RESIDUAL PERTURBATIONS AS SENTENCE-PAIR GENERATORS FOR CONTRASTIVE SENTENCE EMBEDDING TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sentence embeddings, which map sentences to dense vectors that capture their meaning, underpin many NLP tasks such as semantic search, clustering, and retrieval. Training sentence embeddings typically relies on sentence pairs, which are costly to generate. To solve this resource bottleneck, researchers have developed alternative methods to generate sentence pairs efficiently. Unsupervised alternatives bypass this by forming positive pairs from single sentences. SimCSE is a well-established unsupervised method that uses dropout noise to create two views of the same sentence as a positive pair.

In this article, we introduce and explore residual perturbations as an alternative sentence pair generation method. Residual perturbations generate sentence pairs by adding perturbations onto the original embedding between the embedding layer and subsequent attention layers, treating the resulting perturbed embedding as the positive counterpart to the original. We investigate different embedding-level perturbations: random noise, rolling, shuffling, and adding the sentence's centroid.

The results highlight that embedding-level residual perturbations performs on par with dropout-based methods, such as SimCSE. On the MTEB benchmark, residual rolling and residual shuffling perturbations on average match or slightly exceed SimCSE after unsupervised contrastive pretraining, with residual shuffling winning the largest share of individual tasks. Adding the centroid recovers much of this benefit through a simple constant shift, while random noise and, more severely, discarding the original embedding underperform even an untouched baseline.

While our method is comparable to SimCSE overall, viewing performance per task category reveals clear differences in performance with narrow losses on retrieval and clustering tasks and strong improvements on STS and pair classification tasks. The strong performance of the residual shuffle perturbation on STS tasks suggests that perturbations can be chosen according to the downstream task at hand, offering an additional layer of control over sentence embedding generation, as well as a working alternative to dropout-based methods of sentence pair generation.

1 INTRODUCTION

Representing text as semantically meaningful vectors has emerged as a core step for many tasks in natural language processing (Mikolov et al., 2013; Pennington et al., 2014). Downstream tasks such as classification, retrieval, clustering, and semantic textual similarity (STS) all build on such representations, which sentence embedding models aim to provide (Kiros et al., 2015; Conneau et al., 2017; Reimers & Gurevych, 2019). Further, the growing adoption of retrieval-augmented generation (RAG) systems in industry applications has only heightened the importance of sentence embedding models (Lewis et al., 2020; Gao et al., 2023).

State-of-the-art sentence embedding models are generally developed with the same recipe of pretraining and finetuning (Howard & Ruder, 2018; Devlin et al., 2019). In practice, they build on large pretrained encoders (Devlin et al., 2019; Liu et al., 2019), whose token-level outputs are pooled into

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a single fixed-size sentence vector (Reimers & Gurevych, 2019). This pooled representation is then optimized by contrastive finetuning on labeled sentence pairs or triplets to produce semantically meaningful representations (Conneau et al., 2017; Reimers & Gurevych, 2019).

Concretely, this contrastive finetuning step trains the encoder on batches of sentence pairs that are labeled as semantically related, commonly referred to as positives (Conneau et al., 2017; Reimers & Gurevych, 2019). The model is trained in a way that for every sentence in a batch, its paired sentence is pulled closer in embedding space, typically by maximizing their cosine similarity, while all other sentences in the batch act as negatives and are pushed away (Chopra et al., 2005; Hadsell et al., 2006; van den Oord et al., 2018). Repeating this over a large number of such labeled pairs or triplets yields an embedding space in which distance reflects semantic similarity (Cer et al., 2017; Reimers & Gurevych, 2019).

The main limitation of this recipe is its dependence on labeled sentence pairs: datasets that reliably mark sentence pairs as semantically related, are costly to construct, cover only a handful of languages and domains, and limit how far supervised contrastive training can scale (Bowman et al., 2015; Williams et al., 2018).

To overcome this bottleneck, a large body of work has turned to an idea from computer vision: rather than relying on labels, a positive pair can be constructed from a single unlabeled example through data augmentation, by generating two augmented views of the example and treating them as equivalent, while views of other examples serve as in-batch negatives (Chen et al., 2020). Following this principle, several methods have been proposed to automatically generate sentence pairs without human annotation, typically by perturbing an input sentence to obtain a second view of it (Wu et al., 2020; Giorgi et al., 2021; Yan et al., 2021; Wu et al., 2022b). The most widely adopted of these is SimCSE (Gao et al., 2021), which forms a positive pair by encoding the same sentence twice and relying solely on dropout to perturb the resulting embeddings, with every other sentence in the batch treated as a negative.

In this work, we introduce a new method to generate positive sentence pairs. We perturb the input embeddings directly instead of relying on the encoder's dropout as in SimCSE. In a manner reminiscent of a residual connection, immediately after the embedding layer and before any attention computation, we add a perturbed copy of a sentence's input embedding onto itself, and treat the resulting perturbed embedding as the positive counterpart to the unperturbed one. We study multiple residual perturbations, namely random noise, residual shuffling, residual rolling (cyclic shifting) or adding the sentence's centroid, as well as an ablation that replaces the embedding with its rolled copy instead of adding to it. We evaluate all variants against SimCSE on MTEB, both overall and per task category.

2 Related Work

Contrastive sentence embedding methods commonly build on some variant of the InfoNCE objective (van den Oord et al., 2018), which casts representation learning as classifying, for a given anchor, its true positive among a set of negatives via a softmax over similarity scores. Chen et al. (2020) popularized a temperature-scaled version of this loss, commonly known as NT-Xent, for learning visual representations from augmented image pairs, and a closely related formulation underlies SimCSE (Gao et al., 2021) as well as much of the work discussed below.

Since the introduction of InfoNCE, a substantial body of work modifies or extends it directly, independently of how the positive pairs themselves are constructed. Khosla et al. (2020) generalize it to the supervised setting, treating every example that shares a label with the anchor as a positive rather than relying on a single augmented view. Robinson et al. (2021) observe that InfoNCE weights all negatives in a batch equally regardless of how informative they are, and derive an unsupervised, hardness-weighted negative distribution that emphasizes negatives close to the anchor. Chuang et al. (2020) note that negatives sampled uniformly at random are frequently false negatives, i.e., examples semantically similar to the anchor that the loss nonetheless pushes apart, and introduce a debiasing correction. Specific to sentence embeddings, Liu et al. (2023) augment the objective with an auxiliary ranking-consistency and distillation term computed across multiple dropout views of the same batch; and Li & Li (2024) argue that optimizing cosine similarity directly saturates in regions where its gradient vanishes, and reformulate the objective as an angle optimization in complex space. The

methods discussed in the remainder of this section largely keep one of these InfoNCE variants fixed and differ chiefly in how the positive sentence pairs fed into it are constructed. Despite this substantial body of work on InfoNCE improvements, for the purposes of this paper we employ van den Oord et al. (2018) initial version.

SimCSE (Gao et al., 2021) is a very well known and established, unsupervised instance of sentence pair generation. It forms a positive pair by encoding the same sentence twice, relying only on the randomness of dropout inside the encoder to produce two distinct embeddings. Around the same time, ConSERT (Yan et al., 2021) pursued the same goal through computer-vision-style augmentations applied at the token embedding layer, comparing strategies such as adversarial perturbation, token shuffling, cutoff, and dropout. ConSERT finds token-level augmentations, in particular token shuffling, to be most effective and obtains its best results by combining shuffling with feature cutoff. Its token shuffling permutes the position indices passed to the embedding layer and thus replaces the original token order in the second view, whereas our residual shuffling adds a shuffled copy onto the unperturbed embedding and retains the original order in both views.

A number of works build directly on SimCSE by diagnosing and addressing weaknesses of its dropout-based positives. ESimCSE (Wu et al., 2022b) observes that a dropout-only positive necessarily has the same length as its anchor, biasing the model toward treating similar sentence length as a proxy for semantic similarity; it corrects this by randomly repeating words in the input before encoding, and separately enlarges the negative pool with a momentum encoder. DiffCSE (Chuang et al., 2022) keeps SimCSE's dropout-based positive pair but adds a conditional discriminator trained to detect whether tokens in the sentence were replaced by a masked-language-model-based edit, making the resulting embedding sensitive to such edits rather than fully invariant to them, an instance of what its authors call equivariant contrastive learning. InfoCSE (Wu et al., 2022a) instead keeps the dropout-based positive pair unchanged but adds an auxiliary masked-language-modeling head that forces the pooled sentence representation to aggregate more of the input's information, arguing that dropout alone gives the encoder too little incentive to encode a semantically complete sentence vector.

Most closely related to our work, González-Márquez et al. (2026) directly revisit the choice between dropout and input-level augmentation that SimCSE and ConSERT left open. Evaluating on MTEB and additional in-domain benchmarks, they find that cropping, a purely input-level augmentation, strongly outperforms SimCSE's dropout-based positives, calling into question whether dropout is a sufficient source of augmentation for text embeddings. This result motivates a broader search over what an effective positive pair should look like once one moves past dropout, which is precisely the space of embedding-level perturbations that we explore in this work.

3 METHOD

3.1 EMBEDDING-LEVEL PERTURBATIONS FOR SENTENCE-PAIR GENERATION

The general architecture of our model is based on a standard transformer-encoder structure. A standard transformer encoder such as BERT (Devlin et al., 2019) first maps an input sentence of L tokens to a sequence of embeddings $\mathbf{x} \in \mathbb{R}^{L \times d}$ via a combined token, segment, and positional embedding layer, and only then passes $\mathbf{x}$ through a stack of self-attention blocks (Vaswani et al., 2017). Every perturbation studied in this work is applied at this point in the model: immediately after the embedding layer has produced $\mathbf{x}$, and before $\mathbf{x}$ is sent to the first attention block. For a given unlabeled sentence, this yields two views, the unperturbed embedding $\mathbf{x}$ and a perturbed embedding $\mathbf{x}'$, that are subsequently passed through the same, otherwise unmodified encoder. The resulting pooled sentence representations are subsequently processed by a SwiGLU layer and form the positive pair used for contrastive pretraining. The representations of every other sentence in the batch (and their perturbed views) serve as negatives.

Because the perturbation is injected before any attention layer, the encoder architecture and its weights are never modified, in contrast to SimCSE (Gao et al., 2021), which perturbs the sentence representation implicitly by sampling a different dropout mask inside the encoder on each forward pass. Here, the perturbation is confined entirely to embedding space and is applied in a manner reminiscent of a residual connection (He et al., 2016; Srivastava et al., 2015), an additive term computed directly from $\mathbf{x}$ and added back onto itself, as detailed for each perturbation in Section 3.2. Since

attention mixes information across all token positions (Vaswani et al., 2017), a perturbation that only directly touches part of the sequence can still propagate to, and influence, the pooled representation of the entire sentence once $\mathbf{x}'$ is processed by the encoder. Figure 1 illustrates this position in the model.

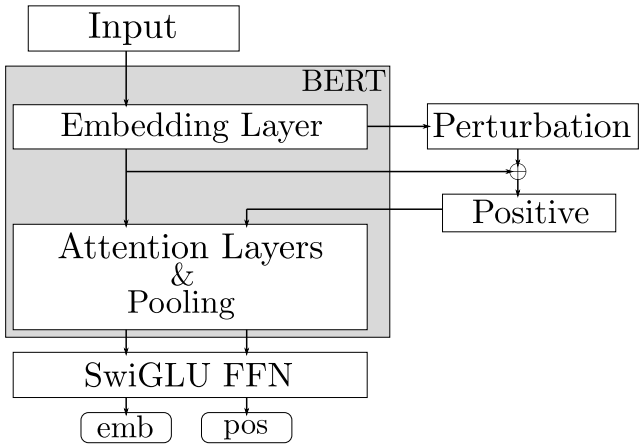

Figure 1: Residual perturbations for sentence-pair generation. The output of the BERT embedding layer is passed to a perturbation function, and the perturbed embedding is added onto the original embedding, similar to a residual connection. Both views of the resulting positive pair are processed by the BERT attention layers, mean pooled, and passed through a final SwiGLU feed-forward layer.

3.2 PERTURBATIONS

We study five perturbations of $\mathbf{x}$, all applied at the position described in Section 3.1. Each defines the perturbed embedding $\mathbf{x}'$ used as the positive counterpart to $\mathbf{x}$. Table 1 summarizes them, Figure 2 illustrates them, and the remainder of this section discusses each in turn.

Table 1: Overview of the embedding-level perturbations studied in this work. p_{noise}, p_{rolled}, and p_{shuffle} are defined in the corresponding paragraphs below.

Perturbation	Definition	Role
Noise	$\mathbf{x}' = \mathbf{x} + p_{\text{noise}}$	unstructured noise baseline
Residual rolling	$\mathbf{x}' = \mathbf{x} + p_{\text{rolled}}$	preserves relative word order
Rolling	$\mathbf{x}' = p_{\text{rolled}}$	ablation of residual rolling
Residual shuffling	$\mathbf{x}' = \mathbf{x} + p_{\text{shuffle}}$	stronger word order disruption
Residual mean	$\mathbf{x}' = \mathbf{x} + \text{mean}(\mathbf{x})$	constant-shift control

Noise: p_{noise} is sampled i.i.d. from a standard normal distribution $\mathcal{N}(0, 1)$, independently for every token position and every forward pass. As the perturbation is unstructured and independent of the sentence's content, it serves as a baseline for how much of any observed effect is due to perturbing the embedding at all, as opposed to perturbing it with structure drawn from the sentence itself.

$$\mathbf{x}' = \mathbf{x} + p_{\text{noise}}$$

Residual rolling: $\text{roll}(\mathbf{x}, k)$ cyclically shifts $\mathbf{x}$ along the token dimension by a random offset k, drawn from the range of the sentence length, i.e., $1 \leq k < L$, so that p_{rolled} reuses the sentence's own token embeddings, reassigned to different positions, rather than drawing from an independent noise source.

$$\mathbf{x}' = \mathbf{x} + p_{\text{rolled}}, \qquad p_{\text{rolled}} = \text{roll}(\mathbf{x}, k)$$

Rolling: Rather than adding the rolled embedding back onto the original, this variant discards $\mathbf{x}$ entirely and uses the rolled embedding by itself as the perturbed view. Comparing it with residual

rolling isolates the contribution of the residual addition of $\mathbf{x}$.

$$\mathbf{x}' = p_{\text{rolled}}$$

Residual shuffling: $\text{shuffle}(\mathbf{x})$ applies a random permutation to the token positions of $\mathbf{x}$ similar to ConSERT. Unlike the cyclic shift of rolling, it does not preserve the relative order of the tokens and therefore disrupts positional structure more strongly, while, like rolling, still drawing the perturbation from the sentence's own embeddings.

$$\mathbf{x}' = \mathbf{x} + p_{\text{shuffle}}, \qquad p_{\text{shuffle}} = \text{shuffle}(\mathbf{x})$$

Residual mean: $\text{mean}(\mathbf{x})$ is the sentence's centroid, i.e., its token embeddings averaged over the sequence dimension and broadcast to every position. Because this perturbation shifts all positions by the same vector and carries no token-level structure, it serves as a control for rolling and shuffling: if they performed no better than adding the centroid, their effect could not be attributed to the rearrangement of the sentence's own embeddings, but merely to an additive offset.

$$\mathbf{x}' = \mathbf{x} + \text{mean}(\mathbf{x})$$

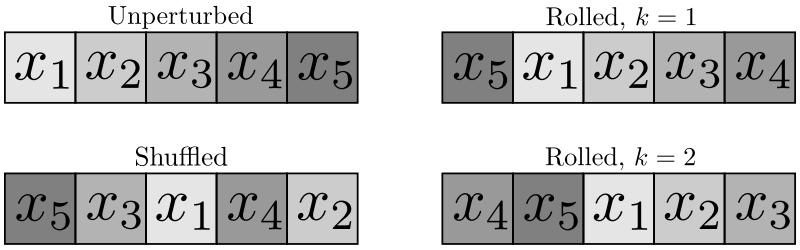

Figure 2: Illustration of the studied embedding-level perturbations.

4 Experimental Setup

All models are initialized from pretrained BERT (Devlin et al., 2019) and use its pretrained tokenizer; the same pretrained model without any contrastive training serves as the BERT baseline. Contrastive training via InfoNCE is performed on 74 million unlabeled samples from the BookCorpus dataset (Zhu et al., 2015), with positive pairs generated by the respective perturbation (Section 3.2) or, for SimCSE, by dropout (Gao et al., 2021). To preserve prevent catastrophic forgetting, we employ gradual unfreezing (Howard & Ruder, 2018): All encoder layers are frozen at the start of training, the last BERT layer is unfrozen after 40% of the training data has been processed, and one additional layer is unfrozen after every further 5% of data, until the entire encoder is unfrozen. We use a learning rate of 10^{-5}, a BERT dropout rate of 0.1 and a maximum input length of 512 tokens. Each model is trained for a single epoch and subsequently evaluated on the English MTEB benchmark (MTEB(eng, v2)) (Muennighoff et al., 2023).

5 Results

We evaluate all variants on English MTEBv2 (Muennighoff et al., 2023) after unsupervised contrastive training.

MTEB results. Table 2 reports category-averaged scores. Residual shuffling attains the highest average (46.70), followed by residual rolling (46.11), SimCSE (45.75) and residual mean (45.24). SimCSE leads in four of seven categories, but by at most 1.14 points; the higher averages of residual shuffling and rolling stem from large gains on STS compared to SimCSE (67.76 and 66.71 vs. 62.57) and pair classification (71.52 and 69.16 vs. 64.33). Random noise (40.53) and only rolling (34.03) fall below the BERT baseline (42.02). This indicates that the non-residual perturbation remove too much information on their own and the residual connection is integral to performance. We further see that no residual perturbation clearly outperforms SimCSE or each other, with residual rolling and shuffling performing slightly better than residual mean.

Table 2: Category-averaged scores on MTEB(eng, v2) with the main metric of each category in parentheses; all scores $\times 100$. Avg.: mean over the seven categories. Best per row in bold.

	BERT	SimCSE	rolled	noise	res. rolled	res. shuffle	res. mean
Classif. (Acc.)	66.21	**68.53**	60.52	64.17	67.97	67.53	67.73
Clustering (V-meas.)	**39.54**	39.34	34.76	33.88	37.90	37.40	37.40
Pair Classif. (AP)	56.33	64.33	36.81	57.47	69.16	**71.52**	63.50
Reranking (MAP)	37.20	**38.73**	33.48	36.66	38.56	38.30	38.09
Retrieval (ndcg@10)	8.70	**13.30**	4.00	5.73	11.43	13.18	12.64
STS (cos. Spear.)	53.38	62.57	36.26	59.89	66.71	**67.76**	65.04
Summarization (cos. Spear.)	32.76	**33.44**	32.37	25.93	31.05	31.20	32.30
Avg.	42.02	45.75	34.03	40.53	46.11	**46.70**	45.24

Per-task MTEB results. Table 3 confirms this pattern per task. Residual shuffling wins 19 of 41 tasks, SimCSE 12 and residual rolling 5, while random noise and only rolling win none. Residual shuffling outperforms SimCSE on all nine STS and three pair classification tasks, residual rolling on all but STS13. SimCSE leads on most classification and clustering tasks, on summarization, and on ClimateFEVER (HN), FEVER (HN), FiQA2018 and Touché2020, whereas residual shuffling wins the remaining six retrieval tasks. This shows a task-dependent division in performance for the different perturbations and SimCSE.

We attribute this division to the nature of STS and pair classification: similarity in these tasks is largely determined by shared content words, and order-insensitive representations such as averaged word embeddings or bag-of-words models typically perform well on them (Pennington et al., 2014; Wieting et al., 2016; Arora et al., 2017; Reimers & Gurevych, 2019). The rolling and shuffling perturbations de-emphasize word order, so the contrastive objective rewards a bag-of-words-like invariance to word order. Such representations are good at capturing which words a sentence contains, but not their order or syntactic structure (Conneau et al., 2018). Consistently, shuffling, the stronger disruption of word order, yields the largest gains. This invariance offers little benefit where lexically similar texts must be distinguished, as in hard-negative retrieval or order-sensitive classification. Neither method is therefore generally superior; each excels on different categories.

Statistical significance over all tasks. To test whether these differences are systematic, we apply a two-sided Wilcoxon signed-rank test to the paired per-task scores and compute 95% percentile bootstrap confidence intervals for the mean paired difference Δ. Over all 41 tasks (Table 4), the BERT baseline, random noise and only rolling are significantly worse than SimCSE ($p < 0.001$), whereas no residual variant differs significantly (residual shuffling $\Delta = +0.99$, $p = 0.456$; residual rolling $+0.35$, $p = 0.837$; residual mean -0.27, $p = 0.354$). All three bootstrap intervals contain zero and bound Δ to $[-1.04, +2.38]$ points, indicating that the residual variants perform on par with SimCSE in aggregate.

Per category (Table 5), all three residual variants significantly outperform SimCSE on STS (residual shuffling $\Delta = +5.20$, $p = 0.004$; residual rolling $+4.15$ and residual mean $+2.48$, both $p = 0.020$). On pair classification, residual shuffling and rolling improve on all three tasks ($+7.18$ and $+4.83$), but with $n = 3$ no sensible p-value can be calculated. Significant deficits occur only for residual rolling on retrieval (-1.87, $p = 0.027$) and residual mean on clustering (-1.94, $p = 0.023$); reranking and summarization contain too few tasks for testing. These p-values show that there are clear and significant differences per task category between the methods. This indicates a potential additional lever to optimize sentence embeddings with regards to a desired downstream task.

Potential Shortcutting. Since the perturbation is added onto $\mathbf{x}$, the encoder could learn to ignore the perturbations, in which case any gains would stem solely from the in-batch negatives. We test this with an *identical* baseline, in which $\mathbf{x}' = \mathbf{x}$ and dropout is disabled, so that both views coincide and only in-batch negatives provide a training signal. Due to computational constraints, we train identical, residual rolling, residual shuffling and SimCSE on approximately one million samples and evaluate them on the 28 classification, clustering, pair classification and STS tasks of MTEBv2(Table 6).

Table 3: MTEB main scores per task for each variant. Best per row in bold.

Task	BERT	SimCSE	rolled	noise	res. rolled	res. shuffle	res. mean
ArguAna	28.44	31.97	32.23	21.72	32.68	**34.64**	34.62
ArXivHier. P2P	**56.95**	55.29	55.60	51.19	55.56	53.47	56.04
ArXivHier. S2S	**55.36**	53.61	49.63	50.99	52.04	50.85	52.80
AskUbuntuDup	45.73	48.63	40.77	46.36	**49.30**	49.14	48.26
BIOSSES	54.70	56.93	39.70	64.76	**66.77**	66.67	61.64
Banking77	63.46	69.20	55.86	61.51	69.29	**70.28**	68.51
BiorxivClu. P2P	33.00	33.15	33.66	28.34	**33.98**	33.81	31.94
CQADup. Gaming	9.42	18.50	0.14	6.68	16.76	**18.82**	17.47
CQADup. Unix	4.36	8.22	0.26	4.31	9.50	**10.39**	9.10
ClimateFEVER (HN)	5.50	**8.97**	0.72	3.27	5.64	6.43	8.74
FEVER (HN)	4.57	**7.00**	0.04	1.85	3.48	4.96	5.37
FiQA2018	2.10	**4.92**	0.02	0.74	2.55	3.24	3.71
HotpotQA (HN)	11.43	17.46	1.35	6.25	16.56	**18.61**	15.55
IMDb	65.35	**66.79**	64.90	59.95	65.15	62.92	66.32
MTOPDomain	82.64	86.69	73.93	82.68	86.55	**86.99**	84.48
MassiveIntent	59.91	63.74	53.12	61.12	**64.43**	63.88	62.58
MassiveScenario	64.28	**68.21**	54.35	65.40	68.16	68.06	66.21
MedrxivClu. P2P	31.20	31.32	31.40	27.72	31.29	**31.75**	30.13
MedrxivClu. S2S	29.01	**29.65**	25.57	24.30	28.94	28.43	28.73
MindSmall	28.66	**28.84**	26.19	26.97	27.82	27.46	27.91
SCIDOCS	2.51	4.39	0.36	1.38	4.17	**4.98**	3.85
SICK-R	58.65	65.50	38.98	63.21	67.09	**67.37**	65.04
STS12	30.87	47.56	29.79	46.35	54.70	**56.85**	52.60
STS13	59.89	69.73	24.02	54.95	68.10	**69.91**	69.21
STS14	47.73	58.91	16.36	51.09	61.77	**63.34**	61.85
STS15	60.29	69.93	36.21	67.21	75.55	**76.68**	73.42
STSBenchmark	47.29	62.40	33.80	57.38	67.58	**68.62**	65.75
SprintDup.	36.81	53.18	11.61	36.30	63.65	**70.90**	46.75
StackExch. Clu.	**51.35**	51.20	39.93	41.39	47.01	46.27	46.25
StackExch. Clu. P2P	33.62	**33.94**	33.34	32.30	33.02	33.35	33.20
TRECCOVID	14.01	19.90	4.85	7.21	18.13	**21.79**	18.69
Touché2020	4.70	**11.69**	0.00	3.85	4.87	7.90	9.35
ToxicConv.	**67.98**	66.84	60.90	60.68	63.66	63.24	65.23
TweetSentiment	51.80	**52.19**	49.93	49.96	51.65	50.97	51.57
20Newsgroups	25.84	**26.57**	8.95	14.79	21.35	21.29	20.15
TwitterSemEval15	55.90	60.54	40.93	59.80	**63.94**	63.47	63.62
TwitterURL	76.29	79.28	57.89	76.31	79.89	**80.18**	80.13
SummEval	32.76	**33.44**	32.37	25.93	31.05	31.20	32.30
AmazonCounterfact.	74.24	74.55	71.15	72.06	74.85	73.87	**76.90**
STS17	64.10	73.50	51.82	75.54	79.36	**80.55**	76.16
STS22	56.93	58.63	55.66	58.56	59.49	**59.87**	59.72
Tasks won	4	12	0	0	5	19	1

Residual rolling (60.36), SimCSE (59.15) and residual shuffling (59.05) again perform comparably, with the residual variants leading on pair classification and SimCSE on clustering; the swapped order of rolling and shuffling is plausibly due to the reduced training data or initialization noise. The identical baseline trails the residual variants by 4 to 18 points on classification, pair classification and STS (average 54.27) and is competitive only on clustering. It is therefore likely that the gains of the residual variants arise from the perturbations rather than from in-batch negatives alone.

In summary, residual shuffling and residual rolling match SimCSE in overall MTEB performance, providing an alternative way to generate positive pairs from unlabeled data without modifying the encoder. This parity conceals a clear division between categories: the residual perturbations significantly improve STS and consistently improve pair classification, while SimCSE retains an advantage on classification, clustering and parts of retrieval.

Table 4: Paired comparison against SimCSE over all 41 MTEB tasks of Table 3. Mean/Median Δ: mean and median paired difference to SimCSE; CI: 95% percentile bootstrap interval of the mean Δ (10,000 task resamples); p: two-sided Wilcoxon signed-rank test. $p < 0.05$ in bold.

Method	Mean Δ	Median Δ	95% Bootstrap CI	Wilcoxon p
res. shuffle	$+0.99$	$+0.32$	$[-0.21, +2.38]$	0.456
res. rolled	$+0.35$	-0.05	$[-0.67, +1.47]$	0.837
res. mean	-0.27	-0.69	$[-1.04, +0.48]$	0.354
BERT baseline	-4.23	-2.99	$[-5.72, -2.87]$	$<$**0.001**
noise	-5.14	-4.10	$[-6.55, -3.72]$	$<$**0.001**
rolled	-12.80	-10.62	$[-16.56, -9.30]$	$<$**0.001**

Table 5: Paired comparison of the residual variants against SimCSE per MTEB task category, computed over the tasks of Table 3; n: number of tasks in the category. Δ: mean paired difference to SimCSE; p: two-sided Wilcoxon signed-rank test ($p < 0.05$ in bold; undefined for $n = 1$); W/L: number of tasks on which the variant scores above/below SimCSE.

		res. rolled			res. shuffle			res. mean		
Category	n	Δ	p	W/L	Δ	p	W/L	Δ	p	W/L
Retrieval	10	-1.87	**0.027**	2/8	-0.13	1.000	6/4	-0.66	0.160	2/8
Clustering	8	-1.44	0.109	2/6	-1.94	0.055	2/6	-1.94	**0.023**	1/7
Classification	8	-0.56	0.461	3/5	-1.00	0.250	3/5	-0.80	0.195	1/7
Pair Classif.	3	$+4.83$	–	3/0	$+7.18$	–	3/0	-0.83	–	2/1
Reranking	2	-0.18	–	1/1	-0.44	–	1/1	-0.65	–	0/2
STS	9	$+4.15$	**0.020**	8/1	$+5.20$	**0.004**	9/0	$+2.48$	**0.020**	7/2
Summarization	1	-2.39	–	0/1	-2.24	–	0/1	-1.14	–	0/1

Table 6: MTEB main scores after contrastive pretraining on a reduced dataset of approximately one million samples. Number of tasks per category in parentheses; the average is taken over all 28 tasks. *Identical*: the unperturbed embedding serves as its own positive, with dropout disabled. Best per row in bold.

Category	identical	res. rolled	res. shuffle	SimCSE
Classification (8)	61.72	**66.96**	65.75	65.54
Clustering (8)	40.44	38.78	38.19	**40.70**
Pair Classification (3)	54.54	**72.62**	71.60	64.06
STS (9)	59.84	**69.57**	67.46	68.23
Average (28)	54.27	**60.36**	59.05	59.15

As for image augmentations (Chen et al., 2020), the perturbation determines the invariances the encoder acquires and can thus be chosen according to the target task: order-disrupting perturbations for paraphrase or duplicate detection, order-preserving ones for tasks that must distinguish lexically similar texts. Whether perturbations can be designed for targeted gains in other categories, such as retrieval, remains open and an exciting future research direction.

6 Conclusion

This work explored residual perturbations applied directly in embedding space, immediately after the embedding layer and before any attention computation, as an alternative to SimCSE's dropout-based positive pairs (Gao et al., 2021). We compared four residual perturbations, random noise, residual rolling, residual shuffling and adding the sentence's centroid, together with only rolling, an ablation that replaces the original embedding instead of adding to it.

After unsupervised contrastive training, residual rolling and residual shuffling perform on par with SimCSE on MTEB: Their average scores are slightly higher, and the differences over all tasks are

not statistically significant. Nevertheless, this parity conceals a clear division between task categories. Both perturbations significantly improve STS and consistently improve pair classification, which we attribute to an induced invariance to word order, while SimCSE retains an advantage on classification, clustering and parts of retrieval. Random noise and, more severely, only rolling fall below the BERT baseline, underlining the importance of retaining the original embedding in the positive view. A baseline trained with identical positives and in-batch negatives alone performs clearly worse, showing that the encoder does not learn to ignore the perturbation.

Taken together, simple, architecture-agnostic perturbations applied in embedding space can rival dropout as a source of automatically generated sentence pairs. Moreover, the choice of perturbation shifts performance towards specific task categories, which opens the possibility of selecting perturbations to optimize for different downstream task.

Ethics Statement

All authors have read and adhere to the ICLR Code of Ethics. This work does not involve human subjects, personally identifiable information or the release of new datasets. We also do not foresee any negative societal impacts. Contrastive pretraining is performed on BookCorpus, which we chose because it is part of the original BERT pretraining corpus and therefore does not introduce data outside the domain the encoder was trained on. We acknowledge that BookCorpus has been criticized for concerns regarding copyright, licensing and the consent of the authors whose books it contains. Due to time constraints, we are unable to repeat our experiments on a different corpus before submission.

Reproducibility Statement

We have taken several steps to ensure the reproducibility of our results. Section 3.1 specifies where the perturbations are applied in the model, and Section 3.2 formally defines each perturbation together with its sampling procedure. Section 4 describes the base model, training data, unfreezing schedule and training hyperparameters. All evaluations use the publicly available MTEB(eng, v2) benchmark, and the per-task scores underlying our analysis are reported in full in Table 3, so that the statistical tests in Section 5 can be reproduced directly from the paper. Our code will be made publicly available during revision after the double blind process.

AI Use Statement

We used large language models (LLMs) as general-purpose assistants during the preparation of this paper. Specifically, LLMs were used to (i) improve the language, structure and readability of the manuscript, particularly for concise language in order to shorten the text, (ii) support the search for related work, (iii) format tables. LLMs were not used to design the method, run the experiments or generate experimental results. All references suggested by LLMs were checked against the original sources, and all LLM-generated text and code were reviewed and verified by the authors, who take full responsibility for the content of this paper.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, 2017.

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546, 2005.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

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

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

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, 2019.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6894–6910, 2021.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

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

Rita González-Márquez, Philipp Berens, and Dmitry Kobak. Cropping outperforms dropout as an augmentation strategy for self-supervised training of text embeddings. *Transactions on Machine Learning Research*, 2026. arXiv:2508.03453.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*, 2015.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 9459–9474, 2020.

Xianming Li and Jing Li. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1825–1839, Bangkok, Thailand, 2024.

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

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, 2013.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, Dubrovnik, Croatia, 2023.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014.

Nils Reimers and Iryna Gurevych. 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)*, pp. 3982–3992, Hong Kong, China, 2019.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations (ICLR)*, 2021.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations, ICLR 2016*, San Juan, Puerto Rico, 2016.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. InfoCSE: Information-aggregated contrastive learning of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022a.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, 2022b.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. CLEAR: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020.

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

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.