# HiddenCut: Simple Data Augmentation for Natural Language Understanding with Better Generalization

**Jiaao Chen, Dinghan Shen[1], Weizhu Chen[1], Diyi Yang**
Georgia Institute of Technology, [1]Microsoft Dynamics 365 AI
{jchen896,dyang888}@gatech.edu
{dishen,wzchen}@microsoft.com

## Abstract

Fine-tuning large pre-trained models with task-specific data has achieved great success in NLP. However, it has been demonstrated that the majority of information within the self-attention networks is redundant and not utilized effectively during the fine-tuning stage. This leads to inferior results when generalizing the obtained models to out-of-domain distributions. To this end, we propose a simple yet effective data augmentation technique, HiddenCut, to better regularize the model and encourage it to learn more generalizable features. Specifically, contiguous spans within the hidden space are dynamically and strategically dropped during training. Experiments show that our HiddenCut method outperforms the state-of-the-art augmentation methods on the GLUE benchmark, and consistently exhibits superior generalization performances on out-of-distribution and challenging counterexamples. We have publicly released our code at https://github.com/GT-SALT/HiddenCut.

## 1 Introduction

Fine-tuning large-scale pre-trained language models (PLMs) has become a dominant paradigm in the natural language processing community, achieving state-of-the-art performances in a wide range of natural language processing tasks (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019a; Joshi et al., 2019; Sun et al., 2019; Clark et al., 2019; Lewis et al., 2020; Bao et al., 2020; He et al., 2020; Raffel et al., 2020). Despite the great success, due to the huge gap between the number of model parameters and that of task-specific data available, the majority of the information within the multi-layer self-attention networks is typically redundant and ineffectively utilized for downstream tasks (Guo et al., 2020; Gordon et al., 2020; Dalvi et al., 2020).

As a result, after task-specific fine-tuning, models are very likely to overfit and make predictions based on spurious patterns (Tu et al., 2020; Kaushik et al., 2020), making them less generalizable to out-of-domain distributions (Zhu et al., 2019; Jiang et al., 2019; Aghajanyan et al., 2020).

In order to improve the generalization abilities of over-parameterized models with limited amount of task-specific data, various regularization approaches have been proposed, such as adversarial training that injects label-preserving perturbations in the input space (Zhu et al., 2019; Liu et al., 2020; Jiang et al., 2019), generating augmented data via carefully-designed rules (McCoy et al., 2019; Xie et al., 2020; Andreas, 2020; Shen et al., 2020), and annotating counterfactual examples (Goyal et al., 2019; Kaushik et al., 2020). Despite substantial improvements, these methods often require significant computational and memory overhead (Zhu et al., 2019; Liu et al., 2020; Jiang et al., 2019; Xie et al., 2020) or human annotations (Goyal et al., 2019; Kaushik et al., 2020).

In this work, to alleviate the above issues, we rethink the simple and commonly-used regularization technique—dropout (Srivastava et al., 2014)—in pre-trained transformer models (Vaswani et al., 2017). With multiple self-attention heads in transformers, dropout converts some hidden units to zeros in a random and independent manner. Although PLMs have already been equipped with the dropout regularization, they still suffer from inferior performances when it comes to out-of-distribution cases (Tu et al., 2020; Kaushik et al., 2020). The underlying reasons are two-fold: (1) the linguistic relations among words in a sentence is ignored while dropping the hidden units randomly. In reality, these masked features could be easily inferred from surrounding unmasked hidden units with the self-attention networks. Therefore, redundant information still exists and gets passed to the upper

4380

layers. (2) The standard dropout assumes that every hidden unit is equally important with the random sampling procedure, failing to characterize the different roles these features play in distinct tasks. As a result, the learned representations are not generalized enough while applied to other data and tasks. To drop the information more effectively, Shen et al. (2020) recently introduce Cutoff to remove tokens/features/spans in the input space. Even though models will not see the removed information during training, examples with large noise may be generated when key clues for predictions are completely removed from the input.

To overcome these limitations, we propose a simple yet effective data augmentation method, **HiddenCut**, to regularize PLMs during the fine-tuning stage. Specifically, the approach is based on the *linguistic* intuition that hidden representations of adjacent words are more likely to contain similar and redundant information. HiddenCut drops hidden units more structurally by masking the whole hidden information of *contiguous spans* of tokens after every encoding layer. This would encourage models to fully utilize all the task-related information, instead of learning spurious patterns during training. To make the dropping process more efficient, we *dynamically* and *strategically* select the informative spans to drop by introducing an attention-based mechanism. By performing HiddenCut in the hidden space, the impact of dropped information is only mitigated rather than completely removed, avoiding injecting too much noise to the input. We further apply a Jensen-Shannon Divergence consistency regularization between the original and these augmented examples to model the consistent relations between them.

To demonstrate the effectiveness of our methods, we conduct experiments to compare our HiddenCut with previous state-of-the-art data augmentation method on 8 natural language understanding tasks from the GLUE (Wang et al., 2018) benchmark for in-distribution evaluations, and 5 challenging datasets that cover single-sentence tasks, similarity and paraphrase tasks and inference tasks for *out-of-distribution* evaluations. We further perform ablation studies to investigate the impact of different selecting strategies on HiddenCut's effectiveness. Results show that our method consistently outperforms baselines, especially on out-of-distribution and challenging counterexamples. To sum up, our contributions are:

- We propose a simple data augmentation method, HiddenCut, to regularize PLMs during fine-tuning by cutting contiguous spans of representations in the hidden space.

- We explore and design different strategic sampling techniques to dynamically and adaptively construct the set of spans to be cut.

- We demonstrate the effectiveness of Hidden-Cut through extensive experiments on both in-distribution and out-of-distribution datasets.

## 2 Related Work

### 2.1 Adversarial Training

Adversarial training methods usually regularize models through applying perturbations to the input or hidden space (Szegedy et al., 2013; Goodfellow et al., 2014; Madry et al., 2017) with additional forward-backward passes, which influence the model's predictions and confidence without changing human judgements. Adversarial-based approaches have been actively applied to various NLP tasks in order to improve models' robustness and generalization abilities, such as sentence classification (Miyato et al., 2017), machine reading comprehension (MRC) (Wang and Bansal, 2018) and natural language inference (NLI) tasks (Nie et al., 2020). Despite its success, adversarial training often requires extensive computation overhead to calculate the perturbation directions (Shafahi et al., 2019; Zhang et al., 2019a). In contrast, our HiddenCut adds perturbations in the hidden space in a more efficient way that does not require extra computations as the designed perturbations can be directly derived from self-attentions.

### 2.2 Data Augmentation

Another line of work to improve the model robustness is to directly design data augmentation methods to enrich the original training set such as creating syntactically-rich examples (McCoy et al., 2019; Min et al., 2020) with specific rules, crowdsourcing counterfactual augmentation to avoid learning spurious features (Goyal et al., 2019; Kaushik et al., 2020), or combining examples in the dataset to increase compositional generalizabilities (Jia and Liang, 2016; Andreas, 2020; Chen et al., 2020b,a). However, they either require careful design (McCoy et al., 2019; Andreas, 2020) to infer labels for generated data or extensive human annotations (Goyal et al., 2019; Kaushik et al., 2020),

which makes them hard to generalize to different tasks/datasets. Recently Shen et al. (2020) introduce a set of cutoff augmentation which directly creates partial views to augment the training in a more task-agnostic way. Inspired by these prior work, our HiddenCut aims at improving models' generalization abilities to out-of-distribution via linguistic-informed strategically dropping spans of hidden information in transformers.

## 2.3 Dropout-based Regularization

Variations of dropout (Srivastava et al., 2014) have been proposed to regularize neural models by injecting noise through dropping certain information so that models do not overfit training data. However, the major efforts have been put to convolutional neural networks and trimmed for structures in images recently such as DropPath (Larsson et al., 2017), DropBlock (Ghiasi et al., 2018), DropCluster (Chen et al., 2020c) and AutoDropout (Pham and Le, 2021). In contrast, our work takes a closer look at transformer-based models and introduces HiddenCut for natural language understanding tasks. HiddenCut is closely related to DropBlock (Ghiasi et al., 2018), which drops contiguous regions from a feature map. However, different from images, hidden dimensions in PLMs that contain syntactic/semantic information for NLP tasks are more closely related (e.g., NER and POS information), and simply dropping spans of features in certain hidden dimensions might still lead to information redundancy.

## 3 HiddenCut Approach

To regularize transformer models in a more structural and efficient manner, in this section, we introduce a simple yet effective data augmentation technique, **HiddenCut**, that reforms dropout to cutting contiguous spans of hidden representations after each transformer layer (Section 3.1). Intuitively, the proposed approach encourages the models to fully utilize all the hidden information within the self-attention networks. Furthermore, we propose an attention-based mechanism to strategically and judiciously determine the specific spans to cut (Section 3.2). The schematic diagram of HiddenCut, applied to the transformer architecture (and its comparison to dropout) are shown in Figure 1.

## 3.1 HiddenCut

For an input sequence $\mathbf{s} = \{w_0, w_1, ..., w_L\}$ with $L$ tokens associated with a label $y$, we employ a pre-trained transformer model $f_{1:M}(\cdot)$ with $M$ layers like RoBERTa (Liu et al., 2019) to encode the text into hidden representations. Thereafter, an inference network $g(\cdot)$ is learned on top of the pre-trained models to predict the corresponding labels. In the hidden space, after layer $m$, every word $w_i$ in the input sequence is encoded into a $D$ dimensional vector $h_i^m \in \mathrm{R}^D$ and the whole sequence could be viewed as a hidden matrix $\mathbf{H}^m \in \mathrm{R}^{L \times D}$.

With multiple self-attention heads in the transformer layers, it is found that there is extensive redundant information across $h_i^m \in \mathbf{H}$ that are linguistically related (Dalvi et al., 2020) (e.g., words that share similar semantic meanings). As a result, the removed information from the standard dropout operation may be easily inferred from the remaining unmasked hidden units. The resulting model might easily overfit to certain high-frequency features without utilizing all the important task-related information in the hidden space (especially when task-related data is limited). Moreover, the model also suffers from poor generalization ability while being applied to out-of-distribution cases.

Inspired by Ghiasi et al. (2018); Shen et al. (2020), we propose to improve the dropout regularization in transformer models by creating augmented training examples through HiddenCut, which drops a contiguous span of hidden information encoded in every layer, as shown in Figure 1 (c). Mathematically, in every layer $m$, a span of hidden vectors, $\mathbf{S} \in \mathrm{R}^{l \times D}$, with length $l = \alpha L$ in the hidden matrix $\mathbf{H}^m \in \mathrm{R}^{L \times D}$ are converted to 0, and the corresponding attention masks are adjusted to 0, where $\alpha$ is a pre-defined hyper-parameter indicating the dropping extent of HiddenCut. After being encoded and hiddencut through all the hidden layers in pre-trained encoders, augmented training data $f^{\mathrm{HiddenCut}}(\mathbf{s})$ is created for learning the inference network $g(\cdot)$ to predict task labels.

## 3.2 Strategic Sampling

Different tasks rely on learning distinct sets of information from the input to predict the corresponding task labels. Performing HiddenCut randomly might be inefficient especially when most of the dropping happens at task-unrelated spans, which fails to effectively regularize model to take advantage of all the task-related features. To this end, we
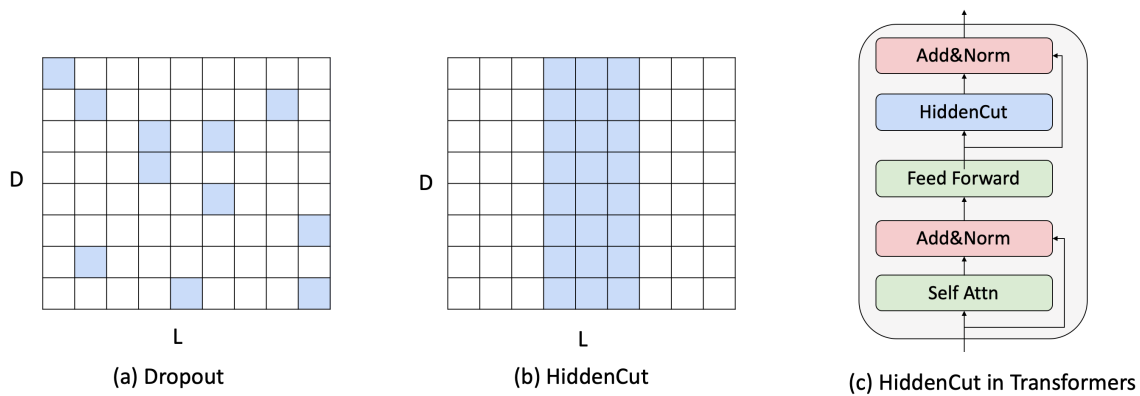
Figure 1: Illustration of the differences between Dropout (a) and HiddenCut (b), and the position of HiddenCut in transformer layers (c). A sentence in the hidden space can be viewed as a $L \times D$ matrix where $L$ is the length of the sentence and $D$ is the number of hidden dimensions. The cells in blue represent that they are masked. Dropout masks random independent units in the matrix while our HiddenCut selects and masks a whole span of hidden representations based on attention weights received in the current layer. In our experiments, we perform HiddenCut after the feed-forward network in every transformer layer.

propose to select the spans to be cut *dynamically* and *strategically* in every layer. In other words, we mask the most informative span of hidden representations in one layer to force models to discover other useful clues to make predictions instead of relying on a small set of spurious patterns.

**Attention-based Sampling Strategy**   The most direct way is to define the set of tokens to be cut by utilizing attention weights assigned to tokens in the self-attention layers (Kovaleva et al., 2019). Intuitively, we can drop the spans of hidden representations that are assigned high attentions by the transformer layers. As a result, the information redundancy is alleviated and models would be encourage to attend to other important information. Specifically, we first derive the average attention for each token, $a_i$, from the attention weights matrix $\mathbf{A} \in \mathrm{R}^{P \times L \times L}$ after self-attention layers, where $P$ is the number of attention heads and $L$ is the sequence length:

$$a_i = \frac{\sum_j^P (\sum_k^L A[j][k][i])}{P}.$$

We then sample the start token $h_i$ for HiddenCut from the set that contains top $\beta L$ tokens with higher average attention weights ($\beta$ is a pre-defined parameter). Then HiddenCut is performed to mask the hidden representations between $h_i$ and $h_{i+l}$. Note that the salient sets are different across different layers and updated throughout the training.

**Other Sampling Strategies**   We also explore other widely used word importance discovery methods to find a set of tokens to be strategically cut by HiddenCut, including:

- **Random**: All spans of tokens are viewed as equally important, thus are randomly cut.

- **LIME** (Ribeiro et al., 2016) defines the importance of tokens by examining the locally faithfulness where weights of tokens are assigned by classifiers trained with sentences whose words are randomly removed. We utilized LIME on top of a SVM classifier to pre-define a fixed set of tokens to be cut.

- **GEM** (Yang et al., 2019b) utilizes orthogonal basis to calculate the *novelty scores* that measure the new semantic meaning in tokens, *significance scores* that estimate the alignment between the semantic meaning of tokens and the sentence-level meaning, and the *uniqueness scores* that examine the uniqueness of the semantic meaning of tokens. We compute the GEM scores using the hidden representations at every layer to generate the set of tokens to be cut, which are updated during training.

- **Gradient** (Baehrens et al., 2010): We define the set of tokens to be cut based on the rankings of the absolute values of gradients they received at every layer in the backward-passing. This set would be updated during training.

## 3.3 Objectives

During training, for an input text sequence $\mathbf{s}$ with a label $y$, we generate $N$ augmented examples $\{f_1^{\text{HiddenCut}}(\mathbf{s}), ..., f_N^{\text{HiddenCut}}(\mathbf{s})\}$ through performing HiddenCut in pre-trained encoder $f(\cdot)$. The whole model $g(f(\cdot))$ is then trained though several objectives including general classification loss ($\mathcal{L}_{ori}$ and $\mathcal{L}_{aug}$) on data-label pairs and consistency regularization ($\mathcal{L}_{js}$) (Miyato et al., 2017, 2018; Clark et al., 2018; Xie et al., 2019; Shen et al., 2020) across different augmentations:

$$\mathcal{L}_{ori} = \mathbf{CE}(g(f(\mathbf{s})), y)$$
$$\mathcal{L}_{aug} = \sum_N \mathbf{CE}(g(f_i^{\text{HiddenCut}}(\mathbf{s})), y)$$
$$\mathcal{L}_{js} = \sum_N \mathbf{KL}[p(y|g(f_i^{\text{HiddenCut}}(\mathbf{s}))||p_{avg}]$$

where $\mathbf{CE}$ and $\mathbf{KL}$ represent the cross-entropy loss and KL-divergence respectively. $p_{avg}$ stands for the average predictions across the original text and all the augmented examples.

Combining these three losses, our overall objective function is:

$$\mathcal{L} = \mathcal{L}_{ori} + \gamma\mathcal{L}_{aug} + \eta\mathcal{L}_{js}$$

where $\gamma$ and $\eta$ are the weights used to balance the contributions of learning from the original data and augmented data.

## 4 Experiments

### 4.1 Datasets

We conducted experiments on both in-distribution datasets and out-of-distribution datasets to demonstrate the effectiveness of our proposed HiddenCut.

**In-Distribution Datasets**　We mainly trained and evaluated our methods on the widely-used GLUE benchmark (Wang et al., 2018) which covers a wide range of natural language understanding tasks: *single-sentence tasks* including: (i) Stanford Sentiment Treebank (SST-2) which predict the sentiment of movie reviews to be positive or negative, and (ii) Corpus of Linguistic Acceptability (CoLA) which predict whether a sentence is linguistically acceptable or not; *similarity and paraphrase tasks* including (i) Quora Question Pairs (QQP) which predict whether two question are paraphrases, (ii) Semantic Textual Similarity Benchmark (STS-B) which predict the similarity ratings between two sentences, and (iii) Microsoft Research Paraphrase Corpus

(MRPC) which predict whether two given sentences are semantically equivalent; *inference tasks* including (i) Multi-Genre Natural Language Inference (MNLI) which classified the relationships between two sentences into entailment, contradiction, or neutral, (ii) Question Natural Language Inference (QNLI) which predict whether a given sentence is the correct answer to a given question, and (iii) Recognizing Textual Entailment (RTE) which predict whether the entailment relation holds between two sentences. Accuracy was used as the evaluation metric for most of the datasets except that Matthews correlation was used for CoLA and Spearman correlation was utilized for STS-B.

**Out-Of-Distribution Datasets**　To demonstrate the generalization abilities of our proposed methods, we directly evaluated on 5 different out-of-distribution challenging sets, using the models that are fine-tuned on GLUE benchmark datasets:

- **Single Sentence Tasks**: Models fine-tuned from SST-2 are directly evaluated on two recent challenging sentiment classification datasets: IMDB Contrast Set (Gardner et al., 2020) including 588 examples and IMDB Counterfactually Augmented Dataset (Kaushik et al., 2020) including 733 examples. Both of them were constructed by asking NLP researchers (Gardner et al., 2020) or Amazon Mechanical Turkers (Kaushik et al., 2020) to make minor edits to examples in the original IMDB dataset (Maas et al., 2011) so that the sentiment labels change while the major contents keep the same.

- **Similarity and Paraphrase Tasks**: Models fine-tuned from QQP are directly evaluated on the recently introduced challenging paraphrase dataset PAWS-QQP (Zhang et al., 2019b) that has 669 test cases. PAWS-QQP contains sentence pairs with high word overlap but different semantic meanings created via word-swapping and back-translation from the original QQP dataset.

- **Inference Tasks**: Models fine-tuned from MNLI are directly evaluated on two challenging NLI sets: HANS (McCoy et al., 2019) with 30,000 test cases and Adversarial NLI (A1 dev sets) (Nie et al., 2020) including 1,000 test cases. The former one was constructed by using syntactic rules (lexical overlap, subsequence and constituent) to generate

| Method | MNLI | QNLI | QQP | RTE | SST-2 | MRPC | CoLA | STS-B | Avg |
|---|---|---|---|---|---|---|---|---|---|
| RoBERTa-base | 87.6 | 92.8 | 91.9 | 78.7 | 94.8 | 89.5 | 63.6 | 91.2 | 86.3 |
| ALUM | 88.1 | 93.1 | **92.0** | 80.2 | 95.3 | 90.9 | 63.6 | 91.1 | 86.8 |
| Token Cutoff | 88.2 | 93.1 | 91.9 | 81.2 | 95.1 | 91.1 | 64.1 | 91.2 | 87.0 |
| Feature Cutoff | 88.2 | 93.3 | **92.0** | 81.6 | 95.3 | 90.7 | 63.6 | 91.2 | 87.0 |
| Span Cutoff | **88.4** | 93.4 | **92.0** | 82.3 | 95.4 | 91.1 | 64.7 | 91.2 | 87.3 |
| HiddenCut † | 88.2 | **93.7** | **92.0** | **83.4** | **95.8** | **92.0** | **66.2** | **91.3** | **87.8** |

Table 1: In-distribution evaluation results on the dev sets of the GLUE benchmark. † means our proposed method.

| Method | Single-Sentence | | Similarity&Paraphrase | Inference | |
|---|---|---|---|---|---|
| | **IMDB-Cont.** | **IMDB-CAD** | **PAWS-QQP** | **HANS** | **AdvNLI (A1)** |
| RoBERTa-base | 84.6 | 88.4 | 38.4 | 67.8 | 31.2 |
| Span Cutoff | 85.5 | 89.2 | 38.8 | 68.4 | 31.1 |
| HiddenCut † | **87.8** | **90.4** | **41.5** | **71.2** | **32.8** |

Table 2: Out-of-distribution evaluation results on 5 different challenging sets. † means our proposed method. For all the datasets, we did not use their training sets to further fine-tune the derived models from GLUE.

non-entailment examples with high premise-hypothesis overlap from MNLI. The latter one was created by adversarial human-and-model-in-the-loop framework (Nie et al., 2020) to create hard examples based on BERT-Large models(Devlin et al., 2019) pre-trained on SNLI (Bowman et al., 2015) and MNLI.

### 4.2 Baselines

We compare our methods with several baselines:

- **RoBERTa** (Liu et al., 2019) is used as our base model. Note that RoBERTa is regularized with dropout during fine-tuning.

- **ALUM** (Liu et al., 2020) is the state-of-the-art adversarial training method for neural language models, which regularizes fine-tuning via perturbations in the embedding space.

- **Cutoff** (Shen et al., 2020) is a recent data augmentation for natural language understanding tasks by removing information in the input space, including three variations: token cutoff, feature cutoff, and span cutoff.

### 4.3 Implementation Details

We used the RoBERTa-base model (Liu et al., 2019) to initialize all the methods. Note that HiddenCut is agnostic to different types of pre-trained models. We followed Liu et al. (2019) to set the linear decay scheduler with a warmup ratio of 0.06 for training. The maximum learning rate was selected from $\{5e-6, 8e-6, 1e-5, 2e-5\}$ and the

max number of training epochs was set to be either 5 or 10. All these hyper-parameters are shared for all the models. The HiddenCut ratio $\alpha$ was set 0.1 after a grid search from $\{0.05, 0.1, 0.2, 0.3, 0.4\}$. The selecting ratio $\beta$ in the important sets sampling process was set 0.4 after a grid search from $\{0.1, 0.2, 0.4, 0.6\}$. The weights $\gamma$ and $\eta$ in our objective function were both 1. All the experiments were performed using a GeForce RTX 2080Ti.

### 4.4 Results on In-Distribution Datasets

Based on Table 1, we observed that, compared to *RoBERTa-base* with only dropout regularization, *ALUM* with perturbations in the embedding space through adversarial training has better results on most of these GLUE tasks. However, the extra additional backward passes to determine the perturbation directions in *ALUM* can bring in significantly more computational and memory overhead. By masking different types of input during training, *Cutoff* increased the performances while being more computationally efficient.

In contrast to *Span Cutoff*, *HiddenCut* not only introduced zero additional computation cost, but also demonstrated stronger performances on 7 out of 8 GLUE tasks, especially when the size of training set is small (e.g., an increase of 1.1 on RTE and 1.5 on CoLA). Moreover, *HiddenCut* achieved the best average result compared to previous state-of-the-art baselines. These in-distribution improvements indicated that, by strategically dropping contiguous spans in the hidden space, *HiddenCut* not

only helps pre-trained models utilize hidden information in a more effective way, but also injects less noise during the augmentation process compared to cutoff, e.g., *Span Cutoff* might bring in additional noises for CoLA (which aims to judge whether input sentences being linguistically acceptable or not) when one span in the input is removed, since it might change the labels.

### 4.5 Results on Out-Of-Distribution Datasets

To validate the better generalizability of Hidden-Cut, we tested our models trained on SST-2, QQP and MNLI directly on 5 out-of-distribution/out-of-domain challenging sets in zero-shot settings. As mentioned earlier, these out-of-distribution sets were either constructed with in-domain/out-of-domain data and further edited by human to make them harder, or generated by rules that exploited spurious correlations such as lexical overlap, which made them challenging to most existing models.

As shown in Table 2, *Span Cutoff* slightly improved the performances compared to *RoBERTa* by adding extra regularizations through creating restricted input. *HiddenCut* significantly outperformed both *RoBERTa* and *Span Cutoff*. For example, it outperformed *Span Cutoff.* by 2.3%(87.8% vs. 85.5%) on IMDB-Conts, 2.7%(41.5% vs. 38.8%) on PAWS-QQP, and 2.8%(71.2% vs 68.4%) on HANS consistently. These superior results demonstrated that, by dynamically and strategically dropping contiguous span of hidden representations, *HiddenCut* was able to better utilize all the important task-related information which improved the model generalization to out-of-distribution and challenging adversary examples.

### 4.6 Ablation Studies

This section presents our ablation studies on different sampling strategies and the effect of important hyper-parameters in *HiddenCut*.

#### 4.6.1 Sampling Strategies in HiddenCut

We compared different ways to cut hidden representations (**DropBlock** (Ghiasi et al., 2018) which randomly dropped spans in certain random hidden dimensions instead of the whole hidden space) and different sampling strategies for *HiddenCut* described in Section 3.2 (including **Random**, **LIME** (Ribeiro et al., 2016), **GEM** (Yang et al., 2019b), **Gradient** (Yeh et al., 2019), **Attention**) based on the performances on SST-2 and QNLI. For these strategies, we also experimented with a reverse set

| Strategy | SST-2 | QNLI |
|----------|-------|------|
| RoBERTa | 94.8 | 92.8 |
| DropBlock | 95.4 | 93.2 |
| Random | 95.4 | 93.5 |
| LIME | 95.2 | 93.1 |
| LIME-R | 95.3 | 93.2 |
| GEM | 95.5 | 93.4 |
| GEM-R | 95.1 | 93.2 |
| Gradient | 95.6 | 93.6 |
| Gradient-R | 95.1 | 93.4 |
| Attention | **95.8** | **93.7** |
| Attention-R | 94.6 | 93.4 |

Table 3: The performances on SST-2 and QNLI with different strategies when dropping information in the hidden space. Different sampling strategies combined with *HiddenCut* are presented. "-R" means sampling outside the set to be cut given by these strategies.

denoted by "-R" where we sampled outside the important set given by above strategies.

From Table 3, we observed that (i) sampling from important sets resulted in better performances than random sampling. Sampling outside the defined importance sets usually led to inferior performances. These highlights the importance of strategically selecting spans to drop. (ii) Sampling from dynamic sets sampled by their probabilities often outperformed sampling from predefined fixed sets (*LIME*), indicating the effectiveness of dynamically adjusting the sampling sets during training. (iii) The *attention-based* strategy outperformed all other sampling strategies, demonstrating the effectiveness of our proposed sampling strategies for *HiddenCut*. (iv) Completely dropping out the spans of hidden representations generated better results than only removing certain dimensions in the hidden space, which further validated the benefit of *HiddenCut* over *DropBlock* in natural language understanding tasks.

#### 4.6.2 The Effect of HiddenCut Ratios

The length of spans that are dropped by *Hidden-Cut* is an important hyper-parameter, which is controlled by the HiddenCut ratio $\alpha$ and the length of input sentences. $\alpha$ could also be interpreted as the extent of perturbations added to the hidden space. We presented the results of *Hidden-Cut* on MNLI with a set of different $\alpha$ including $\{0.05, 0.1, 0.2, 0.3, 0.4\}$ in Table 5. *HiddenCut* achieved the best performance with $\alpha = 0.1$, and

| Method | Original and Counterfactual Sentences | | | | | | | | | Prediction |
|---|---|---|---|---|---|---|---|---|---|---|
| RoBERTa | <s> | I | would | rate | 8 | stars | out | of | 10 | </s> | Positive |
| HiddenCut | <s> | I | would | rate | 8 | stars | out | of | 10 | </s> | Positive |
| RoBERTa | <s> | The | movie | became | more | and | more | intriguing | </s> | | Positive |
| HiddenCut | <s> | The | movie | became | more | and | more | intriguing | </s> | | Positive |
| RoBERTa | <s> | I | would | rate | 8 | stars | out | of | 20 | </s> | Positive |
| HiddenCut | <s> | I | would | rate | 8 | stars | out | of | 20 | </s> | Negative |
| RoBERTa | <s> | The | movie | became | only | slightly | more | intriguing | </s> | | Positive |
| HiddenCut | <s> | The | movie | became | only | slightly | more | intriguing | </s> | | Negative |

Table 4: Visualization of the attention weights at the last layer in models. The sentences in the first section are from IMDB with positive labels and the sentences in the second section is constructed by changing ratings or diminishing via qualifiers (Kaushik et al., 2020) to flip their corresponding labels. Deeper blue represents that those tokens receive higher attention weights.

| $\alpha$ | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| **MNLI** | 88.07 | **88.23** | 88.13 | 88.07 | 87.64 |

Table 5: Performances on MNLI with different Hidden-Cut ratio $\alpha$, which controls the length of span to cut in the hidden space.

| $\beta$ | 0.1 | 0.2 | 0.4 | 0.6 |
|---|---|---|---|---|
| **SST-2** | 95.18 | 95.30 | **95.76** | 95.46 |

Table 6: Performances on SST-2 with different sampling ratio $\beta$, which controls the size of important token set from which HiddenCut would sample.

the performance gradually decreased with higher $\alpha$ since larger noise might be introduced when dropping more hidden information. This suggested the importance of balancing the trade-off between applying proper perturbations to regularize models and injecting potential noises.

### 4.6.3 The Effect of Sampling Ratios

The number of words that are considered important and selected by *HiddenCut* is also an influential hyper-parameter controlled by the sampling ratio $\beta$ and the length of input sentences. As shown in Table 6, we compared the performances on SST-2 by adopting different $\beta$ including $\{0.1, 0.2, 0.4, 0.6\}$. When $\beta$ is too small, the number of words in the important sets is limited, which might lead *HiddenCut* to consistently drop certain hidden spans during the entire training process. The low diversities reduce the improvements over baselines. When $\beta$ is too large, the important sets might cover all the words except stop words in sentences. As a result, the *Attention-based Strategy* actually became *Random Sampling*, which led to lower gains over baselines. The best performance was achieved when $\beta = 0.4$, indicating a reasonable trade-off between diversities and efficiencies.

### 4.7 Visualization of Attentions

To further demonstrate the effectiveness of Hid-denCut, we visualize the attention weights that the special start token ("<s>") assigns to other tokens at the last layer, via several examples and their coun-

terfactual examples in Table 4. We observed that *RoBERTa* only assigned higher attention weights on certain tokens such as "8 stars", "intriguing" and especially the end special token "</s>", while largely ignored other context tokens that were also important to make the correct predictions such as scale descriptions (e.g., "out of 10") and qualifier words (e.g., "more and more"). This was probably because words like "8 stars" and "intriguing" were highly correlated with positive label and *RoBERTa* might overfit such patterns without probable regularization. As a result, when the scale of ratings (e.g., from "10" to "20") or the qualifier words changed (e.g., from "more and more" to "only slightly more"), *RoBERTa* still predicted the label as positive even when the groundtruth is negative. With *HiddenCut*, models mitigated the impact of tokens with higher attention weights and were encouraged to utilize all the related information. So the attention weights in *HiddenCut* were more uniformly distributed, which helped models make the correct predictions for out-of-distribution counterfactual examples. Taken together, *HiddenCut* helps improve model's generalizability by facilitating it to learn from more task-related information.

## 5 Conclusion

In this work, we introduced a simple yet effective data augmentation technique, HiddenCut, to improve model robustness on a wide range of natural language understanding tasks by drop-

ping contiguous spans of hidden representations in the hidden space directed by strategic attention-based sampling strategies. Through HiddenCut, transformer models are encouraged to make use of all the task-related information during training rather than only relying on certain spurious clues. Through extensive experiments on in-distribution datasets (GLUE benchmarks) and out-of-distribution datasets (challenging counterexamples), HiddenCut consistently and significantly outperformed state-of-the-art baselines, and demonstrated superior generalization performances.

## Acknowledgment

## References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(61):1803–1831.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv:2002.12804*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020a. Local additivity based data augmentation for semi-supervised ner. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1241–1251.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. Mix-Text: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Liyan Chen, P. Gautier, and Sergül Aydöre. 2020c. Dropcluster: A structured dropout for convolutional networks. *ArXiv*, abs/2002.02997.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *EMNLP*.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

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

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

G. Ghiasi, Tsung-Yi Lin, and Quoc V. Le. 2018. Dropblock: A regularization method for convolutional networks. In *NeurIPS*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Counterfactual visual explanations. In *ICML*, pages 2376–2384.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2020. Parameter-efficient transfer learning with diff pruning.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pretrained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Gustav Larsson, M. Maire, and Gregory Shakhnarovich. 2017. Fractalnet: Ultra-deep neural networks without residuals. *ArXiv*, abs/1605.07648.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *SCL*.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

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*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. *arXiv: Machine Learning*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Hieu Pham and Quoc V. Le. 2021. Autodropout: Learning dropout patterns to regularize deep networks.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference*

on *Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3358–3369.

Dinghan Shen, M. Zheng, Y. Shen, Yanru Qu, and W. Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *ArXiv*, abs/2009.13818.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019a. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Ziyi Yang, Chenguang Zhu, and Weizhu Chen. 2019b. Parameter-free sentence embedding via orthogonal basis. 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 638–648, Hong Kong, China. Association for Computational Linguistics.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. 2019. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019a. You only propagate once: Painless adversarial training using maximal principle. *arXiv preprint arXiv:1905.00877*, 2(3).

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.