Label-Aware Pseudo-Training Sample Generation for Text Classification

ARASH YOUSEFI JORDEHI, Department of Computer Engineering, University of Guilan, Iran
SEYED ABOLGHASEM MIRROSHANDEL*, Department of Computer Engineering, University of Guilan, Iran
OWEN RAMBOW, Department of Linguistics, Stony Brook University, USA and Institute for Advanced Computational Science, USA

Deep learning models excel in various Natural Language Processing (NLP) tasks, but their performance (excluding approaches like zero-shot learning or few-shot learning) relies on ample data, posing challenges in fields with limited datasets. To address the poverty in the size of training data, a number of approaches could be taken, such as multi-task learning and data augmentation. Aiming to leverage Large Language Models (LLMs), we propose a data augmentation algorithm. It subtly alters sentences by inserting random words and utilizes LLMs to find the most fitting replacements within their embedding space. Taking inspiration from Prompt Tuning, the focus shifts from optimizing the input prompt to updating the inserted tokens' embedding vectors by maximizing the conditional generation probability. This allows for vast sample generation while implicitly benefiting from the knowledge within LLMs. The results from our extensive set of experiments on various benchmark text classification tasks show a substantial improvement over the non-augmented outcomes.

JAIR Associate Editor: Ndapa Nakashole

JAIR Reference Format:
Arash Yousefi Jordehi, Seyed Abolghasem Mirroshandel, and Owen Rambow. 2026. Label-Aware Pseudo-Training Sample Generation for Text Classification. *Journal of Artificial Intelligence Research* 85, Article 22 (February 2026), 27 pages. DOI: 10.1613/jair.1.20868

1 Introduction

Today's digital landscape is overflowing with text data (Melsbach et al. 2025), a key component of many Natural Language Processing (NLP) applications (Kesgin and Amasyali 2024). At the intersection of artificial intelligence (AI) and computational linguistics, NLP is the bedrock of how computers and humans interact through language (Luz 2022; Tsujii 2011). This technology is vital for numerous language-based tasks, such as translation, summarization, and question answering, all of which enable computers to process and generate language with remarkable efficiency. NLP underpins many of the daily-use applications we rely on, from virtual assistants to real-time translation services. Yet, the efficacy of NLP models is intrinsically linked to the availability of extensive, high-quality, annotated datasets (J. Chen et al. 2023b; A. Mumuni and F. Mumuni 2022; Z. Wang, J. Zhang, et al. 2025; Q. Xie et al. 2020b; S. Yu et al. 2024). Acquiring such datasets is often expensive and tedious, posing

*Corresponding Author.

Authors' Contact Information: Arash Yousefi Jordehi, ORCID: 0000-0001-8136-2246, arashy76@phd.guilan.ac.ir, Department of Computer Engineering, University of Guilan, Rasht, Guilan, Iran; Seyed Abolghasem Mirroshandel, ORCID: 0000-0001-8853-9112, mirroshandel@guilan.ac.ir, Department of Computer Engineering, University of Guilan, Rasht, Guilan, Iran; Owen Rambow, ORCID: 0000-0003-2054-039X, owen.rambow@stonybrook.edu, Department of Linguistics, Stony Brook University, Stony Brook, New York, USA and Institute for Advanced Computational Science, Stony Brook, New York, USA.

DOI: 10.1613/jair.1.20868

significant challenges, particularly for under-resourced languages and specialized domains (Ding, Qin, L. Liu, et al. 2022; Onan 2023).

As a strategic response to data scarcity, Data Augmentation (DA) enhances existing corpora by generating linguistically diverse and semantically coherent variants, thereby improving model robustness, generalization, and performance across a wide range of tasks (J. Chen et al. 2023b; Ding, Qin, R. Zhao, et al. 2024; Feng et al. 2021a). Methods such as contextual replacement (Kobayashi 2018), back-translation (Sajjadi et al. 2016), and neural style transfer (Gatys et al. 2015) have demonstrated consistent gains in applications including text classification (Cubuk, Zoph, Mané, et al. 2019), information extraction (Nguyen et al. 2020), and sentence representation learning (Thakur et al. 2020). This makes DA especially critical for training data-intensive deep learning models, where insufficient data can lead to overfitting and poor out-of-domain performance (Cheng et al. 2019; Cubuk, Zoph, Mane, et al. 2019; Dosovitskiy et al. 2016; Miyato et al. 2021).

Our research focuses on DA for text classification within NLP, proposing a novel embedding-space method for tasks such as sentiment analysis and subjectivity classification. Unlike conventional approaches that generate surface-level text variants using Large Language Models (LLMs), our technique operates in the latent space by introducing a small set of learnable artificial tokens into the input representation. This strategy is motivated by findings that even minor perturbations can significantly influence model predictions (Balashankar et al. 2023). To preserve label integrity, our method is explicitly designed to be label-aware: the embeddings of the artificial tokens are optimized to maximize the likelihood of the original class label under the perturbed input. By avoiding surface-form generation, our approach sidesteps the risks of semantic drift and hallucination associated with LLM-based augmentation (L. Cao et al. 2024; Sun et al. 2025), while enhancing model generalization and classification accuracy—contributing to more robust and reliable NLP systems (H. Chen, Han, et al. 2022; Kwon and Y. Lee 2023; Onan 2023).

The core of our approach is a conditional generation mechanism that dynamically integrates the inserted tokens into the original text. Unlike traditional methods that rely on predefined rules or extensive preprocessing, our model updates the embeddings of these new tokens, allowing them to blend naturally with the surrounding context. This ensures that the augmented sentences remain semantically and syntactically coherent, preserving the integrity of the original label and context.

LLMs are particularly well-suited for this task. From a data perspective, using LLMs for DA offers a powerful strategy for overcoming the limitations of manually curated datasets. These models can facilitate the creation of high-quality synthetic data that, in certain instances, can surpass the value of human-generated data (Ding, Qin, R. Zhao, et al. 2024; Peng et al. 2023). This capability makes LLMs an ideal choice for generating diverse and realistic augmented examples, which is crucial for improving the robustness and generalization of NLP models.

Our methodology utilizes the capabilities of LLMs to maintain the contextual semantics of the original text while simultaneously introducing diversity into the training dataset. This integration of tokens is accomplished by optimizing their embeddings within the LLM's latent space, ensuring alignment with the surrounding context to uphold linguistic coherence. In contrast to conventional data augmentation techniques that depend on rule-based transformations or fixed substitutions, our approach dynamically adapts to the subtleties of the input text, facilitating a more flexible and robust augmentation process.

Additionally, the label-awareness inherent in our technique marks a significant advancement in DA practices. By conditioning the updates of embeddings on the target label, the augmented samples generated are specifically designed to enhance the model's comprehension of the task at hand. This label-conditioned augmentation not only improves the model's generalizability but also reduces the likelihood of introducing noise that could negatively impact performance.

Another advantage of our method is its scalability. The random token insertion mechanism can be modified to create multiple augmented versions of the same input, yielding an almost limitless supply of training samples. This scalability is especially beneficial in low-resource contexts, where acquiring labeled data can be particularly

challenging. By enriching the dataset with high-quality, contextually coherent samples, our technique effectively addresses issues related to data scarcity without necessitating extensive human annotation efforts.

In conclusion, our proposed DA technique represents a transformative shift in strategies for NLP tasks. By leveraging the strengths of LLMs, it combines contextual fidelity, label-awareness, and scalability to generate augmented data that significantly enhances model performance across various text classification benchmarks. This innovative approach paves the way for new research opportunities and applications, particularly in areas where data availability poses a significant constraint.

For the evaluation of our proposed method, we have employed multiple benchmark datasets for text classification tasks. These datasets, widely recognized and utilized in the field, provide diverse and representative samples that allow for a comprehensive assessment of our method's performance. This procedure is done using a pre-trained LLM which implicitly has a deep understanding of language itself (Aparna et al. 2025; Jamal et al. 2024). The main contributions of this research can be summarized as follows:

I. Proposing a novel prompt-tuning-style DA framework, which is inspired by the idea of "Prompt Tuning" (Lester et al. 2021) and "Automatic Prompt Construction" (Yousefi Jordehi et al. 2024).
II. Employing LLMs to introduce a new method of Transfer Learning in DA by tapping into the language knowledge embedded within them.
III. Generating new samples in the embedding space rather than finding exact words in vocabulary.
IV. Conducting experiments on text classification benchmarks, we outperformed standard fine-tuning methods by using our generated augmented data in conjunction with the original training data.
V. Establishing a new state-of-the-art (SOTA) record or reaching very close to the current SOTA in various benchmarks.

The remainder of this paper is organized as follows. Section 2 reviews related work on DA in NLP, situating our contribution within both traditional and LLM-based approaches. Section 3 introduces the proposed method in detail, outlining the stages of artificial token insertion and label-aware perturbation realization. Section 4 describes the experimental setup, datasets, and implementation details, followed by Section 5, which presents and analyzes the results. Section 6 reports ablation studies, including robustness and sensitivity analyses. Finally, Section 7 concludes the paper with a summary of contributions and directions for future work.

2 Related Work

This section provides a structured overview of the DA landscape in NLP, situating our work within the broader research trajectory. We first survey **traditional and non-LLM-based methods**, which range from simple word-level heuristics to sophisticated techniques leveraging contextual embeddings and structural recombination. We then analyze the paradigm shift brought about by **LLM-based augmentation**, where large language models act as generators, rewriters, and curators of synthetic text. While powerful, these methods inherit fundamental challenges related to semantic drift, computational cost, and the statistical irregularities of generated text. Finally, we detail the **positioning of our contribution**: a latent-space augmentation framework that circumvents surface-text generation altogether. By repurposing a frozen LLM as a contextualizer rather than a generator, our method offers a parameter-efficient, semantically grounded, and statistically robust alternative that addresses key limitations of both classical and contemporary approaches.

2.1 Traditional and Modern Data Augmentation without LLMs

DA in NLP has long developed independently of large language models, relying on heuristic, structural, and representation-driven methods that directly manipulate surface text or latent features. Classical techniques such as synonym substitution, insertion, deletion, and token shuffling (Wei and Zou 2019; Woolsey et al. 2025; X. Zhang et al. 2015) provided simple and broadly applicable ways to expand datasets, but they frequently risked

semantic drift when applied to polysemous or domain-specific terms. Back-translation (Sennrich et al. 2016) introduced a sentence-level alternative by translating into a pivot language and retranslating to the source, producing paraphrases that improved classification robustness across domains, with later refinements incorporating multiple pivots or controlled noise to enhance diversity (Fabbri et al. 2021; Q. Xie et al. 2020a; A. W. Yu et al. 2018). More recently, domain-aware rule-based approaches have extended these ideas; for instance, H. Chen, Dan, et al. (2024) proposed contextual random replacement and targeted entity replacement to improve Chinese medical Named Entity Recognition (NER), showing that carefully constrained replacements yield measurable gains in specialized applications. Similar attempts to increase semantic richness under limited data conditions include variance-oriented masking, where Yao et al. (2024) introduced M4DA to generate label-consistent augmentations by selecting high-variance tokens, demonstrating clear improvements in class-imbalanced text classification.

As pre-trained encoders and contextual embeddings gained prominence, augmentation methods shifted toward embedding-driven and context-aware strategies. Embedding-based word replacement (Rizos et al. 2019; W. Y. Wang and D. Yang 2015) expanded lexical variety by leveraging distributional similarity, while contextualized masked prediction with Bidirectional Encoder Representations from Transformers (BERT) improved fluency by conditioning replacements on sentence context (Kobayashi 2018; X. Wu et al. 2019). Structural composition also became important: TreeMix (L. Zhang et al. 2022) recombines syntactic subtrees from constituency parses to produce syntactically diverse yet coherent sentences, and AttentionMix (Lewy and Mańdziuk 2023) uses BERT attention weights to guide token-level mixing, balancing diversity with semantic preservation. Representation-level augmentation followed in this trajectory, as exemplified by AugCSE (Z. Tang et al. 2022), which integrates multiple augmentation types into a contrastive framework to improve the quality of sentence embeddings and their transferability across tasks. Parallel work explored adversarial perturbations (Ebrahimi et al. 2018), gradient-guided token substitutions, and noise injection strategies (Y. Li et al. 2017; Z. Xie et al. 2017) as regularizers to improve model resilience to minor textual variation.

Another line of research explores the use of fine-tuned language models for controlled text generation in data augmentation. Bayer, Kaufhold, Buchhold, et al. (2023) propose a method that employs a Generative Pre-trained Transformer-2 (GPT-2) model, fine-tuned on the target dataset and guided by a specific prefix, to generate synthetic training samples. To ensure label consistency, their approach incorporates a document embedding filter to remove generated instances that are semantically misaligned with their intended class. This work demonstrates the effectiveness of combining generative models with filtering mechanisms to produce high-quality augmented data for text classification.

At the task and domain level, augmentation pipelines have been designed to exploit structural constraints and domain knowledge. In legal contract classification, Duffy et al. (2025) showed that combining rule-based transformations with lightweight generative methods led to dramatic F1 improvements even when training data was reduced by up to 75%. In information retrieval, Moon et al. (2025) proposed frequency-based token deletion strategies that adapt deletion probability to both passage- and corpus-level statistics, producing more informative and balanced training corpora for dense retrieval. Revisiting the value of simple approaches, Parmar (2025) demonstrated that ensembles of synonym swaps, insertions, deletions, and back-translation provide consistent improvements in classification tasks under data scarcity, reinforcing the enduring utility of lightweight techniques. Furthermore, Cegin et al. (2025) systematically compared traditional augmentation with LLM-based paraphrasing, finding that established strategies often match or outperform LLM-generated augmentations when sufficient seed data is available, while being far cheaper, faster, and environmentally sustainable. Collectively, these lines of research show that even without LLMs, data augmentation continues to evolve through domain adaptation, structural recomposition, embedding manipulation, and contrastive regularization, forming a robust methodological foundation for semantic preservation, diversity, and robustness that informs and complements more recent LLM-based approaches.

2.2 LLM-Based Data Augmentation

The rapid development of LLMs such as Generative Pre-trained Transformer-3 (GPT-3), Generative Pre-trained Transformer-4 (GPT-4), and Large Language Model Meta AI (LLaMA) has opened a new paradigm for data augmentation in NLP (Sun et al. 2025), shifting the focus from surface-level manipulations and contextual substitutions to controllable generation of semantically coherent and context-rich synthetic data. Unlike heuristic or embedding-driven methods that operate primarily on existing corpora, LLM-based augmentation relies on prompt engineering, in-context learning, and task conditioning to produce novel samples that expand training distributions in both lexical and structural dimensions (Achiam et al. 2023; Brown et al. 2020; Touvron et al. 2023; M. Wang et al. 2024). These capabilities have been particularly impactful in low-resource and few-shot learning scenarios, where even modest amounts of high-quality synthetic data can substantially improve generalization (Balkus and Yan 2024; Dai et al. 2025; Gera et al. 2022; R. Zhang et al. 2023).

Recent comparative studies highlight both the promise and limitations of this approach. Pavlyshenko and Stasiuk (2025) contrasted traditional augmentation techniques such as synonym replacement, contextual embeddings, and abstractive summarization (the Lambada method) with LLM-driven augmentation using GPT-4, LLaMA-3, and Mistral, showing that while LLM-based methods can boost classification accuracy, simpler approaches sometimes rival or surpass them when carefully aligned with task structure and model architecture. In the context of imbalanced classification, H. Zhao et al. (2024) demonstrated that generating new samples from scratch using ChatGPT is often more effective than rewriting existing ones, particularly in technical domains where preserving specialized terminology is critical; however, they also observed that rewriting tends to induce semantic drift by replacing domain-specific expressions with generic alternatives. A hybrid strategy that balances generated and rewritten data was found to improve robustness while mitigating distributional artifacts, further underscoring the importance of prompt specificity and controllability.

A distinct approach to LLM-based data augmentation is demonstrated by Ubani et al. (2023), who use large language models in a zero-shot, instruction-driven manner to generate synthetic data for low-resource text classification. Their method involves crafting imperative prompts that explicitly instruct the LLM to generate a specified number of sentences for each class (e.g., "Generate 20 sentences where a human tells a digital assistant to add music to a playlist"). This approach is evaluated in a few-shot scenario with only 10 labeled examples per class, and the generated data is shown to have low similarity to the original training set, suggesting it mitigates memorization risks. This work highlights the potential of using direct, class-specific instructions to guide LLMs for targeted data synthesis.

Other lines of work focus on semantic enrichment and preprocessing before generation. Chi et al. (2024) proposed a framework for financial short-text classification where input texts are first enriched with label-correlated contextual information before augmentation, thereby reducing noise and ambiguity in the resulting synthetic data. Similarly, Meguellati et al. (2025) reframed augmentation as an explanation-driven process, using LLMs to clean noisy inputs and attach contextual triggers or rationales that improve classification performance without substantially expanding the dataset. These approaches highlight a growing trend in which LLMs act not only as generators but also as rewriters and explainers, with the goal of enhancing semantic consistency rather than maximizing diversity.

Another prominent strand leverages LLMs for cross-lingual or domain-conditioned augmentation. Rahman and Siddiq (2025) applied translation and retranslation of code comments across multiple languages, filtering paraphrases with semantic similarity models such as Sentence-BERT to preserve meaning while increasing linguistic diversity, though diminishing returns were observed as the number of pivot languages grew. Y.-L. Chung et al. (2025) scaled augmentation to millions of claim-generation pairs across English, German, and Spanish by iteratively prompting LLMs to produce supports, refutes, and not-enough-info statements validated against Wikipedia passages, yielding improvements in knowledge-intensive reasoning tasks but also raising questions

about quality assurance and calibration at scale. More domain-focused applications include Arslan et al. (2025), who generated idiom corpora with GPT-4, showing cost and time advantages but documenting persistent gaps in semantic fidelity relative to human-authored data, and Almorjan et al. (2025), who fine-tuned GPT-3.5 with domain-specific indicators of compromise to synthesize labeled social-media comments, producing realistic and controllable task-specific data while facing the challenge of distributional artifacts from surface-level generation.

Building on these empirical advances, recent comprehensive surveys have systematized the landscape of LLM-based augmentation. Chai et al. (2025) categorize over sixty methods along dimensions of strategy, granularity, and evaluation. Complementing this, Ding, Qin, R. Zhao, et al. (2024) propose a unified taxonomy based on data perspectives: (1) *Data Creation* (generating synthetic datasets via few-shot prompting), (2) *Data Labeling* (using LLMs to annotate unlabeled data), (3) *Data Reformation* (paraphrasing or counterfactual generation), and (4) *Co-Annotate* (LLM-human collaboration). This framework clarifies the functional roles of LLMs and highlights their use beyond mere generation.

A key insight from these surveys is the growing **application** of LLM-based DA in *low-resource and multilingual settings*. Z. Wang, P. Wang, et al. (2024) demonstrate that LLMs can generate high-quality training data for under-resourced languages, significantly improving performance in tasks like sentiment analysis and named entity recognition where labeled data is scarce. Their method leverages multilingual LLMs (e.g., mT5, BLOOM) to generate and translate data, enabling cross-lingual transfer without human annotation.

Complementing these, H. Chen, Han, et al. (2022) introduce DoubleMix, a hidden-space augmentation method that combines MixUp (H. Zhang et al. 2017) with manifold interpolation. Rather than generating surface-level text, DoubleMix creates synthetic training examples by interpolating between input embeddings and their corresponding labels in the latent space. This approach avoids the pitfalls of surface-form generation—such as hallucination, semantic drift, and decoding overhead—while promoting smoother decision boundaries and improved generalization. Their results show consistent gains across six benchmark datasets, particularly in low-resource settings.

In contrast to DoubleMix, which generates new training signals by interpolating between the original input and its perturbed versions (e.g., via synonym replacement or back-translation), our method preserves the original input sequence entirely. Instead of modifying or blending existing samples, we enrich the input by inserting a small number of learnable artificial tokens at random positions. The embeddings of these tokens are then optimized to maximize the likelihood of the correct label, effectively creating a label-aware perturbation in the embedding space. This approach does not require generating perturbed variants of the data, avoids any form of interpolation, and introduces minimal structural change to the input. By focusing on learning a compact set of task-specific soft prompts within the frozen LLM, our method offers a more targeted, parameter-efficient, and semantically stable form of latent-space augmentation.

Finally, Feng et al. (2021b) and J. Chen et al. (2023b) provide comprehensive empirical surveys of data augmentation in NLP, summarizing token-level, sentence-level, adversarial, and hidden-space methods. They conclude that while augmentation generally improves robustness, its effectiveness is highly dependent on task, model, and data regime, underscoring the need for principled, evaluation-driven frameworks.

Taken together, the literature on LLM-based augmentation demonstrates a spectrum of roles for LLMs, ranging from generators of fully synthetic instances and paraphrases, to rewriters that restructure or enrich existing inputs, to explainers that attach rationales or contextual triggers, and to curators that filter, relabel, or validate candidate outputs. These methods consistently deliver benefits in data-scarce and fine-grained classification scenarios, but they come with practical trade-offs in controllability, efficiency, privacy, and quality assurance. Their reliance on prompt engineering and validation heuristics reflects both their flexibility and fragility. Nevertheless, LLM-based augmentation marks a major advance over earlier paradigms by enabling task-conditioned, semantically rich data generation at scale, setting the stage for complementary methods—such as latent-space

augmentation—that seek to preserve the benefits of contextual sensitivity while mitigating the drawbacks of surface-form generation.

2.3 Positioning of This Work

While recent advances in LLM-based data augmentation have demonstrated impressive performance gains—often achieved through paraphrasing, explanation generation, or full synthetic sampling—they share a fundamental dependency on surface-form text generation. This dependency necessitates extensive prompt engineering, autoregressive decoding, and laborious post hoc filtering to ensure semantic fidelity and label consistency. These methods treat the LLM primarily as a generator or rewriter of natural language, and their effectiveness hinges on careful control of stochastic outputs, mitigation of vocabulary bias, and quality assurance via downstream heuristics. Consequently, they face persistent challenges in **label preservation**, **computational efficiency**, and **scalability**. These challenges are particularly acute in low-resource, domain-specific, or privacy-sensitive settings, where uncontrolled generation can introduce noise, semantic drift, or distributional artifacts.

A growing body of work highlights deeper, more fundamental concerns regarding the linguistic authenticity of LLM-generated text. Despite achieving high fluency, synthetic samples often deviate from core statistical regularities inherent to human language, such as Zipf's law (governing word frequency distributions), Heaps' law (describing vocabulary growth), and Mandelbrot's multifractal scaling properties (Z. Wang, G. Xu, et al. 2024). These deviations reflect a systemic deficit in long-range correlations and structural complexity, suggesting that even semantically coherent outputs may lack the rich, self-similar organization of natural discourse. To mitigate this, frameworks such as ZGPTDA (Z. Wang, G. Xu, et al. 2024) have proposed quality-aware augmentation using fuzzy logic and Z-numbers to model both the truth (i.e., conformity to scaling laws) and the reliability of generated samples. While such approaches can improve downstream task performance by filtering statistically atypical outputs, they introduce significant computational overhead and remain entrenched in the inherently flawed **generate-then-correct** paradigm, which relies on costly decoding and external validation.

In stark contrast, our proposed method circumvents the need for the LLM to emit any surface text whatsoever. Instead, we operate entirely within the latent representation space of a *frozen* encoder-decoder architecture. Our core innovation involves the introduction of a small set of artificial tokens whose embedding vectors are optimized to induce label-preserving perturbations within the input's representation. By learning only these token embeddings—while keeping the entire pre-trained model fixed—we reframe data augmentation as a controlled, deterministic intervention within the embedding manifold. This approach entirely avoids the stochasticity, prompt sensitivity, and decoding variability that are intrinsic to generative methods.

This design confers several distinct advantages:

- **Immunity to Statistical Artifacts:** Since no text is generated, our method is inherently immune to violations of linguistic scaling laws and other statistical anomalies that plague synthetic text. There is no need to assess conformity to Zipfian distributions or fractal complexity, as our pseudo-samples exist exclusively within the semantic representation space.
- **Semantic Grounding:** Augmentation is performed within a context-aware, semantically grounded representation space, ensuring that perturbations respect both local syntactic structures and global semantic meaning.
- **Parameter Efficiency and Privacy:** The approach is highly parameter-efficient, as the vast majority of the network (the LLM itself) remains completely frozen. Only a minimal number of new token embeddings are learned. This avoids the computational cost and catastrophic forgetting risks of full model fine-tuning and eliminates privacy risks associated with generating and storing new text—making it particularly suitable for sensitive or regulated domains.

Furthermore, by strategically enriching neighborhoods around existing samples within the embedding manifold, our method effectively densifies sparse regions of the decision boundary. This enhances model robustness to adversarial perturbations and improves generalization, especially in data-scarce regimes. It is therefore particularly effective for applications involving **fine-grained classification**, **imbalanced datasets**, and **specialized lexicons**—precisely the scenarios where generative augmentation often fails due to semantic drift or overgeneralization.

Thus, whereas prior work leverages LLMs as *generators*, *rewriters*, or *curators* of text—each role entailing significant trade-offs between diversity, fidelity, and computational cost—our research repurposes the LLM as a fixed *contextualizer*. We exploit its frozen representational capacity not to produce language, but to define a rich, structured space in which augmentation becomes a precise, label-aware operation. In doing so, we move beyond the limitations of the generate-filter paradigm and propose a new direction: **augmentation as latent-space sculpting**, guided by task semantics rather than surface fluency.

Our contribution represents a conceptual and practical shift in data enrichment for NLP—away from explicit text generation and toward implicit representation engineering. It complements both traditional and LLM-based methods by offering a more efficient, stable, and theoretically sound alternative that preserves the benefits of deep contextual sensitivity while eliminating the core liabilities of surface-form synthesis. This literature further motivates our novel, scalable, and label-aware method for embedding-space data augmentation that complements and extends previous DA paradigms in NLP.

3 Proposed Method

In this section, we present a detailed explanation of our proposed method, breaking it down into distinct stages to facilitate understanding. Each stage is carefully designed to address specific aspects of the methodology, ensuring clarity and coherence in its implementation.

3.1 Perturbations Insertion

A dataset $\mathscr{D}$ is typically partitioned into training ($\mathscr{D}^{\text{train}}$), development ($\mathscr{D}^{\text{dev}}$), and test ($\mathscr{D}^{\text{test}}$) subsets, although in some cases only a training and test split is provided. Our proposed DA is applied exclusively to $\mathscr{D}^{\text{train}}$.

We begin by selecting a subset of M samples from $\mathscr{D}^{\text{train}}$, where $M \leq N$ and N denotes the total number of training instances. This subset serves as the operational scope for our augmentation method. The algorithm then proceeds by systematically applying a defined transformation procedure to each of the M selected samples[1].

For a given sentence $s = (w_1, w_2, \ldots, w_n)$, we insert a set of K **artificial tokens** at randomly selected positions within the sequence. Formally, this involves generating a sequence of K distinct indices, sampled uniformly from the possible token positions, indicating where the artificial tokens are to be injected.

Consequently, we denote the augmented sentence as $s' = w_1, w_2, ..., w_{n+K}$. The K artificial tokens are initially chosen from the lexicon of the LLMs, symbolized as $\mathscr{V}$. This study departs from the "soft prompt-tuning" method of Lester et al. (2021) by introducing the novel concept of mid-sentence prompts, where prompts can be strategically placed at any desired location within the text.

We are inspired by the significant research of Yousefi Jordehi et al. (2024), who developed a prompt construction technique that locates learnable prompts in the embedding space to improve opinion mining tasks. However, we diverge from their approach by extending the identification of prompt words to every preferred position (i.e., index) within a sentence. Additionally, we differ from their methodology by focusing on generating new samples in the embedding space to increase the number of data points (i.e., sentences) as a novel DA technique,

[1]When augmenting data to a larger size ($M > N$), we only need to vary the indices generated in each augmentation run. Essentially, we achieve this by applying "sampling with replacement" method. Consequently, we can generate as much data as necessary to reach the desired amount M.

rather than identifying a specific prompt for a given task. While our primary strategy involves random token insertion, we recognize that position can influence how artificial embeddings interact with the surrounding context. Inserting tokens at the beginning may bias sentence-level semantics, while mid-sentence placement allows perturbations to blend more naturally into the structure. To avoid overfitting to any single strategy, we employ random placement across the sequence, which distributes perturbations throughout the embedding space and maximizes diversity. In Section 6.2, we further analyze the effect of token position and demonstrate that moderate values of K with varied placement preserve syntactic coherence and yield consistent performance gains. The overall procedure of our system is shown in Figure 1.

We are also different from other previous DA conditional generation techniques which involve producing supplementary text using a language model that is conditioned on a specific label. Once the model is trained to reconstruct the original text based on the provided label, it can subsequently generate new text instances (Anaby-Tavor et al. 2020; Kumar et al. 2019; Niu and Bansal 2018).

Although random insertion may appear to disrupt surface syntax, the frozen LLM contextualizer interprets artificial tokens as latent perturbations rather than grammatical elements. As a result, sentence coherence is preserved at the representational level, even when tokens are placed at positions that would otherwise break syntax in surface text.

3.2 Perturbations Realization

In this work, we employ the Text-to-Text Transfer Transformer (T5) sequence-to-sequence model (Raffel et al. 2020) as the foundational LLM for realizing perturbations within input sentences. Our approach involves optimizing the embedding vectors of a small set of artificially inserted tokens that act as controlled perturbations. Crucially, during training, **only** the embeddings of these artificial tokens are updated through back-propagation, while all other model parameters—including the entire LLM—remain frozen. This selective optimization leverages the rich linguistic and semantic knowledge already encoded in the frozen model, enabling the injected tokens to integrate seamlessly into the existing sentence structure and meaning without compromising the original context.

Our method operates within a **conditional generation** framework, a core principle in modern machine learning where an output is produced based on a specific input. Formally, this can be expressed as maximizing the conditional probability $Pr_\theta(\text{output} \mid \text{input})$.

In our case, the "input" is the original sentence with the inserted artificial tokens (the modified sequence), and the "output" is the desired ground-truth label. Label-awareness is incorporated directly into the optimization objective. Specifically, the embeddings of the artificial tokens are updated so as to maximize the conditional likelihood of the ground-truth label given the perturbed input. This means that gradients flow only through the artificial tokens with respect to the label-specific loss, ensuring that the learned perturbations are explicitly tied to preserving and reinforcing the correct label. In this way, augmentation is not merely stochastic noise injection, but a label-conditioned intervention in the representation space.

Unlike embedding-based replacement methods, our approach does not decode optimized vectors back into discrete vocabulary tokens. Instead, the artificial embeddings are retained as continuous vectors and directly integrated into the LLM's latent space. This design eliminates the risk of producing semantically irrelevant or out-of-vocabulary tokens, since no surface decoding is performed at any stage of augmentation. The augmented samples therefore remain label-consistent and semantically grounded by construction, without requiring post-decoding filtering.

This targeted approach is highly effective because it focuses the model's objective on maintaining the original label. Unlike traditional fine-tuning where the entire model's parameters (θ) are updated, our approach strictly

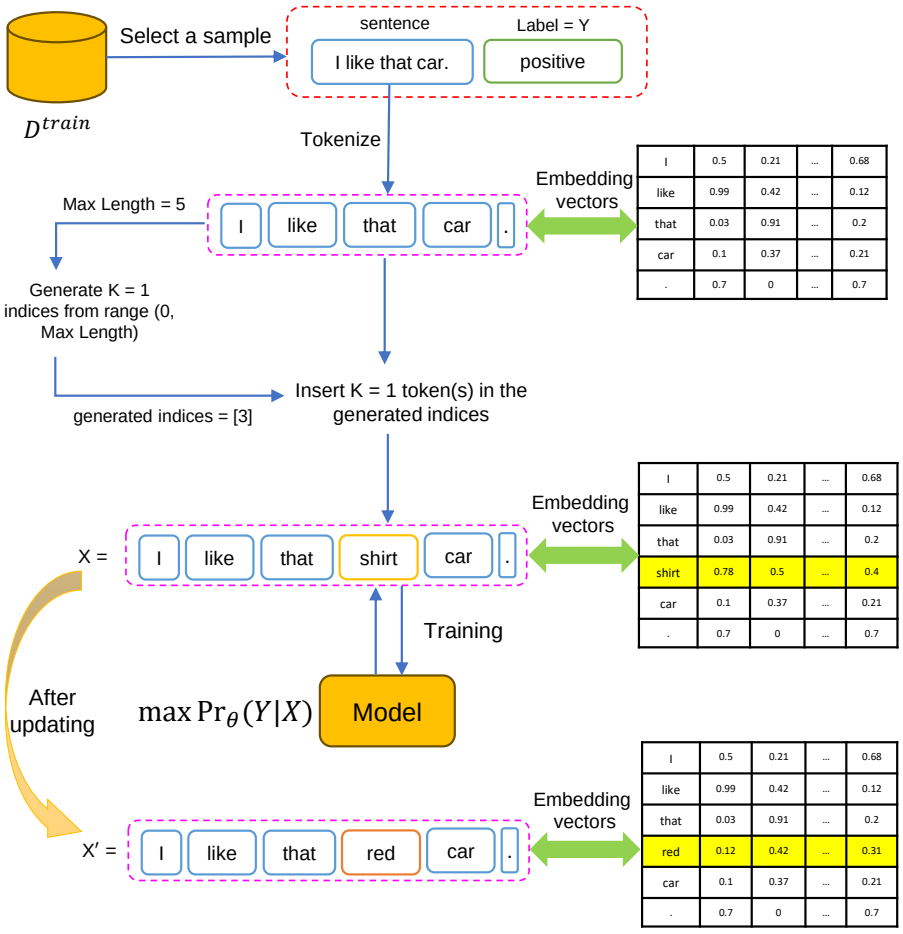

Fig. 1. A conceptual view of our methodology, shown here for a single training pair. The numbers and words depicted are purely illustrative and do not correspond to actual decoded tokens. In practice, the artificial tokens exist only as learned embeddings in the latent space and are never converted back into surface words. This representation is provided solely to visualize the overall augmentation process.

confines parameter updates to the embedding vectors of the artificial tokens. This creates a limited parameter subspace for optimization, allowing us to learn specific perturbations without altering the core model.

By isolating the optimization to these new tokens, we achieve a powerful, yet efficient, form of learning. The model learns to adjust the embeddings of the artificial tokens, effectively "blending" them into the sentence in a way that allows the final modified sequence to be correctly classified with its original ground-truth label. This ensures that:

- **Label-Awareness is Maintained:** The learned perturbations are directly tied to the desired outcome of correct classification, preventing the model from introducing changes that would alter the original sentence's label.

- **Core Model Integrity is Preserved:** We avoid unnecessary modifications to the core LLM, preventing catastrophic forgetting and ensuring the model's general language capabilities remain intact.
- **Computational Efficiency:** Limiting the updated parameters significantly reduces computational costs and training time compared to full fine-tuning.

In essence, **conditional generation** provides the perfect mechanism for our technique, allowing us to precisely control the augmentation process and learn meaningful, label-preserving perturbations without retraining the entire large language model. This targeted and efficient approach is what makes our method both effective and scalable for data augmentation.

Concretely, for each training instance, the perturbations correspond to a tensor in $\mathbb{R}^{K\times d}$, where K is the number of artificial tokens and d the embedding dimension of the LLM. For a total of M augmented samples, the overall trainable parameter space forms $\mathbb{R}^{M\times K\times d}$. This parameterization is illustrated schematically in Figure 1 and Algorithm 1, which detail the process of embedding optimization and integration into the input sequence.

Our approach introduces a novel and effective paradigm for latent-space augmentation, distinct from existing methodologies. Unlike DoubleMix, which generates synthetic training signals through interpolation between an original input and its perturbed variants—created via operations such as synonym replacement or back-translation—our method preserves the integrity of the original input sequence without modification. Rather than altering or combining existing samples, we enhance the input by injecting a small set of learnable artificial tokens at random positions. The embeddings of these tokens are then optimized to maximize the conditional likelihood of the true label, thereby introducing a label-aware perturbation within the embedding space. This strategy eliminates the need for generating auxiliary perturbed data, avoids interpolation altogether, and ensures minimal disruption to the input structure, offering a more direct and controlled form of augmentation.

Unlike approaches that attempt to decode learned vectors back into discrete vocabulary words, our method exclusively leverages the contextualized latent representations of these embeddings. This strategy circumvents the intrinsic limitations of mapping to single words, which often fail to capture fine-grained semantic nuances and may exhibit low similarity to any vocabulary token. In effect, the artificial embeddings serve as purposeful, structured perturbations that enrich the representational neighborhood around each training instance, enhancing model robustness through expanded diversity.

While drawing on the paradigm of conditional generation, our approach fundamentally differs from adversarial perturbations (Niu and Bansal 2018) and class-conditioned decoders (Anaby-Tavor et al. 2020) by restricting learning solely to token embeddings within a frozen model framework. This restriction achieves controlled, label-aware diversity and aligns well with recent empirical findings on the quality and novelty of augmented samples (J. Chen et al. 2023a).

Our work is inspired by the emerging paradigm of automatic prompt construction (Yousefi Jordehi et al. 2024), where a small set of parameters (a prompt) is learned to condition a frozen LLM for a downstream task. However, we extend this idea in a significant way: rather than learning a prompt that is prepended to the input, we learn the embeddings of artificial tokens that are *injected* into the input sequence itself. This allows the perturbation to be distributed throughout the input, creating a more integrated and context-sensitive augmentation. By focusing on learning a compact set of task-specific soft prompts within the frozen LLM, our method offers a more targeted, parameter-efficient, and semantically stable alternative to both surface-level generation and hidden-space interpolation.

Importantly, our framework is highly flexible and can be adapted to diverse encoder-decoder architectures within the LLM domain, making it broadly applicable.

Mathematically, the training update for the learnable artificial token embeddings is formulated as the minimization of the standard encoder-decoder loss. The process begins by selecting M samples from the training set $\mathscr{D}^{\text{train}}$. For each sample, K insertion positions $\mathscr{I} = \{i_1, \ldots, i_K\}$ are randomly sampled and recorded. During the

training phase, the model processes the original input sequence by dynamically inserting the current state of the K artificial token embeddings $\mathbf{p}_1, \ldots, \mathbf{p}_K$ at these pre-stored positions, forming the augmented embedding sequence $\mathbf{s'}_e$. The forward and backward passes are then executed as follows:

$$\mathbf{h}_e = \text{Encoder}(\mathbf{s'}_e)$$

$$\mathbf{y} = \text{Decoder}(\mathbf{h}_e)$$

$$\mathbf{p}_i \leftarrow \mathbf{p}_i - \eta \nabla_{\mathbf{p}_i} \mathscr{L}(\mathbf{y}, \mathbf{t}), \quad \forall i \in \{1, \ldots, K\}$$

where $\mathbf{s'}_e$ is the sequence of embeddings with the artificial tokens inserted, $\mathbf{p}_i$ are the embeddings being optimized, η is the learning rate, and $\mathscr{L}$ is the loss function comparing the model output $\mathbf{y}$ to the target $\mathbf{t}$.

This optimization process continues until the training loss converges below a predefined threshold. At this point, the learning phase for the artificial tokens is complete. The final, optimized embedding vectors $\{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$, along with their corresponding insertion positions $\{\mathscr{I}_j\}$ for each of the M augmented samples, are stored as a compact augmentation module. Crucially, it is these learned *embedding vectors*—not discrete tokens—that are preserved. This avoids the ill-posed problem of decoding a continuous vector back into a single, potentially semantically impoverished word, which may not adequately represent the learned perturbation.

In the subsequent downstream fine-tuning stage, this stored module is applied: for each of the M selected training samples, the original input sequence is retrieved, and the learned embedding vectors $\mathbf{p}_i$ are directly inserted at their pre-recorded positions $\mathscr{I}_j$ within the embedding space.

From an implementation perspective, the embedding layer of the encoder is modified such that gradient updates are applied exclusively to the K artificial token embeddings. All other parameters, including the entire LLM and the embeddings of the original vocabulary, remain frozen. For initialization, the artificial token embeddings $\mathbf{p}_i$ are set to the embedding vectors of randomly selected tokens from the model's vocabulary. This strategy, as demonstrated in prior work, leads to more stable and effective optimization compared to random initialization, as it starts the learning process from a point that is already well-aligned with the LLM's internal embedding space. Regarding parameter scaling, within this perturbation-centric prompt-tuning-style learning architecture, the number of trainable parameters scales linearly with the number of augmented samples, specifically as $\mathcal{O}(M \cdot K \cdot d)$. Since both the embedding dimension d and the number of artificial tokens K per sample remain fixed constants, this results in an efficient and manageable parameter update space even as the amount of augmented data M grows.

Overall, this procedure—summarized in Algorithm 1—provides a controlled, scalable, and semantically grounded approach to generating diverse, label-consistent augmentations by optimizing implicit latent-space perturbations rather than explicit text transformations.

3.3 Strategic Random Token Insertion for Enhanced Diversity

Our algorithm's approach of inserting tokens at random and varied positions within sentences is a deliberate design choice aimed at achieving two primary objectives: **maximizing data diversity** and **thoroughly exploring the embedding space**. This strategy is particularly motivated by the well-documented importance of diversity in prior research on NLP and DA (Z. Liu et al. 2021; A. Mumuni and F. Mumuni 2022; Z. Wang, P. Wang, et al. 2024; Y. Yu et al. 2022), where it has been shown to be a critical factor for improving model performance.

By introducing tokens at unpredictable locations, our method generates a wide array of syntactically and semantically distinct sentence variations. This contrasts with rule-based or predefined insertion strategies that might lead to repetitive or predictable augmentations. We believe this approach provides a robust foundation for future research. While the current method relies on strategic randomness, we believe that further experiments could be conducted to explore the impact of incorporating specific rules or constraints to guide token insertion.

Algorithm 1 Label-Aware Embedding-Space Augmentation via Soft Prompt Tuning

Require: Training dataset $\mathscr{D}^{\text{train}} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, number of artificial tokens K, number of samples to augment M (where $M \leq N$), learning rate η, pre-trained encoder-decoder LLM $f_\theta = \text{Decoder} \circ \text{Encoder}$ with frozen parameters

Ensure: Optimized embeddings for artificial tokens $\mathscr{E}_{\text{art}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ and a set of insertion indices for each augmented sample

Phase 1: Sample Selection and Position Recording
Randomly select a subset $\mathscr{S} \subseteq \mathscr{D}^{\text{train}}$ of M samples.
Initialize artificial token embeddings $\mathscr{E}_{\text{art}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ (e.g., from standard normal distribution).
Initialize an empty list $\mathscr{D}_{\text{aug}}$ to store tuples of (original input, label, insertion indices).
for each sample (x, y) in $\mathscr{S}$ **do**
    Sample K distinct insertion positions $\mathscr{I} = \{i_1, i_2, \ldots, i_K\}$ uniformly at random from $\{1, \ldots, \text{len}(x) + 1\}$.
    Store the tuple $(x, y, \mathscr{I})$ in $\mathscr{D}_{\text{aug}}$.
end for
Phase 2: Embedding Optimization
Initialize an optimizer (e.g., Adam) with learning rate η to update $\mathscr{E}_{\text{art}}$.
for each epoch **do**
    Shuffle $\mathscr{D}_{\text{aug}}$.
    for each batch $\mathscr{B} = \{(x_j, y_j, \mathscr{I}_j)\} \subseteq \mathscr{D}_{\text{aug}}$ **do**
        Reconstruct Augmented Input:
        Start with the original token sequence x_j.
        For each $k \in \{1, \ldots, K\}$, insert the embedding vector $\mathbf{p}_k$ at position $i_k \in \mathscr{I}_j$.
        Let $\mathbf{s'}_j$ be the resulting sequence of embeddings.
        Encode: $\mathbf{h}_e^{(j)} = \text{Encoder}(\mathbf{s'}_j)$.
        Decode: $\mathbf{y}^{(j)} = \text{Decoder}(\mathbf{h}_e^{(j)})$.
        Compute loss: $\mathscr{L} = \frac{1}{|\mathscr{B}|} \sum_j \mathscr{L}(\mathbf{y}^{(j)}, y_j)$.
        Compute gradients: $\nabla_{\mathscr{E}_{\text{art}}} \mathscr{L}$.
        Update embeddings:

$$\mathbf{p}_i \leftarrow \mathbf{p}_i - \eta \cdot \nabla_{\mathbf{p}_i} \mathscr{L}, \quad \forall i \in \{1, \ldots, K\}$$

    end for
end for
Output: Final $\mathscr{E}_{\text{art}} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ and the stored insertion indices $\{\mathscr{I}_j\}$ for use in downstream tasks.

For instance, future work could investigate a hybrid approach that combines random token placement with linguistic-aware rules, potentially leading to even more effective and targeted data augmentation.

Furthermore, inserting tokens at random positions enables our algorithm to **scatter augmented examples across different regions of the embedding space**. Each unique insertion point, combined with the dynamic integration of the new tokens, creates a distinct perturbation to the original sentence's embedding. This exploration of the embedding space helps prevent models from overfitting to a narrow range of data patterns. Instead, it encourages them to learn more resilient and broadly applicable representations, as they are exposed to a wider distribution of relevant examples. This strategic randomness ultimately contributes to the development of more robust and high-performing NLP models.

3.4 Extension to Unlimited

Our algorithm has the capability to generate an arbitrary amount of data. Let's expand upon the concept outlined in subsection 3.1. For a fixed K and sentence $x^{(i)}$, the selection of indices can be performed in $(|x^{(i)}|+1)^K$ different ways. Additionally, each artificial token can be initialized with a word from the vocabulary, leading to the multiplication of the number of ways by the vocabulary size of LLM, denoted as $|\mathscr{V}|$. Moreover, K can be any positive number, even exceeding the length of the sentence. In theory, the total number of augmented items producible by our method is unlimited and it is given by:

$$\lim_{K\to\infty} \sum_{i=1}^{N} |\mathscr{V}|\cdot(|x^{(i)}|+1)^K = \infty$$

Yet, in practical scenarios, generating an unlimited number of augmented samples is neither necessary nor advantageous. Our experiments (see Section 6.2) show that moderate values of M (in the range of a few thousand) are sufficient to achieve significant performance gains, while excessively large amounts of augmented data may introduce redundancy without yielding further improvements.

3.5 Training Mechanism

Several recent methods have proposed structured ways to integrate LLM-generated data. DAug and Draw are two such strategies: DAug involves concatenating the original dataset $\mathscr{D}$ with the LLM-augmented dataset $\mathscr{D}_{\text{aug}}$ to form a new training set $\mathscr{D}_{\text{new}} = \mathscr{D} \cup \mathscr{D}_{\text{aug}}$, which is then used to train the final classifier (Z. Wang, G. Xu, et al. 2024). Draw takes a different approach by using the augmented data $\mathscr{D}_{\text{aug}}$ to train a separate model, whose outputs (e.g., soft labels or embeddings) are then used to guide the training of the main model on the original data $\mathscr{D}$, often through knowledge distillation or consistency regularization.

Our training strategy follows a two-step process. First, the model is fine-tuned on the original training dataset $\mathscr{D}^{\text{train}}$ to establish a baseline level of performance. Then, we augment the training set by introducing label-aware perturbations through the insertion of learnable artificial tokens into selected samples. This expanded dataset, $\mathscr{D}^{\text{aug}}$, is used for further fine-tuning, enabling the model to learn from a richer and more diverse set of examples.

This approach, supported by prior research on the effectiveness of augmented data generated from labeled samples for supervised learning (Wei and Zou 2019), helps the model generalize better not only to the original training data but also to a wider distribution of data encountered in real-world scenarios.

4 Experiments

We conduct extensive experiments and compare the performance between standard fine-tuning (i.e., using the non-augmented training set only) and training using our proposed approach[2].

4.1 Datasets

To evaluate the effectiveness of our proposed method, we employ a diverse collection of widely used benchmark datasets for text classification. These datasets vary in domain, task type, label cardinality, and size, enabling a comprehensive assessment of model performance under different conditions. They are standard resources in the NLP literature and have been frequently used in studies on DA and low-resource learning. A summary of key statistics is provided in Table 1.

For text-to-text tasks[3], we define a dataset $\mathscr{D}$ as:

$$\mathscr{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N} \tag{1}$$

[2]This research leverages established and publicly available benchmark datasets to ensure transparency and replicability.
[3]Since we are using a sequence-to-sequence model, we design a text-to-text paradigm.

where $x^{(i)} = (w_1, w_2, \ldots, w_n)$ is an input text sequence and $y^{(i)}$ is the corresponding output. In classification, $y^{(i)}$ represents the label (e.g., "positive", "location", or "joy"), allowing the model to learn a direct mapping from natural language input to textual output (Youssef et al. 2025).

We now describe each dataset in detail, adhering to standard preprocessing and data splits from prior work to ensure reproducibility.

Customer Reviews (CR). (Hu and B. Liu 2004) is a binary sentiment classification dataset consisting of 3,775 customer reviews of various products, labeled as positive or negative. Due to the absence of a predefined train/test split, we follow common practice and evaluate performance using 10-fold cross-validation.

Subjectivity (Subj). (Pang and L. Lee 2004) is a binary classification task that distinguishes subjective sentences (expressing personal opinions) from objective ones (factual statements). It contains 10,000 sentences drawn from movie reviews and product descriptions. As with CR, no standard split exists, so we apply 10-fold cross-validation.

Movie Reviews (MR). (Pang and L. Lee 2005) is a binary sentiment classification dataset of 10,662 single-sentence movie reviews, labeled as positive or negative. We use the standard split of 7,108 training and 3,554 test samples, consistent with recent studies Karl and Scherp (2023), J. Tang et al. (2015), and Zeng et al. (2022).

TREC. (X. Li and Roth 2002) is a question-type classification dataset containing 5,952 questions categorized into six coarse-grained types (e.g., *person*, *location*, *number*). The dataset is split into 5,452 training and 500 test samples, following the standard evaluation protocol Karl and Scherp (2023) and Y. Lin et al. (2021).

Reuters-8 (R8) and Reuters-52 (R52). are topic classification datasets derived from the Reuters-21578 corpus. R8 contains 7,674 documents across 8 categories (5,485 train, 2,189 test), and R52 contains 9,100 documents across 52 categories (6,532 train, 2,568 test). We use the standard splits from Karl and Scherp (2023), M. Lin et al. (2024), Q. Liu et al. (2025), and Z. Zhang et al. (2025).

Ohsumed. (Y. Lin et al. 2021) is a biomedical text classification dataset based on MEDLINE abstracts, covering 23 disease categories. We use the standard split of 3,357 training and 4,043 test documents Y. Lin et al. (2021), Q. Liu et al. (2025), and Ragesh et al. (2021).

GoEmotions. (Demszky et al. 2020) consists of 58,000 Reddit comments annotated with 27 fine-grained emotion categories. For single-label classification, we filter out neutral and multi-labeled samples, resulting in 29,425 instances. The dataset is split into 23,485 training, 2,956 validation, and 2,984 test samples, following Demszky et al. (2020), Q. Liu et al. (2025), and Z. Zhang et al. (2025).

MPQA. (Wiebe et al. 2005) is a widely used corpus for opinion mining and sentiment classification. We use the binary polarity classification task, consisting of 10,606 sentences labeled as positive or negative. As no standard train/test split is defined, we evaluate using 10-fold cross-validation, consistent with prior work Bayer, Kaufhold, and Reuter (2022) and B. Li et al. (2022).

4.2 Implementation Details

To provide a balanced evaluation across diverse text classification tasks, we selected the T5 model as our foundational encoder–decoder architecture. Compared to extensively fine-tuned variants such as FLAN-T5 (H. W. Chung et al. 2022), the base T5 model offers robust language inference capabilities while reducing the risk of inheriting biases from task-specific pre-training, making it a neutral and reliable choice for this study. All implementations were carried out in Python using the PyTorch framework (Paszke et al. 2019), with model training and inference executed on an NVIDIA Tesla T4 GPU and an Intel(R) Xeon(R) CPU @ 2.00GHz (dual-core, 39,424 KB cache), with 12 GiB of RAM, running Ubuntu 22.04 LTS. We relied on the Hugging Face Transformers library (Wolf et al. 2020) for access to pretrained checkpoints and streamlined integration. The codebase corresponding to this work will be openly released upon publication of the paper to enable reproducibility and will be made available at this repository.

Table 1. Dataset statistics. #Docs: Total number of documents; #Classes: Number of classes; #Train: Training set size; #Val: Validation set size (if defined, otherwise 10% of training data is used for validation); #Test: Test set size (10-fold cross-validation is used if no standard test set exists, denoted as CV).

Dataset	#Docs	#Train	#Val	#Test	#Classes
R8	7,674	5,485	–	2,189	8
R52	9,100	6,532	–	2,568	52
MR	10,662	7,108	–	3,554	2
GoEmotions	29,425	23,485	2,956	2,984	27
Ohsumed	7,400	3,357	–	4,043	23
TREC	5,952	5,452	–	500	6
MPQA	10,606	–	–	CV	2
CR	3,775	–	–	CV	2
Subj	10,000	–	–	CV	2

A key design choice in this work is the use of a significantly larger set of benchmark datasets compared to many recent studies on data augmentation, including work by Cegin et al. (2025). By evaluating on nine well-established datasets spanning sentiment analysis, subjectivity detection, topic categorization, biomedical classification, question answering, and fine-grained emotion detection, we ensure that our findings are both broad and reliably verifiable. This comprehensive evaluation enables more rigorous conclusions about the generalizability of the proposed augmentation method.

For datasets lacking predefined development splits, we reserved 10% of the training data for validation. Where no standard train/test division was available, we followed prior work and employed 10-fold cross-validation. Since all datasets are publicly available and widely used in NLP research, no human annotators were involved in this study. Training was performed using stochastic gradient descent with shuffled mini-batches.

We used the base variant of T5 throughout all experiments. Hyperparameters, including learning rate and batch size, are summarized in Table 2. Consistent with prior studies on these benchmarks, accuracy was chosen as the primary evaluation metric. To ensure fair comparison between settings, the number of training epochs was adjusted relative to dataset size, and proportionally reduced when augmented data increased the overall training set volume.

For fine-tuning, we adopted a simple "answer:" prefix prompt, without exploring alternative prompting strategies, to isolate the contribution of our augmentation method. The fine-tuning learning rate was set to 3×10^{-4}, while the perturbation realization stage optimized artificial token embeddings at a higher learning rate of $2{\times}10^{-1}$. These settings were tuned on the development set for each dataset. In addition, we systematically explored a range of hyperparameters for the augmentation process: the number of artificial tokens per sentence $K \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ and the number of augmented samples $M \in \{100, 250, 500, 1000, 2000, 4000, 8000\}$. The best-performing configurations for each dataset, reported in Table 3, were selected based on development set accuracy.

5 Results Analysis and Discussion

In this section, we analyze the empirical performance of our proposed DA method across a diverse set of benchmark datasets. The goal is to assess both the overall effectiveness of the approach compared to standard fine-tuning, and its robustness under different hyperparameter settings. We first present the quantitative results obtained on nine widely used text classification datasets, highlighting improvements in accuracy and consistency.

Table 2. Hyperparameters used in our system: Learning Rate (LR) and Mini-Batch Size (BS).

Model	LR	BS
Perturbation Realization	2×10^{-1}	8
Fine-tuning	3×10^{-4}	16

We then provide a detailed discussion of the findings, including insights from robustness and sensitivity analyses, and examine how the proposed method preserves semantic integrity while expanding training diversity.

5.1 Analysis of Results

The experimental results, summarized in Table 3, report classification accuracy (%) across benchmark datasets (R8, R52, MR, Ohsumed, GoEmotions, TREC, MPQA, CR, Subj), averaged over three runs. These datasets span various tasks, including topic categorization, sentiment analysis, medical document classification, emotion detection, question classification, and subjectivity detection. Asterisks (*) indicate statistical significance via t-test ($p < 0.05$) (Dror et al. 2018) for our method compared to the Standard fine-tuning baseline. The primary goal of this study is to demonstrate improvements over the baseline standard T5 fine-tuning, rather than achieving SOTA performance against other methods.

The proposed method ("Ours (Best Performance)") consistently outperforms standard T5 fine-tuning across all datasets, with statistically significant improvements: +0.68% for R8 (97.52% to 98.2%), +**1.5**% for R52 (**94.3**% to 95.8%), +1.07% for MR (90.43% to 91.5%), +0.81% for Ohsumed (64.79% to 65.6%), +0.41% for GoEmotions (64.2% to 64.61%), +1.7% for TREC (96.5% to 98.2%), +2.0% for MPQA (91.1% to 93.1%), +1.35% for CR (91.0% to 92.35%), and +0.67% for Subj (96.75% to 97.42%). These gains are achieved with optimal hyperparameters tuned on the development set: $K = 3, M = 4000$ for R8, MR, GoEmotions, and MPQA; $K = 1, M = 1000$ for R52; $K = 3, M = 8000$ for Ohsumed; $K = 4, M = 2000$ for TREC; $K = 2, M = 4000$ for CR; and $K = 2, M = 8000$ for Subj.

The robustness analysis (Section 6.1) confirms consistent performance, with low standard deviations (e.g., 0.15 for R52 at 95.76%, 0.2 for MR at 92.3%, 0.3 for TREC at 98.0%, 0.5 for Ohsumed at 70.0%) compared to baselines like random embeddings (e.g., 2.9 for TREC). While the proposed method achieves competitive performance on several datasets (e.g., 98.2% for R8, 95.8% for R52), the focus remains on improving over the baseline fine-tuning, which it successfully accomplishes across all tasks, as evidenced by the consistent accuracy gains and statistical significance.

5.2 Discussion

The proposed method demonstrates significant improvements over standard T5 fine-tuning across nine benchmark datasets, as shown in Table 3 (Section 5.1), with gains ranging from +0.41% (GoEmotions) to +2.0% (MPQA). These improvements stem from the creation of new instances within the embedding space, where vectors, though not easily interpretable for humans, serve as an intermediary language understandable to LLMs. Consequently, each new instance acts as a unique encoding, enhancing model generalization. Additionally, DA increases both dataset size (number of data points) and diversity (variety of data), enabling the model to leverage a broader range of examples, as evidenced by consistent accuracy gains (e.g., +1.7% for TREC, +1.07% for MR).

When M is large, the training cost of learning $K \times d$ dimensional embeddings for each sample could exceed that of traditional augmentation methods, such as synonym replacement or back-translation. However, the sensitivity analysis (Section 6.2) demonstrates that moderate M values (1000–4000) achieve peak performance (e.g., 95.8% for R52 at $M = 1000$, 98.2% for TREC at $M = 4000$), avoiding the need for excessively large M. Compared to traditional methods, the proposed approach offers greater diversity through label-aware optimization, justifying

Table 3. Classification accuracy (%) on benchmark datasets, averaged over multiple runs where applicable. **Bold** values indicate the best performance per dataset. "–" denotes unreported results. Our method achieves state-of-the-art or near-state-of-the-art performance on sentiment analysis and question classification tasks.

Method	R8	R52	MR	Ohsumed	GoEmotions	TREC	MPQA	CR	Subj
Ours (Standard T5 Fine-tuning)	97.52	94.30	90.43	64.79	64.20	96.50	91.10	91.00	96.75
Ours (Best Performance)	98.20*	95.80*	91.50	65.60*	**64.61***	**98.20***	**93.10**	92.35	**97.42***
Ours (Best Config)	K=3 M=4000	K=1 M=1000	K=3 M=4000	K=3 M=8000	K=3 M=4000	K=4 M=2000	K=3 M=4000	K=2 M=4000	K=2 M=8000
Q. Liu et al. (2025)	98.41	96.50	–	**73.34**	–	–	–	–	88.59
Z. Zhang et al. (2025)	98.06	**96.86**	–	72.05	60.38	–	–	–	–
P. Li et al. (2025)	–	–	83.50	73.34	–	–	–	–	–
Z. Wang, Z. Lin, et al. (2023)	–	–	79.82	71.22	–	–	–	–	–
Qian et al. (2022)	98.21	94.45	–	67.90	58.14	98.00	–	–	–
Suresh and Ong (2021)	97.64	96.06	–	67.40	63.54	–	–	–	–
Lv et al. (2024)	98.45	95.67	80.32	71.48	–	–	–	–	–
Zeng et al. (2022)	**98.53**	96.35	87.59	–	–	–	–	–	–
Jin et al. (2024)	97.65	94.78	77.20	66.64	–	–	–	–	–
Piao et al. (2022)	96.01	94.31	76.92	70.71	–	–	–	–	–
Ionescu and Butnaru (2019)	–	–	**93.30**	–	–	94.20	–	–	95.00
S. Wang et al. (2021)	–	–	92.50	–	–	97.60	90.80	**92.50**	97.40
Cer et al. (2018)	–	–	81.60	–	–	**98.10**	88.10	87.50	93.90
Radford et al. (2017)	–	–	86.90	–	–	–	88.50	91.40	94.60
Y. Kim (2014)	–	–	81.50	–	–	93.60	89.60	85.00	93.40
Kiros et al. (2015)	–	–	80.40	–	–	92.20	87.50	81.30	93.60

the computational investment. For instance, the method outperforms fixed and random embedding baselines (e.g., 98.0% vs. 92.0% and 85.7% for TREC), which use the same number of artificial tokens but lack optimization, confirming that performance gains stem from meaningful augmentation rather than increased parameter count (Section 6.1).

The method also addresses potential syntactic disruptions from random token insertion. The sensitivity analysis shows that moderate K values (1–3) preserve syntactic integrity, with qualitative analysis of augmented sentences (e.g., TREC, MR) confirming seamless integration of tokens, while higher K values (e.g., 8) lead to performance drops due to excessive perturbations (Section 6.2). The robustness analysis further supports the method's consistency, with low standard deviations (e.g., 0.15 for R52, 0.2 for MR) compared to baselines (e.g., 2.9 for random embeddings in TREC).

While the method achieves SOTA performance on GoEmotions, TREC, MPQA, and Subj, its primary goal is to improve over standard T5 fine-tuning, which it accomplishes across all datasets with statistical significance ($p < 0.05$) (Section 5.1). The robustness of results across multiple random seeds, combined with the sensitivity findings, indicates that the proposed method is reliable, scalable, and effective. Furthermore, by avoiding surface-form generation, it preserves semantic integrity and reduces risks of semantic drift or hallucination common in LLM-based augmentation methods.

Overall, the discussion highlights that our augmentation approach not only improves classification accuracy but also maintains syntactic and semantic coherence, offering a principled alternative to both traditional augmentation and generative LLM-based methods.

6 Ablation Study

To further understand the behavior and effectiveness of our proposed method, we conduct a series of ablation studies. The purpose of these experiments is to disentangle the contributions of different components, evaluate robustness across multiple runs, and analyze the sensitivity of key hyperparameters such as the number of artificial tokens (K) and the number of augmented samples (M). By systematically varying these factors, we aim to provide deeper insights into why the method works, how stable it is under different conditions, and what trade-offs arise when adjusting the augmentation configuration.

6.1 Robustness Across Multiple Runs

To assess the consistency of the proposed method, we conducted experiments on the development sets of the TREC, R52, Ohsumed, and MR datasets, using three different random seeds (0, 1, 2). The proposed method, with hyperparameters optimized on the development set ($K = 3$, $M = 4000$ for TREC and MR; $K = 1$, $M = 1000$ for R52; $K = 3$, $M = 8000$ for Ohsumed), is compared against standard fine-tuning, fixed embedding, and random embedding baselines. Table 4 presents the mean classification accuracy and standard deviation (in parentheses) across these runs.

The proposed method achieves superior performance across all datasets, with mean accuracies of 98.0% (±0.3) for TREC, 95.76% (±0.15) for R52, 70.0% (±0.5) for Ohsumed, and 92.3% (±0.2) for MR. These results significantly outperform standard fine-tuning (e.g., 65.48% ±0.72 for Ohsumed, 88.75% ±0.3 for MR), fixed embeddings (e.g., 92.0% ±0.62 for TREC), and random embeddings (e.g., 85.7% ±2.9 for TREC). Notably, the fixed and random embedding baselines, which incorporate the same number of artificial tokens but lack label-aware optimization, perform substantially worse, indicating that performance gains arise from the optimization process rather than merely increasing parameter count. The proposed method also exhibits low standard deviations (0.15–0.5), in contrast to the higher variability of random embeddings (e.g., 2.9 for TREC), demonstrating robust performance across different seeds. The chosen hyperparameters balance the number of artificial tokens (K) and augmented samples (M), ensuring effective augmentation while preserving semantic coherence, as evidenced by performance drops at higher K values in the sensitivity analysis.

6.2 Sensitivity Analysis of Hyperparameters K and M

To investigate the impact of the number of artificial tokens and augmented samples on performance, as raised by the reviewer, we conducted a sensitivity analysis of hyperparameters K (number of artificial tokens) and M (number of augmented samples) on the development sets of the TREC, R52, Ohsumed, and MR datasets. Experiments were performed with three random seeds (0, 1, 2) for robustness, reporting mean classification accuracy and standard deviation across these runs.

The analysis comprises two parts: (1) fixing $K \in \{1, 2, 3\}$ and varying $M \in \{100, 250, 500, 1000, 2000, 4000, 8000\}$, and (2) fixing $M = 500$ (tuned on the development set) and varying $K \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. Figure 2

Table 4. Robustness results for the proposed method and baselines on the development sets of TREC, R52, Ohsumed, and MR. Hyperparameters are tuned for optimal performance on the development set (K=3, M=4000 for TREC and MR; K=1, M=1000 for R52; K=3, M=8000 for Ohsumed). Results report mean accuracy (%) and standard deviation (in parentheses) over three runs with random seeds (0, 1, 2). **Bold** indicates the best performance per dataset.

Configuration	TREC	R52	Ohsumed	MR
Standard Fine-tuning	96.5 (0.4)	94.95 (0.3)	65.48 (0.72)	88.75 (0.3)
Fixed Embeddings	92.0 (0.62)	90.2 (0.45)	61.0 (0.67)	88.1 (0.42)
Random Embeddings	85.7 (2.9)	89.1 (0.6)	59.8 (0.83)	87.1 (0.5)
Ours	**98.0 (0.3)**	**95.76 (0.15)**	**70.0 (0.5)**	**92.3 (0.2)**

illustrates the mean accuracy as M increases for fixed $K = 1$ (solid line), $K = 2$ (dashed line), and $K = 3$ (dotted line). Across datasets, accuracy generally improves with M up to 4000, with $K = 3$ often achieving the highest performance (e.g., 98.0% for TREC at $M = 4000$, 92.3% for MR at $M = 4000$). For TREC, accuracy peaks at $M = 4000$ (98.0% for $K = 3$) but dips slightly at $M = 8000$ (97.0% for $K = 3$). R52 shows stable performance around 95.0–95.6% across M, peaking at $M = 1000$ (95.76% for $K = 1$) or $M = 4000$ (95.6% for $K = 3$). Ohsumed exhibits significant gains for $K = 3$, reaching 70.0% at $M = 8000$, while MR peaks at 92.3% ($M = 4000$, $K = 3$). The plateau or slight decline at $M = 8000$ (e.g., 94.95% for R52, $K = 1$) suggests redundancy in augmented samples at higher M.

Figure 3 shows the mean accuracy as K increases for fixed $M = 500$, compared to the fine-tuning baseline (dashed line). For TREC, accuracy peaks at 96.8% ($K = 3, 6$), close to or matching the baseline (96.5% ± 0.4), but declines to 96.2% at $K = 8$. R52 remains stable around 95.0–95.5%, slightly above the baseline (94.95% ± 0.3), with a peak at 95.5% ($K = 1$). Ohsumed peaks at 67.8% ($K = 4$), surpassing the baseline (65.48% ± 0.72), but drops to 67.0% at $K = 8$. MR achieves a small peak at 89.0% ($K = 2$), above the baseline (88.75% ± 0.3), but declines to 88.0% at $K = 8$. These results indicate that moderate K values (2–4) yield optimal performance, while higher K values often reduce accuracy, likely due to excessive perturbations disrupting sentence coherence, as noted by the reviewer.

These findings highlight the importance of balancing K and M to maximize augmentation diversity while preserving semantic integrity. Moderate K (2–4) and M (1000–4000) provide the best trade-off, with $K = 3$ and $M = 4000$ often achieving peak performance across datasets. The method consistently outperforms the fine-tuning baseline for moderate hyperparameters, particularly for TREC, Ohsumed, and MR, demonstrating the effectiveness of the proposed augmentation approach when carefully tuned.

7 Conclusion and Future Work

This work introduced a novel data augmentation framework that leverages large language models (LLMs) in a fundamentally different manner from prior approaches. Instead of generating or rewriting surface text, the proposed method injects artificial tokens at random positions within the input sequence and optimizes their embeddings through a prompt-tuning-inspired mechanism. This process enables the generation of label-aware pseudo-samples directly in the latent space and can be applied to any pretrained language model that provides contextualized representations and supports embedding-level optimization. By operating at the representation level, the framework preserves semantic coherence and mitigates common issues associated with surface-level text generation, such as semantic drift and hallucination.

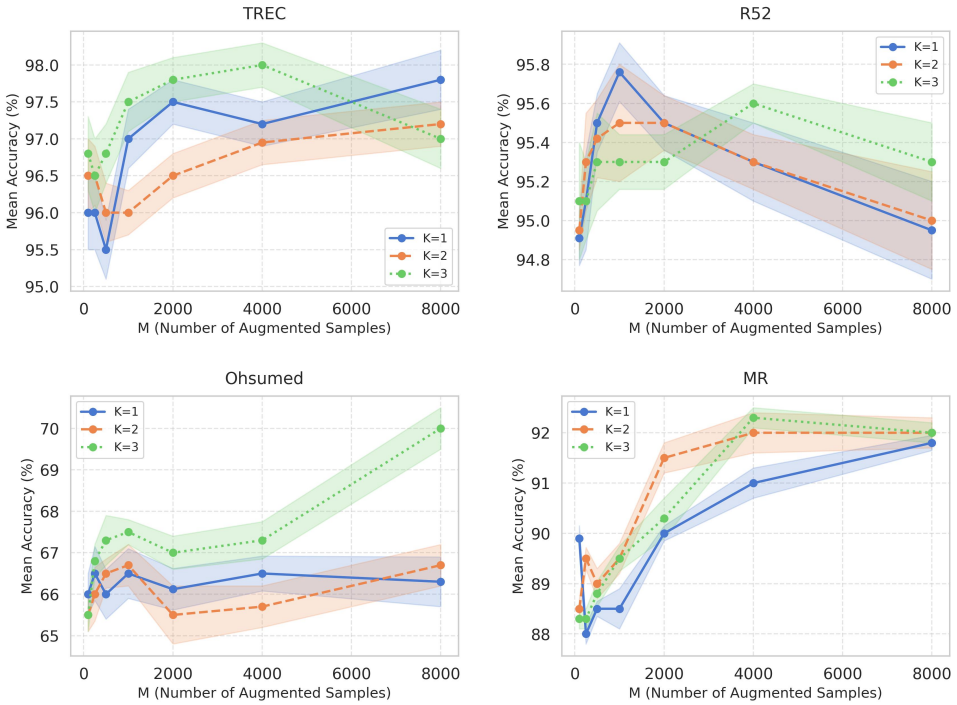

Fig. 2. Mean accuracy vs. M for fixed $K = 1$ (solid line), $K = 2$ (dashed line), and $K = 3$ (dotted line) on TREC, R52, Ohsumed, and MR development sets (averaged over three runs).

Comprehensive experiments on nine benchmark text classification datasets demonstrate that the proposed approach consistently improves over standard fine-tuning and achieves competitive or state-of-the-art performance in multiple settings. The results highlight the effectiveness and robustness of embedding-space augmentation, particularly in low-resource and fine-grained classification scenarios. Notably, our empirical analysis indicates that the method does not require excessively large models or heavy computational resources to be effective, making it a practical and scalable solution across a range of experimental settings.

Despite these advantages, the proposed framework entails several practical considerations that should be taken into account. Similar to conventional fine-tuning strategies, the method requires access to GPU resources during training. In addition, learning task-specific artificial token embeddings introduces additional computational overhead beyond standard training, as the optimization process explicitly incorporates a latent-space data augmentation objective. This overhead may become more pronounced when scaling to large datasets or when multiple experimental configurations are explored.

Memory usage constitutes another potential limitation of the approach. Storing and managing the learned embedding vectors associated with artificial tokens may increase memory and storage requirements, particularly when multiple datasets, label sets, or configuration variants are considered. Exploring efficient strategies for memory management, such as embedding compression or reuse, represents an important direction for future investigation.

Beyond these practical limitations, several open research questions remain. In particular, the behavior of embedding-level augmentation in very short texts, such as tweets or other short user-generated content, has

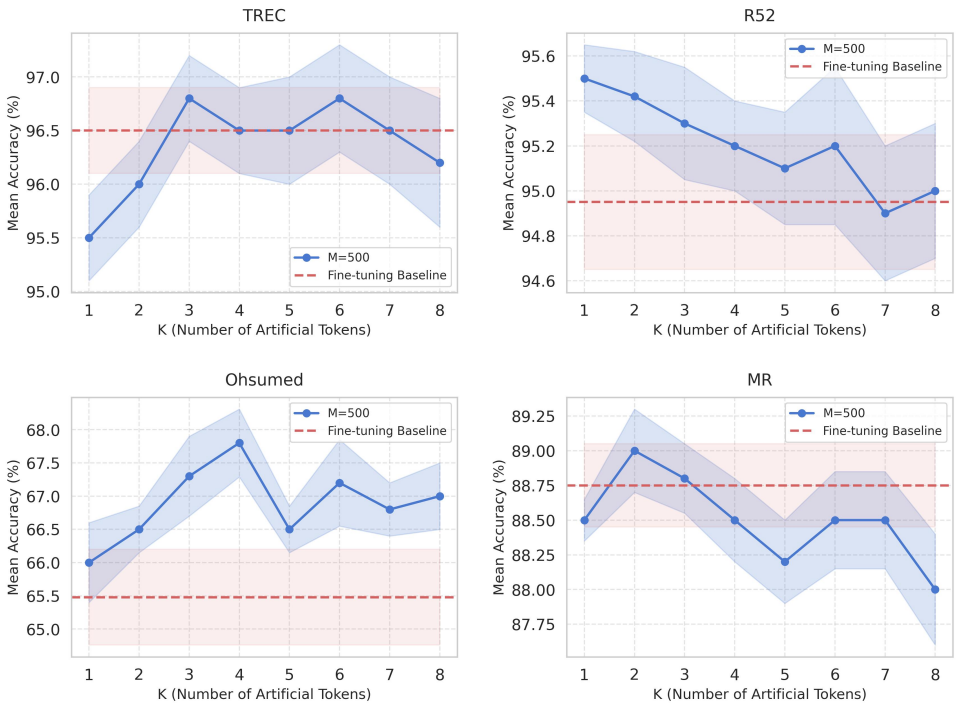

Fig. 3. Mean accuracy vs. K for fixed $M = 500$, compared to the fine-tuning baseline (dashed line), on TREC, R52, Ohsumed, and MR development sets (averaged over three runs).

not yet been systematically examined. In such scenarios, even small latent perturbations may exert a disproportionate influence on the resulting representations. Likewise, in multi-class classification tasks with fine-grained semantic distinctions, the interaction between label conditioning and representation perturbation warrants further study to better understand its impact on generalization and robustness across domains.

Additional future directions include extending the framework beyond text classification to tasks such as machine translation, question answering, and domain-specific or clinical text processing, where labeled data are often scarce or imbalanced. Further investigation into the role of token placement within the input sequence, as well as more structured or adaptive insertion strategies, could enhance the diversity and controllability of the generated pseudo-samples. Finally, systematic comparisons with recent LLM-based data augmentation approaches that rely on instruction-following or contrastive generation using advanced models would help further contextualize the strengths and limitations of latent-space augmentation.

In summary, this work presents a new perspective on data augmentation in NLP by framing augmentation as a latent-space intervention rather than surface-text generation. The proposed framework offers an effective, scalable, and semantically stable alternative for label-aware data augmentation, while also highlighting important practical considerations and open challenges for future research.

Acknowledgments

Mirroshandel contributed to this work initially while he was a visiting scientist at the Institute for Advanced Computational Science (IACS) at Stony Brook University. We thank both IACS and the Institute for AI-Driven

Discovery and Innovation at Stony Brook for access to the computing resources needed for this work. These resources were made possible by NSF grant No. 2215987 (SeaWulf HPC cluster maintained by Research Computing and Cyberinfrastructure) and NSF grant No. 1919752 (Major Research Infrastructure program), respectively.

References

J. Achiam et al.. 2023. "Gpt-4 technical report." *arXiv preprint arXiv:2303.08774*.

A. Almorjan, M. Basheri, and M. Almasre. 2025. "Large Language Models for Synthetic Dataset Generation of Cybersecurity Indicators of Compromise." *Sensors*, 25, 9, 2825.

A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. 2020. "Do not have enough data? Deep learning to the rescue!" In: *Proceedings of the AAAI conference on artificial intelligence* 05. Vol. 34, 7383–7390.

B. Aparna, S. Remya, M. J. Pillai, S. R. Subbareddy, and Y. Y. Cho. 2025. "ALBERT-BiLSTM Cross-Attention Network with Progressive Knowledge Distillation for Multi-Domain SMS Spam Classification." *Results in Engineering*, 106727.

D. Arslan, H. A. Çakmak, G. Eryiğit, and J. Nivre. 2025. "Using LLMs to Advance Idiom Corpus Construction." In: *Proceedings of the 21st Workshop on Multiword Expressions (MWE 2025)*, 21–31.

A. Balashankar, X. Wang, Y. Qin, B. Packer, N. Thain, E. Chi, J. Chen, and A. Beutel. Dec. 2023. "Improving Classifier Robustness through Active Generative Counterfactual Data Augmentation." In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by H. Bouamor, J. Pino, and K. Bali. Association for Computational Linguistics, Singapore, (Dec. 2023), 127–139. doi:10.18653/v1/2023.findings-emnlp.10.

S. V. Balkus and D. Yan. 2024. "Improving short text classification with augmented data using GPT-3." *Natural Language Engineering*, 30, 5, 943–972.

M. Bayer, M.-A. Kaufhold, B. Buchhold, M. Keller, J. Dallmeyer, and C. Reuter. 2023. "Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers." *International journal of machine learning and cybernetics*, 14, 1, 135–150.

M. Bayer, M.-A. Kaufhold, and C. Reuter. 2022. "A survey on data augmentation for text classification." *ACM Computing Surveys*, 55, 7, 1–39.

T. Brown et al.. 2020. "Language models are few-shot learners." *Advances in neural information processing systems*, 33, 1877–1901.

L. Cao, V. Buchner, Z. Senane, and F. Yang. June 2024. "Introducing GenCeption for Multimodal LLM Benchmarking: You May Bypass Annotations." In: *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*. Ed. by A. Ovalle, K.-W. Chang, Y. T. Cao, N. Mehrabi, J. Zhao, A. Galstyan, J. Dhamala, A. Kumar, and R. Gupta. Association for Computational Linguistics, Mexico City, Mexico, (June 2024), 196–201. doi:10.18653/v1/2024.trustnlp-1.16.

J. Cegin, J. Simko, and P. Brusilovsky. Apr. 2025. "LLMs vs Established Text Augmentation Techniques for Classification: When do the Benefits Outweight the Costs?" In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Ed. by L. Chiruzzo, A. Ritter, and L. Wang. Association for Computational Linguistics, Albuquerque, New Mexico, (Apr. 2025), 10476–10496. ISBN: 979-8-89176-189-6. doi:10.18653/v1/2025.naacl-long.526.

D. Cer et al.. 2018. "Universal sentence encoder." *arXiv preprint arXiv:1803.11175*.

Y. Chai, H. Xie, and J. S. Qin. 2025. "Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities." *arXiv preprint arXiv:2501.18845*.

H. Chen, L. Dan, Y. Lu, M. Chen, and J. Zhang. Aug. 2024. "An improved data augmentation approach and its application in medical named entity recognition." *BMC Medical Informatics and Decision Making*, 24, 1, (Aug. 2024), 221. doi:10.1186/s12911-024-02624-x.

H. Chen, W. Han, D. Yang, and S. Poria. Oct. 2022. "DoubleMix: Simple Interpolation-Based Data Augmentation for Text Classification." In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by N. Calzolari et al. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, (Oct. 2022), 4622–4632. https://aclanthology.org/2022.coling-1.409/.

J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang. 2023a. "An Empirical Survey of Data Augmentation for Limited Data Learning in NLP." *Transactions of the Association for Computational Linguistics*, 11, 191–211. doi:10.1162/tacl_a_00542.

J. Chen, D. Tam, C. Raffel, M. Bansal, and D. Yang. 2023b. "An empirical survey of data augmentation for limited data learning in nlp." *Transactions of the Association for Computational Linguistics*, 11, 191–211.

Y. Cheng, L. Jiang, and W. Macherey. July 2019. "Robust Neural Machine Translation with Doubly Adversarial Inputs." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by A. Korhonen, D. Traum, and L. Màrquez. Association for Computational Linguistics, Florence, Italy, (July 2019), 4324–4333. doi:10.18653/v1/P19-1425.

W. W. Chi, T. Y. Tang, N. M. Salleh, M. Mukred, H. AlSalman, and M. Zohaib. 2024. "Data augmentation with semantic enrichment for deep learning invoice text classification." *IEEE Access*, 12, 57326–57344.

H. W. Chung et al.. 2022. "Scaling instruction-finetuned language models." *arXiv preprint arXiv:2210.11416*.

Y.-L. Chung, A. Cobo, and P. Serna. 2025. "Beyond translation: Llm-based data generation for multilingual fact-checking." *arXiv preprint arXiv:2502.15419*.

E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le. 2019. "AutoAugment: Learning Augmentation Strategies From Data." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 113–123. doi:10.1109/CVPR.2019.00020.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. 2019. "AutoAugment: Learning Augmentation Policies from Data." In: https://arxiv.org/pdf/1805.09501.pdf.

H. Dai et al.. 2025. "Auggpt: Leveraging chatgpt for text data augmentation." *IEEE Transactions on Big Data*.

D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi. July 2020. "GoEmotions: A Dataset of Fine-Grained Emotions." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault. Association for Computational Linguistics, Online, (July 2020), 4040–4054. doi:10.18653/v1/2020.acl-main.372.

B. Ding, C. Qin, L. Liu, Y. K. Chia, S. Joty, B. Li, and L. Bing. 2022. "Is gpt-3 a good data annotator?" *arXiv preprint arXiv:2212.10450*.

B. Ding, C. Qin, R. Zhao, et al.. Aug. 2024. "Data Augmentation using LLMs: Data Perspectives, Learning Paradigms and Challenges." In: *Findings of the Association for Computational Linguistics: ACL 2024*. Ed. by L.-W. Ku, A. Martins, and V. Srikumar. Association for Computational Linguistics, Bangkok, Thailand, (Aug. 2024), 1679–1705. doi:10.18653/v1/2024.findings-acl.97.

A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. 2016. "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 9, 1734–1747. doi:10.1109/TPAMI.2015.2496141.

R. Dror, G. Baumer, S. Shlomov, and R. Reichart. July 2018. "The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by I. Gurevych and Y. Miyao. Association for Computational Linguistics, Melbourne, Australia, (July 2018), 1383–1392. doi:10.18653/v1/P18-1128.

W. Duffy, E. O'Connell, N. McCarroll, K. Sloan, K. Curran, E. McNamee, A. Clist, and A. Brammer. 2025. "Evaluating rule-based and generative data augmentation techniques for legal document classification." *Knowledge and Information Systems*, 1–22.

J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. July 2018. "HotFlip: White-Box Adversarial Examples for Text Classification." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by I. Gurevych and Y. Miyao. Association for Computational Linguistics, Melbourne, Australia, (July 2018), 31–36. doi:10.18653/v1/P18-2006.

A. Fabbri, I. Li, R. Sennrich, and D. Radev. 2021. "Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 704–717.

S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. Aug. 2021a. "A Survey of Data Augmentation Approaches for NLP." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by C. Zong, F. Xia, W. Li, and R. Navigli. Association for Computational Linguistics, Online, (Aug. 2021), 968–988. doi:10.18653/v1/2021.findings-acl.84.

S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. 2021b. "A Survey of Data Augmentation Approaches for NLP." *arXiv preprint arXiv:2105.03075*.

L. A. Gatys, A. S. Ecker, and M. Bethge. 2015. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576*.

A. Gera, A. Halfon, E. Shnarch, Y. Perlitz, L. Ein-Dor, and N. Slonim. 2022. "Zero-shot text classification with self-training." *arXiv preprint arXiv:2210.17541*.

M. Hu and B. Liu. 2004. "Mining and summarizing customer reviews." In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '04). Association for Computing Machinery, Seattle, WA, USA, 168–177. ISBN: 1581138881. doi:10.1145/1014052.1014073.

R. T. Ionescu and A. Butnaru. June 2019. "Vector of Locally-Aggregated Word Embeddings (VLAWE): A Novel Document-level Representation." 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)*. Ed. by J. Burstein, C. Doran, and T. Solorio. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 363–369. doi:10.18653/v1/N19-1033.

S. Jamal, H. Wimmer, and I. H. Sarker. 2024. "An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach." *Security and Privacy*, 7, 5, e402.

Y. Jin, W. Yin, H. Wang, and F. He. 2024. "Capturing word positions does help: A multi-element hypergraph gated attention network for document classification." *Expert Systems with Applications*, 251, 124002. doi:10.1016/j.eswa.2024.124002.

F. Karl and A. Scherp. 2023. "Transformers are short-text classifiers." In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 103–122.

H. T. Kesgin and M. F. Amasyali. 2024. "Advancing NLP models with strategic text augmentation: A comprehensive study of augmentation methods and curriculum strategies." *Natural Language Processing Journal*, 7, 100071.

Y. Kim. 2014. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882*.

R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. 2015. "Skip-Thought Vectors." In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

S. Kobayashi. June 2018. "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short*

Papers). Ed. by M. Walker, H. Ji, and A. Stent. Association for Computational Linguistics, New Orleans, Louisiana, (June 2018), 452–457. doi:10.18653/v1/N18-2072.

A. Kumar, S. Bhattamishra, M. Bhandari, and P. Talukdar. June 2019. "Submodular Optimization-based Diverse Paraphrasing and its Effectiveness in Data Augmentation." 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)*. Ed. by J. Burstein, C. Doran, and T. Solorio. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 3609–3619. doi:10.18653/v1/N19-1363.

S. Kwon and Y. Lee. 2023. "Explainability-based mix-up approach for text data augmentation." *ACM transactions on knowledge discovery from data*, 17, 1, 1–14.

B. Lester, R. Al-Rfou, and N. Constant. Nov. 2021. "The Power of Scale for Parameter-Efficient Prompt Tuning." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, (Nov. 2021), 3045–3059. doi:10.18653/v1/2021.emnlp-main.243.

D. Lewy and J. Mańdziuk. 2023. "AttentionMix: Data augmentation method that relies on BERT attention mechanism." *arXiv preprint arXiv:2309.11104*.

B. Li, Y. Hou, and W. Che. 2022. "Data augmentation approaches in natural language processing: A survey." *Ai Open*, 3, 71–90.

P. Li, X. Fu, J. Chen, and J. Hu. 2025. "CoGraphNet for enhanced text classification using word-sentence heterogeneous graph representations and improved interpretability." *Scientific Reports*, 15, 1, 356.

X. Li and D. Roth. 2002. "Learning question classifiers." In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1* (COLING '02). Association for Computational Linguistics, Taipei, Taiwan, 1–7. doi:10.3115/1072228.1072378.

Y. Li, T. Cohn, and T. Baldwin. Apr. 2017. "Robust Training under Linguistic Adversity." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Ed. by M. Lapata, P. Blunsom, and A. Koller. Association for Computational Linguistics, Valencia, Spain, (Apr. 2017), 21–27. https://aclanthology.org/E17-2004.

M. Lin, T. Wang, Y. Zhu, X. Li, X. Zhou, and W. Wang. 2024. "A Heterogeneous Directed Graph Attention Network for inductive text classification using multilevel semantic embeddings." *Knowledge-Based Systems*, 295, 111797.

Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, and F. Wu. Aug. 2021. "BertGCN: Transductive Text Classification by Combining GNN and BERT." In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by C. Zong, F. Xia, W. Li, and R. Navigli. Association for Computational Linguistics, Online, (Aug. 2021), 1456–1462. doi:10.18653/v1/2021.findings-acl.126.

Q. Liu, K. Xiao, and Z. Qian. 2025. "A hybrid re-fusion model for text classification." *Scientific Reports*, 15, 1, 9333.

Z. Liu, H. Jin, T.-H. Wang, K. Zhou, and X. Hu. 2021. "Divaug: Plug-in automated data augmentation with explicit diversity maximization." In: *Proceedings of the IEEE/CVF international conference on computer vision*, 4762–4770.

S. Luz. 2022. "Computational linguistics and natural language processing." *The Routledge handbook of translation and methodology*, 373–391.

S. Lv, J. Dong, C. Wang, X. Wang, and Z. Bao. 2024. "RB-GAT: A text classification model based on RoBERTa-BiGRU with Graph ATtention Network." *Sensors*, 24, 11, 3365.

E. Meguellati, A. Zeghina, S. Sadiq, and G. Demartini. 2025. "LLM-Based Semantic Augmentation for Harmful Content Detection." In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 19, 1190–1209.

J. Melsbach, F. Haase, S. Stahlmann, S. Hirschmeier, and D. Schoder. 2025. "Contrastive transformer network for long tail classification." *Knowledge-Based Systems*, 113607.

T. Miyato, A. M. Dai, and I. Goodfellow. 2021. *Adversarial Training Methods for Semi-Supervised Text Classification*. (2021). arXiv: 1605.07725 (stat.ML).

A. Moon, K. Kim, J. Lee, et al.. 2025. "Data augmentation for dense passage retrieval using corpus-passage frequency-based token deletion." *Journal of Big Data*, 12, 1, 1–28.

A. Mumuni and F. Mumuni. 2022. "Data augmentation: A comprehensive survey of modern approaches." *Array*, 16, 100258.

X.-P. Nguyen, S. Joty, K. Wu, and A. T. Aw. 2020. "Data diversification: A simple strategy for neural machine translation." *Advances in Neural Information Processing Systems*, 33, 10018–10029.

T. Niu and M. Bansal. Oct. 2018. "Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models." In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Ed. by A. Korhonen and I. Titov. Association for Computational Linguistics, Brussels, Belgium, (Oct. 2018), 486–496. doi:10.18653/v1/K18-1047.

A. Onan. 2023. "SRL-ACO: A text augmentation framework based on semantic role labeling and ant colony optimization." *Journal of King Saud University-Computer and Information Sciences*, 35, 7, 101611.

B. Pang and L. Lee. July 2004. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts." In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, (July 2004), 271–278. doi:10.3115/1218955.1218990.

B. Pang and L. Lee. June 2005. "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales." In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ed. by K. Knight, H. T. Ng, and K. Oflazer. Association for Computational Linguistics, Ann Arbor, Michigan, (June 2005), 115–124. doi:10.3115/1219840.1219855.

N. Parmar. June 2025. "Ensemble of Data Augmentation Techniques for Efficient Augmentation in NLP." *International Journal of Innovative Research in Advanced Engineering*, 11, (June 2025), 2706–2734.

A. Paszke et al.. 2019. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems*, 32.

B. Pavlyshenko and M. Stasiuk. Mar. 2025. "Using Large Language Models for Data Augmentation in Text Classification Models." *International Journal of Computing*, 24, 1, (Mar. 2025), 148–154. doi:10.47839/ijc.24.1.3886.

B. Peng, C. Li, P. He, M. Galley, and J. Gao. 2023. "Instruction tuning with gpt-4." *arXiv preprint arXiv:2304.03277*.

Y. Piao, S. Lee, D. Lee, and S. Kim. June 2022. "Sparse Structure Learning via Graph Neural Networks for Inductive Document Classification." *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 10, (June 2022), 11165–11173. doi:10.1609/aaai.v36i10.21366.

T. Qian, F. Li, M. Zhang, G. Jin, P. Fan, and W. Dai. 2022. "Contrastive learning from label distribution: A case study on text classification." *Neurocomputing*, 507, 208–220.

A. Radford, R. Jozefowicz, and I. Sutskever. 2017. "Learning to generate reviews and discovering sentiment." *arXiv preprint arXiv:1704.01444*.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. 2020. "Exploring the limits of transfer learning with a unified text-to-text transformer." *The Journal of Machine Learning Research*, 21, 1, 5485–5551.

R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam. 2021. "Hetegcn: heterogeneous graph convolutional networks for text classification." In: *Proceedings of the 14th ACM international conference on web search and data mining*, 860–868.

M. Rahman and M. L. Siddiq. 2025. "Code Comment Classification with Data Augmentation and Transformer-Based Models." In: *2025 IEEE/ACM International Workshop on Natural Language-Based Software Engineering (NLBSE)*, 33–36.

G. Rizos, K. Hemker, and B. Schuller. 2019. "Augment to prevent: short-text data augmentation in deep learning for hate-speech classification." In: *Proceedings of the 28th ACM international conference on information and knowledge management*, 991–1000.

M. Sajjadi, M. Javanmardi, and T. Tasdizen. 2016. "Regularization with Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning." In: *Proceedings of the 30th International Conference on Neural Information Processing Systems* (NIPS'16). Curran Associates Inc., Barcelona, Spain, 1171–1179. ISBN: 9781510838819.

R. Sennrich, B. Haddow, and A. Birch. Aug. 2016. "Improving Neural Machine Translation Models with Monolingual Data." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by K. Erk and N. A. Smith. Association for Computational Linguistics, Berlin, Germany, (Aug. 2016), 86–96. doi:10.18653/v1/P16-1009.

Y. Sun, Q. Liu, H. Zhu, and F. Tian. 2025. "LLMSeR: Enhancing Sequential Recommendation via LLM-based Data Augmentation." *arXiv preprint arXiv:2503.12547*.

V. Suresh and D. Ong. Nov. 2021. "Not All Negatives are Equal: Label-Aware Contrastive Loss for Fine-grained Text Classification." In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, (Nov. 2021), 4381–4394. doi:10.18653/v1/2021.emnlp-main.359.

J. Tang, M. Qu, and Q. Mei. 2015. "Pte: Predictive text embedding through large-scale heterogeneous text networks." In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1165–1174.

Z. Tang, M. Y. Kocyigit, and D. Wijaya. 2022. "AugCSE: Contrastive sentence embedding with diverse augmentations." *arXiv preprint arXiv:2210.13749*.

N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych. 2020. "Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks." *arXiv preprint arXiv:2010.08240*.

H. Touvron et al.. 2023. "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288*.

J. Tsujii. 2011. "Computational linguistics and natural language processing." In: *International Conference on Intelligent Text Processing and Computational Linguistics*, 52–67.

S. Ubani, S. O. Polat, and R. Nielsen. 2023. "Zeroshotdataaug: Generating and augmenting training data with chatgpt." *arXiv preprint arXiv:2304.14334*.

M. Wang, H. Gao, P. Zhang, and J. Zhang. 2024. "Prompt-Based Data Augmentation Framework for Few-Shot Named Entity Recognition." In: *International Conference on Intelligent Computing*, 451–462.

S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma. 2021. "Entailment as few-shot learner." *arXiv preprint arXiv:2104.14690*.

W. Y. Wang and D. Yang. Sept. 2015. "That's So Annoying‼!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Ed. by L. Màrquez, C. Callison-Burch, and J. Su. Association for Computational Linguistics, Lisbon, Portugal, (Sept. 2015), 2557–2563. doi:10.18653/v1/D15-1306.

Z. Wang, P. Wang, K. Liu, P. Wang, Y. Fu, C.-T. Lu, C. C. Aggarwal, J. Pei, and Y. Zhou. 2024. "A comprehensive survey on data augmentation." *arXiv preprint arXiv:2405.09591*.

Z. Wang, J. Zhang, X. Zhang, K. Liu, P. Wang, and Y. Zhou. July 2025. "Diversity-oriented Data Augmentation with Large Language Models." In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by W. Che, J.

Nabende, E. Shutova, and M. T. Pilehvar. Association for Computational Linguistics, Vienna, Austria, (July 2025), 22265–22283. ISBN: 979-8-89176-251-0. doi:10.18653/v1/2025.acl-long.1084.

Z. Wang, G. Xu, and M. Ren. 2024. "Llm-generated natural language meets scaling laws: New explorations and data augmentation methods." *arXiv preprint arXiv:2407.00322*.

Z. Wang, Z. Lin, S. Li, Y. Wang, W. Zhong, X. Wang, and J. Xin. 2023. "Dynamic multi-task graph isomorphism network for classification of alzheimer's disease." *Applied Sciences*, 13, 14, 8433.

J. Wei and K. Zou. Nov. 2019. "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks." 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)*. Ed. by K. Inui, J. Jiang, V. Ng, and X. Wan. Association for Computational Linguistics, Hong Kong, China, (Nov. 2019), 6382–6388. doi:10.18653/v1/D19-1670.

J. Wiebe, T. Wilson, and C. Cardie. 2005. "Annotating expressions of opinions and emotions in language." *Language resources and evaluation*, 39, 2, 165–210.

T. Wolf et al.. Oct. 2020. "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Q. Liu and D. Schlangen. Association for Computational Linguistics, Online, (Oct. 2020), 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

C. Woolsey, G. Leroy, and N. Maltman. 2025. "Enhancing text datasets with scaling and targeting data augmentation to improve BERT-based machine learners." *Expert Systems with Applications*, 128151.

X. Wu, S. Lv, L. Zang, J. Han, and S. Hu. 2019. "Conditional bert contextual augmentation." In: *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV 19*, 84–95.

Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. 2020a. "Unsupervised Data Augmentation for Consistency Training." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 6256–6268. https://proceedings.neurips.cc/paper_files/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf.

Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. 2020b. "Unsupervised data augmentation for consistency training." *Advances in neural information processing systems*, 33, 6256–6268.

Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng. 2017. "Data noising as smoothing in neural network language models." *arXiv preprint arXiv:1703.02573*.

X. Yao, Z. Huang, X. Hu, J. Yang, and Y. Guo. 2024. "Masking the unknown: leveraging masked samples for enhanced data augmentation." In: *The 40th Conference on Uncertainty in Artificial Intelligence*.

A. Yousefi Jordehi, M. Hosseini Khasheh Heyran, S. Ahmadnia, S. A. Mirroshandel, and O. Rambow. 2024. "Improving Opinion Mining Through Automatic Prompt Construction." *Journal of Information Systems and Telecommunication (JIST)*, 3, 47, 216.

L. Youssef, Z. Elhoussaine, N. Soufiane, and M. Noureddine. 2025. "Enhancing Arabic Aspect Category Detection Using Large Language Models (LLMs)." *Results in Engineering*, 105049.

A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. 2018. "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension." In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

S. Yu et al.. 2024. "Improving Text Classification by Leveraging Large Language Models for Data Augmentation." *Academic Journal of Computing & Information Science*, 7, 12, 91–95.

Y. Yu, S. Khadivi, and J. Xu. Oct. 2022. "Can Data Diversity Enhance Learning Generalization?" In: *Proceedings of the 29th International Conference on Computational Linguistics*. Ed. by N. Calzolari et al. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, (Oct. 2022), 4933–4945. https://aclanthology.org/2022.coling-1.437/.

F. Zeng, N. Chen, D. Yang, and Z. Meng. Dec. 2022. "Simplified-Boosting Ensemble Convolutional Network for Text Classification." *Neural Process. Lett.*, 54, 6, (Dec. 2022), 4971–4986. doi:10.1007/s11063-022-10843-4.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. 2017. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412*.

L. Zhang, Z. Yang, and D. Yang. 2022. "TreeMix: Compositional constituency-based data augmentation for natural language understanding." *arXiv preprint arXiv:2205.06153*.

R. Zhang, Y.-S. Wang, and Y. Yang. 2023. "Generation-driven contrastive self-training for zero-shot text classification with instruction-following LLM." *arXiv preprint arXiv:2304.11872*.

X. Zhang, J. Zhao, and Y. LeCun. 2015. "Character-level convolutional networks for text classification." *Advances in neural information processing systems*, 28.

Z. Zhang, M. Liu, X. Jia, G. Miao, X. Wang, H. Ni, and G. Wu. 2025. "Improving text classification via computing category correlation matrix from text graph." *Computer Speech & Language*, 89, 101688.

H. Zhao, H. Chen, T. A. Ruggles, Y. Feng, D. Singh, and H.-J. Yoon. 2024. "Improving text classification with large language model-based data augmentation." *Electronics*, 13, 13, 2535.

Received 26 October 2025; accepted 17 January 2026