idea.

Table 9: Retrieved top-3 examples by SimCSE and DiffCSE from STS-B test set.

Model/Recall	**@1**	**@5**	**@10**
SimCSE-BERT$_{\text{base}}$	77.84	92.78	95.88
DiffCSE-BERT$_{\text{base}}$	**78.87**	**95.36**	**97.42**

Table 10: The retrieval results for SimCSE and DiffCSE.



(a) SimCSE

(b) DiffCSE

Figure 2: The distribution of cosine similarities from SimCSE/DiffCSE for STS-B test set. Along the y-axis are 5 groups of data splits based on human ratings. The x-axis is the cosine similarity.

recall@1/5/10, showing the effectiveness of using DiffCSE for the retrieval task.

6.3 Distribution of Sentence Embeddings

To look into the representation space of DiffCSE, we plot the cosine similarity distribution of sentence pairs from STS-B test set for both SimCSE and DiffCSE in Figure 2. We observe that both SimCSE and DiffCSE can assign cosine similarities consistent with human ratings. However, we also observe that under the same human rating, DiffCSE assigns slightly higher cosine similarities compared with SimCSE. This phenomenon may be caused by the fact that ELECTRA and other Transformer-based pretrained LMs have the problem of squeezing the representation space, as mentioned by Meng et al. (2021). As we use the sentence embeddings as the input of ELECTRA to perform conditional ELECTRA training, the sentence embedding will be inevitably squeezed to fit the input distribution of ELECTRA. We follow prior studies (Wang and Isola, 2020; Gao et al., 2021) to use *uniformity* and *alignment* (details in Appendix C) to measure the quality of representation space for DiffCSE and SimCSE in Table 11. Compared to averaged BERT embeddings, SimCSE has similar alignment (0.177 v.s. 0.172) but better uniformity (-2.313). In contrast, DiffCSE has similar uniformity as Avg. BERT (-1.438 v.s. -1.468) but much better alignment (0.097). It indicates that SimCSE and DiffCSE are optimizing the representation space in two different directions. And the improvement of DiffCSE may come from its better alignment.

Model	**Alignment**	**Uniformity**	**STS**
Avg. BERT$_{\text{base}}$	0.172	-1.468	56.70
SimCSE-BERT$_{\text{base}}$	0.177	**-2.313**	76.16
DiffCSE-BERT$_{\text{base}}$	**0.097**	-1.438	**78.49**

Table 11: *Alignment* and *Uniformity* (Wang and Isola, 2020) measured on STS-B test set for SimCSE and DiffCSE. The smaller the number is better. We also show the averaged STS score in the right-most column.

7 Conclusion

In this paper, we present DiffCSE, a new unsupervised sentence embedding framework that is aware of, but not invariant to, MLM-based word replacement. Empirical results on semantic textual similarity tasks and transfer tasks both show the effectiveness of DiffCSE compared to current state-of-the-art sentence embedding methods. We also conduct extensive ablation studies to demonstrate the different modeling choices in DiffCSE. Qualitative study and the retrieval results also show that DiffCSE can produce a better embedding space for sentence retrieval. One limitation of our work is that we do not explore the supervised setting that uses human-labeled NLI datasets to further boost the performance. We leave this topic for future work. We believe that our work can provide researchers in the NLP community a new way to utilize augmentations for natural language and thus produce better sentence embeddings.

Acknowledgements

In this research, Yung-Sung and Hongyin were partially supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Fund (InnoHK).

References

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.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension.

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, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. 2021. Equivariant contrastive learning. *arXiv preprint arXiv:2111.00899*.

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.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. 2020. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. 2021. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for bert sentence representations. *arXiv preprint arXiv:2106.07345*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. pages 3294–3302.

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.

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

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2020. Universal sentence representation learning with conditional masked language model. *arXiv preprint arXiv:2012.14388*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.

A Training Details

We use a single NVIDIA 2080Ti GPU for each experiment. The averaged running time for DiffCSE is 3-6 hours. We use grid-search of batch size $\in \{64, 128\}$ learning rate $\in$ {2e-6, 3e-6, 5e-6, 7e-6, 1e-5} and masking ratio $\in \{0.15, 0.20, 0.30, 0.40\}$ and $\lambda \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$. The temperature τ in SimCSE is set to 0.05 for all the experiments. During the training process, we save the checkpoint with the highest score on the STS-B development set. And then we use STS-B development set to find the best hyperparameters (listed in Table 12) for STS task; we use the averaged score of the development sets of 7 transfer tasks to find the best hyperparameters (listed in Table 13) for transfer tasks. All numbers in Table 1 and Table 2 are from a single run.

hyperparam	BERT$_{\text{base}}$	RoBERTa$_{\text{base}}$
learning rate	7e-6	1e-5
masking ratio	0.30	0.20
λ	0.005	0.005
training epochs	2	2
batch size	64	64

Table 12: The main hyperparameters in STS tasks.

hyperparam	BERT$_{\text{base}}$	RoBERTa$_{\text{base}}$
learning rate	2e-6	3e-6
masking ratio	0.15	0.15
λ	0.05	0.05
training epochs	2	2
batch size	64	128

Table 13: The main hyperparameters in transfer tasks.

Method	BERT$_{\text{base}}$	RoBERTa$_{\text{base}}$
SimCSE	110M	125M
DiffCSE (train)	220M	250M
DiffCSE (test)	110M	125M

Table 14: The number of parameters used in our models.

During testing, we follow SimCSE to discard the MLP projector and only use the `[CLS]` output to extract the sentence embeddings.

The numbers of model parameters for BERT$_{\text{base}}$ and RoBERTa$_{\text{base}}$ are listed in Table 14. Note that in training time DiffCSE needs two BERT models to work together (sentence encoder + discriminator), but in testing time we only need the sentence encoder, so the model size is the same as the SimCSE model.

Method	STS-B	Avg. transfer
SimCSE	81.47	83.91
+ Additional positives		
MLM 15%	73.59	83.33
random insert 15%	80.39	83.92
random delete 15%	78.58	81.80
+ Additional negatives		
MLM 15%	83.02	84.49
random insert 15%	55.65	79.86
random delete 15%	55.13	82.56
+ Equivariance (Ours)		
MLM 15%	**84.48**	85.95
randomly insert 15%	82.20	85.96
randomly delete 15%	82.59	**85.97**

Table 15: Development set results of STS-B and transfer tasks for using three types of augmentations (replace, insert, delete) in different ways.

Projector with BatchNorm In Section 5, we mention that we use a projector with BatchNorm as the final layer of our model. Here we provided the PyTorch code for its structure:

```
class ProjectionMLP(nn.Module):
   def __init__(self, hidden_size):
      super().__init__()
      in_dim = hidden_size
      middle_dim = hidden_size * 2
      out_dim = hidden_size
      self.net = nn.Sequential(
      nn.Linear(in_dim, middle_dim,
          bias=False),
      nn.BatchNorm1d(middle_dim),
      nn.ReLU(inplace=True),
      nn.Linear(middle_dim, out_dim,
          bias=False),
      nn.BatchNorm1d(out_dim,
          affine=False))
```

B Using Augmentations as Positive/Negative Examples

In Section 5, we try to use different augmentations (e.g. insertion, deletion, replacement) for learning equivariance. In Table 15 we provide the results of using these augmentations as additional positive or negative examples along with the SimCSE training paradigm. We can observe that using these augmentations as additional positives only decreases the performance. The only method that can improve the performance a little bit is to use MLM 15% replaced examples as additional negative examples. Overall, none of these results can perform better than our proposed method, e.g. using these augmentations to learn equivariance.

C Uniformity and Alignment

Wang and Isola (2020) propose to use two properties, *alignment* and *uniformity*, to measure the quality of representations. Given a distribution of positive pairs p_{pos} and the distribution of the whole dataset p_{data}, *alignment* computes the expected distance between normalized embeddings of the paired sentences:

$$\ell_{\text{align}} \triangleq \underset{(x,x^+)\sim p_{\text{pos}}}{\mathbb{E}} \left\|f(x) - f\left(x^+\right)\right\|^2 .$$

Uniformity measures how well the embeddings are uniformly distributed in the representation space:

$$\ell_{\text{uniform}} \triangleq \log \underset{x,y \overset{i.i.d.}{\sim} p_{\text{data}}}{\mathbb{E}} e^{-2\|f(x)-f(y)\|^2} .$$

The smaller the values of uniformity and alignment, the better the quality of the representation space is indicated.

D Source Code

We build our model using the PyTorch implementation of SimCSE[7] Gao et al. (2021), which is based on the HuggingFace's Transformers package.[8] We also upload our code[9] and pretrained models (links in `README.md`). Please follow the instructions in `README.md` to reproduce the results.

E Potential Risks

On the risk side, insofar as our method utilizes pretrained language models, it may inherit and propagate some of the biases present in such models. Besides that, we do not see any other potential risks in our paper.

[7] https://github.com/princeton-nlp/SimCSE

[8] https://github.com/huggingface/transformers

[9] https://github.com/voidism/DiffCSE