
Less is More: Adaptive Coverage for Synthetic Training Data

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Synthetic training data generation with Large Language Models (LLMs) like
2 Google’s Gemma and OpenAI’s GPT offer a promising solution to the challenge
3 of obtaining large, labeled datasets for training classifiers. When rapid model
4 deployment is critical, such as in classifying emerging social media trends or
5 combating new forms of online abuse tied to current events, the ability to generate
6 training data is invaluable. While prior research has examined the comparability
7 of synthetic data to human-labeled data, this study introduces a novel sampling
8 algorithm, based on the maximum coverage problem, to select a representative
9 subset from a synthetically generated dataset. Our results demonstrate that training
10 a classifier on this contextually sampled subset achieves superior performance
11 compared to training on the entire dataset. This “less is more” approach not only
12 improves model accuracy but also reduces the volume of data required, leading to
13 potentially more efficient model fine-tuning.

14 1 Introduction

15 In recent years, the remarkable advancement in large language models (LLMs) such as OpenAI’s
16 GPT [1] or Google’s Gemma [51] have dramatically expanded the capability to generate extensive
17 synthetic textual data. Such synthetic data promises substantial utility for training machine learning
18 models, especially in domains where human-labeled data are prohibitively costly, inaccessible due
19 to privacy or ethical constraints, or impractical to acquire at scale [4, 11]. Consequently, synthetic
20 data generation has quickly become an appealing alternative for tuning models for various down-
21 stream tasks, including text classification, sentiment analysis, relation extraction, and information
22 retrieval [36].

23 However, the mere abundance of synthetic data does not guarantee superior model performance.
24 Increasing evidence demonstrates that naively utilizing large synthetic datasets introduces critical
25 pitfalls: notably, redundancy and imbalance [15, 31, 34]. LLM-generated samples frequently exhibit
26 redundancy by over-representing certain common patterns or phrases, potentially saturating datasets
27 with semantically repetitive information. Consider hate speech detection, where nuanced distinctions
28 between offensive, sarcastic, or context-dependent language are crucial: when prompted to generate
29 training examples, an LLM may produce many straightforwardly toxic utterances, yet underrepresent
30 borderline, coded, or indirect forms of harm [15, 21]. Such skewed representation not only dilutes
31 the informative value of synthetic datasets but actively harms model generalization and robustness
32 by obscuring valuable minority cases. Consequently, models trained on these synthetic corpora risk
33 becoming overly specialized on frequent cases, compromising predictive accuracy on more nuanced
34 real-world scenarios.

35 Motivated by these gaps, we propose Adaptive Coverage Sampling (ACS), a novel method that
36 effectively curates synthetic text datasets by carefully balancing redundancy, representational diversity,

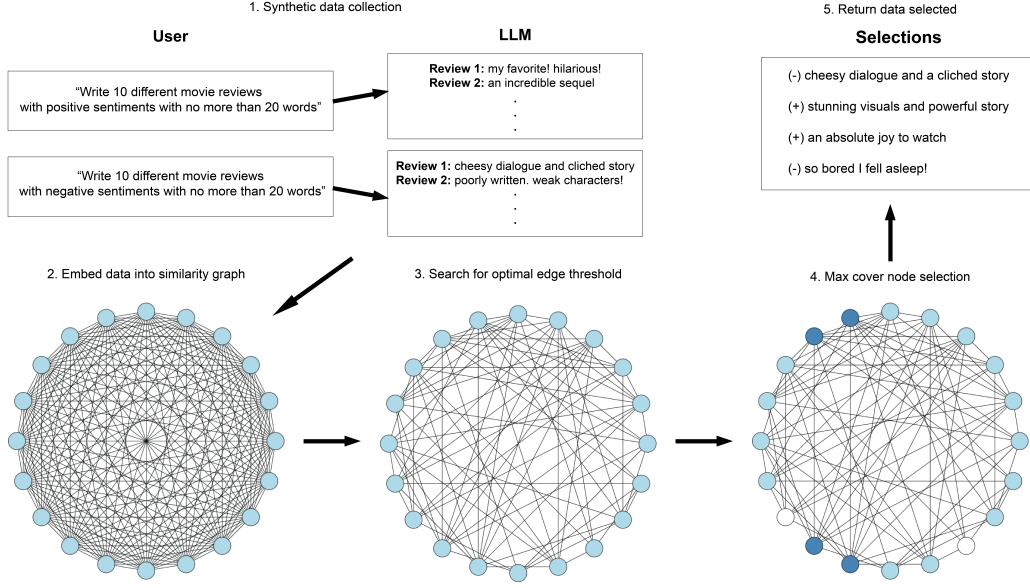


Figure 1: Overview of the ACS pipeline. (1) Prompt an LLM to generate a large pool of synthetic samples under user-specified constraints. (2) Samples are embedded into a semantic space and connected into a complete, weighted similarity graph. (3) Perform a binary search over edge-weight threshold to induce a subgraph. (4) Greedy max-cover procedure then iteratively selects the k nodes (highlighted in dark blue) that together cover desired fraction of the remaining graph (uncovered nodes depicted as white). (5) Selected subset is returned for downstream model training.

and computational efficiency. ACS uniquely frames synthetic data downsampling as a structured maximum-coverage optimization problem defined over a graph representation of the data. Specifically, synthetic text samples are first embedded into a latent semantic space, forming nodes within a complete graph where edges represent semantic similarity. Our approach leverages a binary search to systematically determine the optimal similarity threshold for edge pruning, thus inducing a sparser subgraph. Subsequently, a greedy maximum-coverage approximation algorithm selects the subset of k samples maximizing representational coverage, where coverage is defined as the proportion of the dataset “covered” by the selected subset and its similarity-neighbors.

A key strength of ACS lies in its use of a theoretically grounded binary search procedure to tune the pruning threshold, automating the trade-off between dataset compactness and semantic coverage. This allows the method to systematically filter out repetitive or redundant samples that might otherwise hinder model performance.

We evaluate ACS across several NLP tasks—including sentiment classification, relation extraction, and named entity recognition—and find that models fine-tuned on ACS-selected subsets match or outperform those trained on full synthetic datasets typically with just 10–30% of the original corpus. These results highlight the promise of principled data selection in synthetic data regimes: by identifying compact yet diverse training sets, ACS improves generalization while significantly reducing training compute cost. In doing so, our approach moves beyond heuristic-driven methods, offering a scalable and theoretically informed path toward more effective use of synthetic data.

2 Related Work

Large Language Models. LLMs, built upon the transformer architecture introduced by [53], have transformed language processing, achieving unprecedented performance across a broad spectrum of tasks including language modeling, translation, classification, and question-answering [3, 40, 50]. These models leverage massive-scale pretraining on extensive datasets to encode rich linguistic and factual knowledge, enabling fluent and contextually relevant text generation [51]. Consequently, the sophistication of LLM-generated content increasingly blurs the line between synthetic and authentic

human-written text [22, 43, 49, 56]. This indistinguishability raises an intriguing question: under what conditions and to what extent can LLM-generated data replace or complement human-annotated examples for training machine learning models?

Synthetic Training Data Generation. High-quality datasets crucially underpin the performance and generalization capabilities of modern machine learning systems. However, acquiring diverse and representative labeled data from human annotators is frequently costly, labor intensive, and fraught with privacy or ethical challenges [27, 16, 25, 45]. Moreover, human-generated annotations inherently carry biases or inconsistencies, potentially limiting their effectiveness in certain contexts. To overcome these limitations, synthetic data generation has emerged as a promising alternative, aimed at artificially populating underrepresented data regions and mitigating biases or gaps in existing datasets [15, 31].

To address data scarcity in specialized or emerging domains, researchers frequently employ data augmentation techniques to enhance model robustness and accuracy [10, 54]. Moreover, semi-supervised learning [37], multi-task learning [17], unsupervised pretraining [9, 41], and few-shot learning [8, 23] constitute alternative frameworks for learning from limited labeled examples. However, while effective in certain contexts, these approaches typically presume access to at least some high-quality human-generated examples as seed data, limiting their broader applicability.

Leveraging LLMs for Synthetic Data. LLMs offer a compelling approach to synthetic data generation due to their fluency, versatility, and capacity to mimic diverse linguistic styles and content structures [11]. Recent studies have demonstrated promising outcomes leveraging prompt-based methods (zero and few-shot) for generating training data for NLP tasks [34]. The effectiveness of synthetic datasets produced by these models depends critically on task characteristics, including the complexity of label spaces [11], the inherent subjectivity or ambiguity of the task [30], and crucially, the diversity and representativeness of generated samples [21]. Though the models are promising, these factors can impede naively employed models trained on synthetic datasets, potentially exacerbating redundancy and bias. Thus, underscoring the necessity of methods to carefully select or filter synthetic samples to maximize utility and minimize detrimental impacts.

Data Filtering and Downsampling. Filtering datasets to identify informative subsets for training constitutes a widely explored solution to the challenges posed by redundancy and imbalance, where conventional data selection techniques frequently rely on heuristic-based strategies and sample re-weighting schemes [2, 7, 35, 38, 39, 47, 52, 55, 57]. These methods largely revolve around assigning differential importance to data points based on criteria such as correctness, informativeness, or influence on model parameters [18].

Heuristic approaches typically leverage training dynamics or statistical properties of samples. For instance, dataset cartography [48] identifies and emphasizes data points classified as difficult or ambiguous through repeated training runs. Influence functions quantify individual data sample contributions by approximating how their exclusion alters model parameters [26]. Other methods, such as EL2N scoring [38], forgetting scores [52], and prototypicality assessments [47], attempt to prioritize or prune samples based on specific diagnostic measures. Recent studies have further explored the utility of LLM-based raters to directly score or filter synthetic samples based on quality heuristics. Notably, [6] proposed AlpaGasus, demonstrating that a curated, high-quality synthetic subset significantly improves downstream model performance over full synthetic datasets. However, their approach entails repeated queries to an LLM to iteratively refine sample sets, yielding a black-box rating metric which necessitates a computational (and potentially monetary) overhead in addition to careful threshold tuning.

In contrast, our ACS methodology provides a principled, computationally efficient, and explainable solution for optimal synthetic subsets without extensive manual tuning or iterative refinement. By formulating the selection problem as a graph-based maximum coverage optimization and leveraging an adaptive binary search to systematically adjust similarity thresholds, ACS ensures theoretical rigor and practical efficacy. Crucially, ACS consistently demonstrates superior performance using significantly smaller synthetic subsets compared to prior filtering methods, thereby establishing a new benchmark for efficient and effective synthetic data utilization.

3 Preliminaries & Methodology

In this section, we detail our comprehensive pipeline for curating a representative subset from large synthetic datasets, specifically designed to improve model training efficiency and downstream task performance. We begin by describing the generation and preprocessing of synthetic textual data, then present multiple baseline downsampling methods employed for comparative evaluation. Subsequently, we introduce and rigorously define our novel ACS method, highlighting its theoretical foundations and practical implementation. Finally, we describe our approach for fine-tuning the BERT model with the selected subset.

3.1 Synthetic Data Generation

We utilize a synthetic corpus of text generated by GPT-3.5 [1]. The corpus employed is based on established prompt templates tailored to specific downstream tasks (e.g. sentiment analysis), as detailed by prior work [11]. Each dataset is balanced across labels to ensure sufficient diversity, carefully selecting an equal number of data points per label. While synthetic datasets provide vast training material, redundancy frequently arises as similar semantic content is generated repeatedly [34].

3.2 Downsampling Methods.

To mitigate redundancy and maximize representational coverage, we explore several distinct downsampling techniques. Our goal is to select a subset of size $k < N$ from an initial corpus of size N , preserving data diversity while enhancing computational efficiency.

Baseline Methods. We benchmark our novel ACS approach against widely used benchmark methods. **Random** sampling henceforth refers to uniformly at random selecting k samples from the corpus. **EL2N** [38] ranks samples by the average L_2 distance between model predictions and true labels across early training checkpoints, emphasizing persistently challenging examples. **Forgetting scores** [52] count transitions between correct and incorrect model predictions per sample during training, emphasizing samples near the decision boundary. **Prototypicality** [47] which computes class-specific embeddings and prioritizes samples closest to their class centroids, capturing representative class examples. **LLM rater (AlpaGasus)** [6] employs GPT-3.5 to assign quality ratings to each synthetic input-output pair, retaining only the highest ranked samples, thereby enhancing subset quality through language-model-informed filtering. Each baseline is implemented to rank the dataset according to the respective criteria, selecting the top k samples for training.

Adaptive Coverage Sampling. ACS introduces a graph-based max-coverage sampling technique to systematically select representative subsets. Samples are first embedded into a latent semantic space using Gecko embeddings [29], though ACS is broadly compatible with alternative embedding methods. We construct a similarity graph where each node represents a sample, and edges indicate cosine similarity exceeding a dynamic threshold. This threshold is optimized via a binary search to achieve a user-specified graph coverage level. Coverage formally quantifies representational breadth:

Definition 1 (Coverage). *Let $G = (V, E)$ be a graph with vertex set V , edge set E , and self-loop for all vertices. A subset $H \subseteq V$ of size $|H| = k$ achieves coverage $c \in [0, 1]$ if*

$$\left| \bigcup_{i \in H} N_i \right| = c \cdot |V|$$

where N_i is the neighborhood of vertex $i \in H$ (ie. i covers the elements of N_i , including itself).

A coverage of 1.0 thus ensures every node is either selected or directly adjacent to a selected node, while lower coverage levels strategically exclude less representative samples. We leverage the following theorem guaranteeing monotonicity of an exact solution to the max cover problem with respect to similarity thresholds on the pruned graph, validating our subsequent binary search procedure.

Theorem 1. *Let D be a dataset, and for each similarity threshold s_i , construct a similarity graph $G_i(V, E_i)$, where V represents the data points and $(u, v) \in E_i$ if and only if the cosine similarity between u and v exceeds s_i . Let $H_i \subseteq V$ be the set of k samples selected by the max coverage algorithm on G_i , and let c_i denote the coverage achieved by H_i . For any two thresholds s_i and s_j*

160 such that $s_j < s_i$, the similarity graph $G_j(V, E_j)$ has a coverage $c_j \geq c_i$ when maximally covered
 161 by k samples.

162 *Proof.* Consider two similarity thresholds s_i and s_j such that $s_j < s_i$. The corresponding similarity
 163 graphs $G_i(V, E_i)$ and $G_j(V, E_j)$ are constructed by adding edges between data points whose cosine
 164 similarity exceeds s_i and s_j , respectively. Since $s_j < s_i$, it follows that $E_i \subseteq E_j$; that is, G_j includes
 165 all the edges from G_i , possibly with additional edges.

166 Now, let $H_i \subseteq V$ be the set of k samples selected by the max coverage algorithm on G_i , which
 167 achieves coverage c_i . The coverage c_i is defined as the proportion of vertices in V that are adjacent
 168 to at least one vertex in H_i . Since $E_i \subseteq E_j$, the set of neighbors of each vertex in H_i in G_i is a
 169 subset of the neighbors of the same vertex in G_j . Therefore, the coverage achieved by H_i in G_j is at
 170 least as large as the coverage in G_i . More formally, if H_j is the set of k samples selected by the max
 171 coverage algorithm on G_j , we have:

$$c_j = \left| \bigcup_{v \in H_j} N_j(v) \right| \quad \text{and} \quad c_i = \left| \bigcup_{v \in H_i} N_i(v) \right|,$$

172 where $N_j(v)$ and $N_i(v)$ denote the neighborhoods of v in G_j and G_i , respectively. Since $E_i \subseteq E_j$,
 173 we have $N_i(v) \subseteq N_j(v)$ for all $v \in V$, implying that the coverage in G_j is at least as large as the
 174 coverage in G_i . Therefore, $c_j \geq c_i$. \square

175 The monotonicity of coverage allows us to find the largest similarity threshold that achieves a
 176 coverage equal to, or greater than, the target coverage. This thresholding ensures that the max
 177 coverage component of ACS focuses on the most relevant and diverse samples to achieve the target
 178 coverage. We note that the max coverage problem is NP-hard [14], and that our implementation uses
 179 the greedy approximation [24] which is not guaranteed to be monotonic. However, we show that, in
 180 practice, this monotonicity persists (see Section 4.1).

181 Leveraging this result, we conduct a binary search on the similarity threshold for edge pruning and
 182 execute the greedy max cover algorithm. Specifically, we sequentially select the node of highest
 183 degree, add the selected node and all of its neighbors to the set of “covered nodes” and repeat until
 184 k nodes are selected. We subsequently compute the coverage of the full dataset from the selected
 185 subset and, based on this coverage’s deviation from the target, adjust the threshold in accordance
 186 with the binary search until convergence. The k selected points from the max cover execution on the
 187 optimally pruned graph are finally returned.

188 To ensure scalability and enhance representational diversity, we impose a maximum nearest neighbors
 189 constraint per node, significantly reducing computational complexity and ensuring effective cover-
 190 age. Specifically, we define a strict constraint d_{\max} , a bound ensuring sufficient but limited graph
 191 connectivity, derived via the extended pigeonhole principle: $d_{\max} > cN/k$. This constraint further
 192 improves computational tractability and sample diversification, analogous to scalability techniques like
 193 Locality-Sensitive Hashing (LSH) with limited bucket sizes [44]. Parameter sensitivity experiments
 194 are detailed in Appendix ??.

195 3.3 Comparative Experiments

196 After generating and downsampling the synthetic dataset to obtain k training samples, we employ
 197 two comparative measures. First, we fine-tune a BERT model [9] on the selected subset and report
 198 the F1-scores as a function of the number of subsamples selected for model training¹. We use the
 199 BERT_{base}, uncased model (108 million parameters) and fine-tune it for multiple epochs (the exact
 200 number is defined for each respective experiment in Section 5). The model’s weights are mostly
 201 initialized using pre-trained weights, while the parameters of the final classification layer (2048
 202 units) are randomly initialized. Specifically, the weights of this layer are initialized from a normal
 203 distribution with a mean of 0 and a standard deviation of 0.02, following standard practices for
 204 fine-tuning transformer-based models.[9, 12, 28, 32]. The fine-tuning process uses a batch size of
 205 16, a learning rate of 2×10^{-5} , and a dropout rate of 0.1. All experiments are conducted on a
 206 high-performance GPU cluster with 16GB of RAM, with $n = 5$ distinct random seeds used for

¹We here use F1, rather than accuracy, to remain robust to potential class imbalances in the test sets.

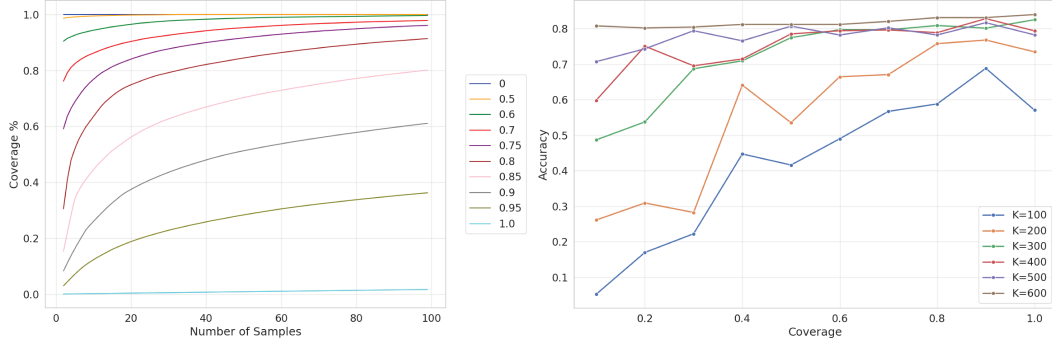


Figure 2: (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

model initialization. Details of the implementation, including all hyperparameters, are provided in the supplementary material, along with the training codes.

Second, we compute the self-bilingual evaluation understudy (or SelfBLEU) metric as a quantifiable measure of subset diversity [58]. This widely used metric computes word similarity between sentences or documents within a dataset. Crucially, a higher SelfBLEU score indicates a dataset with higher self-similarity. Thus, the *reciprocal* of this metric is used as a diversity measure (higher self-similarity implies less diversity in the set).

4 Empirical Analysis of ACS

In this section, we empirically validate critical aspects of our sampling method. Specifically, we first verify the empirical monotonicity of coverage as a function of similarity threshold for the greedy approximate algorithm for the max coverage problem, aligning with the theoretical guarantees provided by Theorem 1. We then systematically identify and analyze the coverage parameter value, demonstrating that coverage below 1.0 consistently yields better performance in downstream NLP tasks.

4.1 Empirical Validation of Monotonicity

A central assumption is ACS is that coverage monotonically increases or remains constant as the similarity threshold decreases, as formally established for the exact max coverage solution. To confirm this assumption’s practical validity under the greedy approximation algorithm [24], we conducted detailed empirical experiments across varying similarity thresholds.

We focus initially on the synthetic textual data generated to emulate the SST2 sentiment analysis task [46]. This synthetic dataset comprises short movie reviews labeled as positive or negative sentiments. Additional validation on other datasets, is provided in the supplementary materials.

Each text sample was first embedded into a latent semantic space using Gecko embeddings [29]. Subsequently, similarity graphs were constructed for multiple fixed similarity thresholds, after which the greedy max-coverage approximation algorithm was executed to select subsets of varying sizes k . As illustrated in the left-hand plot of Figure 2, coverage consistently exhibits monotonic behavior: as the similarity threshold decreases (adding more edges), coverage either remains constant or strictly increases, validating our core theoretical assumption in practical scenarios. Notably, the maximum possible coverage (full coverage, $c = 1$) is achieved quickly at lower thresholds, while all plots achieve a minimal coverage of $c = k/N$.

4.2 Determining the Optimal Coverage Level

While full coverage ($c = 1$) intuitively seems optimal, in practice, we demonstrate that lower coverage values yield better model performance. We hypothesize that this is due to the exclusion of redundant

or noisy samples. To systematically investigate this, we varied the coverage parameter across a broad range of values, maintaining a fixed subset size of k for the synthetic SST2 dataset (with analogous findings for additional tasks reported in the Appendix A.1).

For each coverage setting, ACS selected a subset of exactly k samples which achieved an effective coverage of the target. Using these subsets, we fine-tuned BERT_{base} models and evaluated their accuracy on a human-annotated test set. The resulting accuracy trends are presented on the right Figure 2. Notably, accuracy significantly improves as the target coverage increases from lower levels, reflecting greater representational completeness. However, accuracy consistently peaks before reaching full coverage, with performance slightly deteriorating at or near the full coverage ($c = 1$). These results robustly support our assertion that carefully selecting subsets with moderate coverage offers superior model generalization and training efficiency.

5 Fine-Tuning for Downstream Tasks

In this section, we rigorously evaluate the performance of ACS against several established baseline downsampling methods on multiple NLP benchmarks. Specifically, we assess sequence-level tasks (sentiment analysis and relation extraction) and a token-level task (named entity recognition), demonstrating ACS’s consistent advantages in terms of model performance and diversity of selected subsets. These evaluations compliment one another: improved performance corresponding to higher diversity in the selected subsets and vice versa.

5.1 Sequence-Level Tasks

Sentiment Analysis We first evaluate our approach on the binary sentiment classification task using the synthetic corpus from [11], designed to emulate the SST2 dataset [46]. This dataset contains $N = 6,000$ synthetic movie reviews, equally split between positive and negative sentiments. Following the prior literature [11], we fine-tune a BERT_{base} model for 32 epochs (with early stopping) on subsets selected by each downsampling method.

Results. Figure 3 compares the performance (F1-score) of ACS against the baseline methods, averaging results over five random initializations of the BERT classification layer weights. ACS consistently outperforms alternative methods across all subset sizes, with particularly notable improvements at smaller subset sizes. Remarkably, ACS achieves performance comparable to training on the *full* synthetic dataset (black dashed line) using only approximately 10% of the data, underscoring its effectiveness in isolating highly informative samples. We note that while ACS performs the best, all methods (apart from random) yield aggressively pruned datasets which can match performance on the full dataset. This suggests that for the simpler task of positive or negative sentiment detection, only a few meaningful examples are needed to train a sophisticated classifier to effectively categorize the inputs.

To further elucidate why ACS in particular achieves superior performance, Figure 3 further plots diversity (inverse SelfBLEU score) across subset sizes. ACS-selected subsets consistently exhibit greater diversity compared to baseline methods, strongly correlating with improved downstream task performance. This enhanced diversity seems to mitigate redundancy and better equips models to generalize effectively, particularly when training data sizes are limited.

Relation Extraction Relation extraction, exemplified by the FewRel dataset [20], represents a significantly more challenging classification task due to its large set of 64 distinct relation labels. The task involves predicting the labeled relation between two marked entities within a sentence, necessitating both greater diversity and precision in the synthetic data generation process. For instance, the sentence, “Chester Alan Arthur, 21st President of the United States, died of this disease on November 18, 1886,” could be labeled with the relation “head of government” to capture the connection between Arthur and his role as President. This increased complexity necessitates careful selection of diverse and informative examples. We employ the synthetic corpus of relation extraction data from [11], uniformly sampling $N = 12,800$ examples spread across all relation labels in accordance with the FewRel dataset. The BERT_{base} model is fine-tuned over 3 epochs as in the prior work.

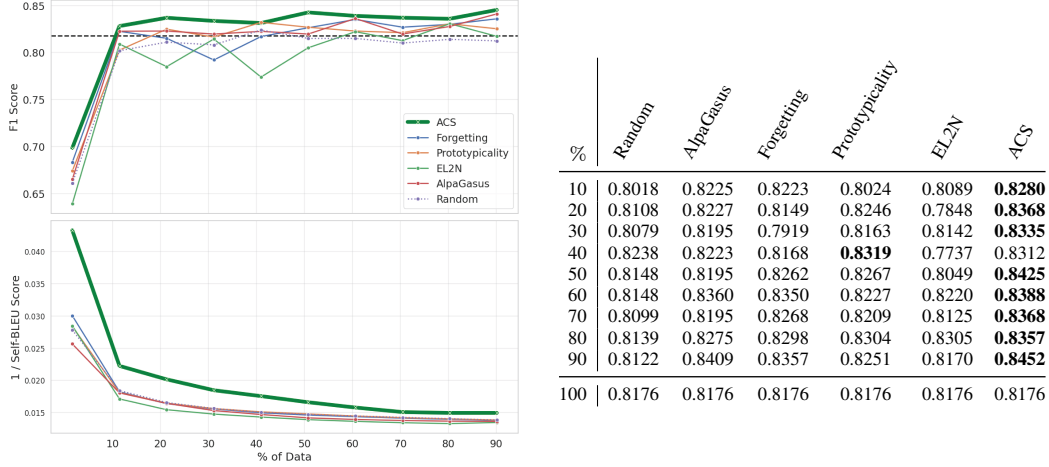


Figure 3: (L) F1 scores (top) and SelfBLEU diversity (bottom) for SST2 as a function of subset size, comparing downsampling methods. Horizontal dotted line represents model performance when trained on all available data (no pruning). (R) Tabulated F1 results corresponding with plots. ACS matches or outperforms full-data training with only 10% of the data.

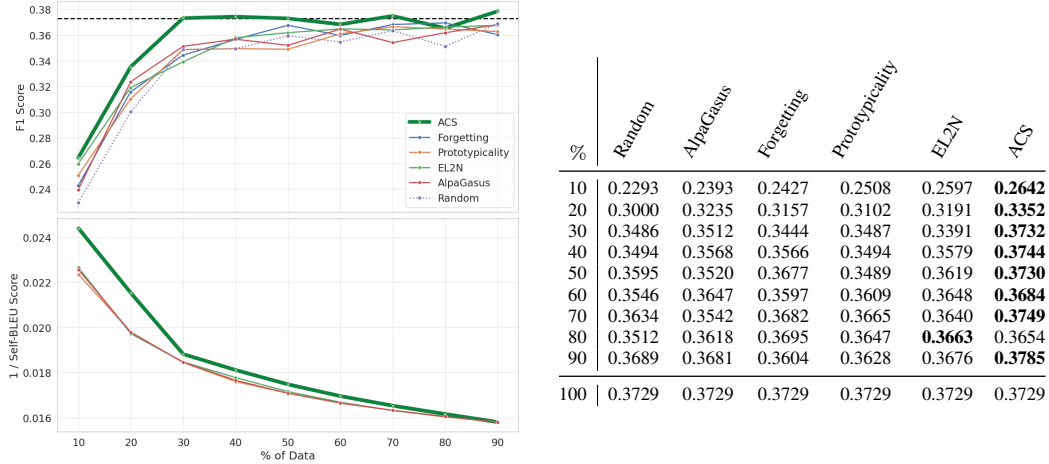


Figure 4: (L) F1 scores (top) and SelfBLEU diversity (bottom) for FewRel as a function of subset size, comparing downsampling methods. Horizontal dotted line represents model performance when trained on all available data (no pruning). (R) Tabulated F1 results corresponding with plots. ACS matches or outperforms full-data training with only 30% of the data.

Results. Figure 4 presents the F1-score results on the synthetic FewRel dataset, clearly demonstrating that ACS consistently surpasses the baseline methods at nearly all data subsampling proportions. Similar to the sentiment analysis task, ACS achieves competitive or superior performance using just 30% of the available synthetic data. Figure 4 provides additional support by showing that subsets selected by ACS, again, obtain substantially lower SelfBLEU scores, indicating greater representational diversity. This enhanced diversity is particularly valuable for relation extraction, which benefits from nuance and varied training examples to better capture the complex semantic relations between entities.

5.2 Token-Level Task: Named Entity Recognition

We lastly validate ACS on the token-level named entity recognition (NER) task using a synthetic corpus generated to match the AI domain split of CrossNER [33]. This task involves labeling each token in a sentence with one of 14 distinct entity classes or a null identifier. For example, on the sentence: “We evaluated BERT using the SQuAD benchmark and compared its performance with

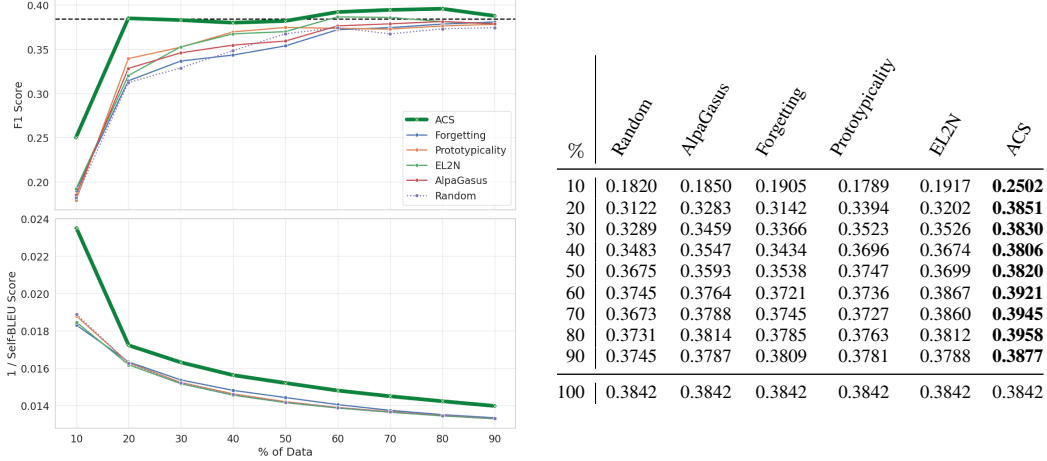


Figure 5: (L) F1 scores (top) and SelfBLEU diversity (bottom) for CrossNER as a function of subset size, comparing downsampling methods. Horizontal dotted line represents model performance when trained on all available data (no pruning). (R) Tabulated F1 results corresponding with plots. ACS matches or outperforms full-data training with only 20% of the data.

BiDAF on multiple F1-score metrics.” a classifier would have to mark the relevant tokens (BERT, SQuAD, BiDaF, F1-score) with the labels (Tool, Dataset, Tool, Metric) respectively. The synthetic corpus used here contains $N = 3,000$ sentences, each carefully generated to reflect diverse entity mentions. We crucially highlight for this *token-level* classification task, we still deploy ACS on the *sentence* embeddings to isolate the most representative samples. The selected sentences are subsequently parsed back into their tokenization for classification. We fine-tune a BERT_{base} model specific to the NER task [42] over 50 epochs, applying early stopping to prevent overfitting.

Results. Figure 5 illustrates ACS’s performance on the token-level classification task. Using only 20% of the original synthetic dataset, ACS achieves accuracy comparable to training on the entire dataset. Furthermore, ACS consistently selects subsets with notably greater diversity, as evidenced by lower SelfBLEU scores compared to baselines. This confirms ACS’s capability to effectively capture a wide representation of the dataset, even for precise token-level predictions.

6 Discussion

Our experiments convincingly demonstrate that ACS effectively distills large synthetic datasets into smaller, highly representative subsets, significantly enhancing model training efficiency and accuracy. Several distinctive strength set ACS apart from existing downsampling and filtering methods.

First, ACS reliably identifies remarkably small subsets—often around 20% or even less of the original synthetic dataset—that allow models to achieve performance matching or surpassing that of models trained on the full dataset. This capability underscores the potential efficiency gains and practical utility of ACS in real-world scenarios, especially when computational resources or training time are limited. Second, unlike many alternative methods that require fitting multiple models or extensive hyperparameter tuning to gauge sample importance, ACS does not depend on repeated training iterations. Instead, our method leverages a straightforward binary search on a similarity graph. Third, ACS does not rely on label information during the subset selection phase, making it broadly applicable to both supervised and unsupervised scenarios. This feature notably enhances its versatility, enabling effective deployment in diverse data scenarios without requiring preliminary labeling efforts. Lastly, ACS explicitly focuses on identifying optimal *collections* of data points rather than individual samples with maximal individual contribution. This collection oriented approach ensures that the selected subsets comprehensively represent the overall dataset diversity and structure, rather than emphasizing potentially redundant or outlier points that individually maximize some criterion. As such, ACS offers a robust, efficient, and versatile approach to synthetic data distillation, delivering substantial improvements in downstream task performance through highly informative and diverse subsets.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.
- [3] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [4] Andreas Bunte, Frank Richter, and Rosanna Diovialvi. Why it is hard to find ai in smes: A survey from the practice and how to promote it. In *ICAART (2)*, pages 614–620, 2021.
- [5] CJ Carey, Jonathan Halcrow, Rajesh Jayaram, Vahab Mirrokni, Warren Schudy, and Peilin Zhong. Stars: Tera-scale graph building for clustering and learning. *Advances in Neural Information Processing Systems*, 35:21470–21481, 2022.
- [6] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpapasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.
- [7] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.
- [8] Shumin Deng, Ningyu Zhang, Zhanlin Sun, Jiaoyan Chen, and Huajun Chen. When low resource nlp meets unsupervised language model: Meta-pretraining then meta-learning for few-shot text classification (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13773–13774, 2020.
- [9] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [10] Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. Daga: Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv:2011.01549*, 2020.
- [11] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. Is gpt-3 a good data annotator? *arXiv preprint arXiv:2212.10450*, 2022.
- [12] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.
- [13] Alessandro Epasto, Andrés Muñoz Medina, Steven Avery, Yijian Bai, Robert Busa-Fekete, CJ Carey, Ya Gao, David Guthrie, Subham Ghosh, James Ioannidis, et al. Clustering for private interest-based advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2802–2810, 2021.
- [14] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
- [15] Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets. *arXiv preprint arXiv:2404.14361*, 2024.
- [16] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.
- [17] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

- [18] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coresets selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022.
- [19] Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, and Bryan Perozzi. Grale: Designing networks for graph learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2523–2532, 2020.
- [20] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018.
- [21] Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. Synthetic data in ai: Challenges, applications, and ethical implications. *arXiv preprint arXiv:2401.01629*, 2024.
- [22] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, 2022.
- [23] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021.
- [24] Dorit S Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems*, pages 94–143. 1996.
- [25] Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard. In *The Twelfth International Conference on Learning Representations*.
- [26] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.
- [27] Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.
- [28] Z Lan. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.
- [29] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.
- [30] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*, 2023.
- [31] Ruijie Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data. In *First Conference on Language Modeling*, 2024.
- [32] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [33] Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13452–13460, 2021.
- [34] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*, 2024.

- [35] Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *arXiv preprint arXiv:2310.07931*, 2023.
- [36] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477, 2022.
- [37] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.
- [38] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.
- [39] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.
- [40] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- [41] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- [42] Thilina C Rajapakse, Andrew Yates, and Maarten de Rijke. Simple transformers: Open-source for all. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 209–215, 2024.
- [43] Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, 2022.
- [44] Sarath Shekkizhar, Neslihan Bulut, Mohamed Farghal, Sasan Tavakkol, MohammadHossein Bateni, and Animesh Nandi. Data sampling using locality sensitive hashing for large scale graph learning. 2023.
- [45] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*.
- [46] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.
- [47] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.
- [48] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, 2020.
- [49] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*, 2023.
- [50] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

- 481 [51] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya
482 Pathak, Laurent Sifre, Morgane Rivi re, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open
483 models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- 484 [52] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio,
485 and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network
486 learning. *arXiv preprint arXiv:1812.05159*, 2018.
- 487 [53] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*,
488 2017.
- 489 [54] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on
490 text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.
- 491 [55] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A
492 universal method of data selection for real-world data-efficient deep learning. In *The Eleventh*
493 *International Conference on Learning Representations*, 2022.
- 494 [56] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and
495 Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings*
496 *of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–
497 11669, 2022.
- 498 [57] Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for
499 high pruning rates. In *The Eleventh International Conference on Learning Representations*.
- 500 [58] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu.
501 Texusgen: A benchmarking platform for text generation models. In *The 41st international ACM*
502 *SIGIR conference on research & development in information retrieval*, pages 1097–1100, 2018.

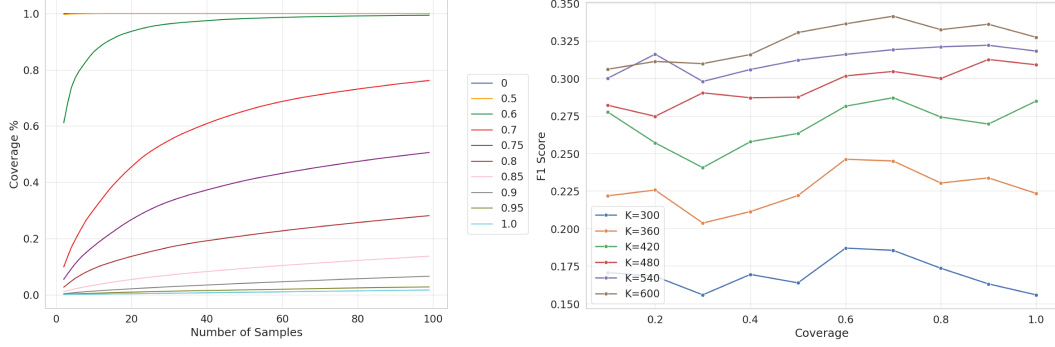


Figure 6: Empirical results for the FewRel dataset. (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

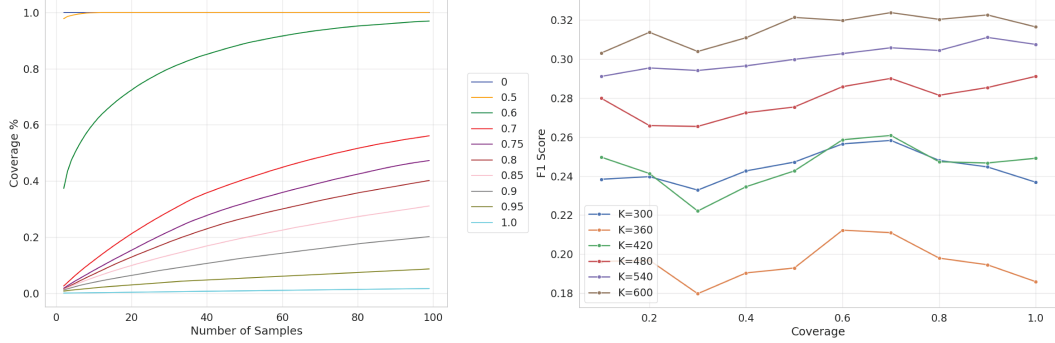


Figure 7: Empirical Results for the CrossNER dataset. (L) Coverage of data increases with k or when decreasing the similarity threshold. Colors correspond to the fixed similarity thresholds depicted in the legend. (R) Model accuracy as a function of coverage level for the sentiment analysis tasks. Performance peaks at a coverage level below 1.0.

503 A Omitted Results

504 We here present the empirical analysis of Section 4 on the FewRel and CrossNER datasets. We
 505 further present a sensitivity analysis to the max degree parameter for all of the datasets.

506 A.1 Empirical Analysis of ACS

507 We begin with the empirical ACS validation of Section 4 for the remaining datasets. In both instances,
 508 we observe consistent monotonicity in the coverage as a function of k -selection with decreasing
 509 similarity thresholds, as well as improved downstream task performance with coverage values less
 510 than 1.0. Figure 6 presents the empirical results for the FewRel dataset and Figure 7 for CrossNER.
 511 In both instances, the greedy approximation to max coverage exhibits monotonicity as needed for the
 512 binary search procedure. We further see that full coverage is non-optimal in most instances, further
 513 motivating our usages of coverage = 0.9 throughout the experimental results.

514 B Scalability of Adaptive Coverage Sampling

515 In large-scale settings, the computational cost of optimizing the similarity threshold τ for ACS can
 516 become prohibitive due to the $O(n^2)$ complexity of evaluating pairwise similarities. Though we can
 517 speed up such computations with methods such as Locality Sensitive Hasing (LHS) or hop-spanner
 518 methods [5, 13, 19], we further propose a scalable variant of ACS that conducts threshold selection
 519 on a small random subset of the data. For a desired downsampling value of $k \ll N$, we uniformly

at random select a small subgraph of $N' < N$ nodes and run the ACS procedure on the reduced instance. Once the optimal edge similarity threshold τ^* is identified on this subset, it is reused to construct the similarity graph and perform ACS on the *full* dataset. This approach significantly reduces computational cost while maintaining effective coverage.

Formally, let $G = (V, E)$ be the similarity graph constructed on the full dataset, where edges are defined between points with similarity exceeding a threshold τ . Let $V' \subset V$ denote a uniformly random subsample of size N' , and let $G' = (V', E')$ be the induced subgraph. For any subset $S \subset V$, we define the normalized coverage as the fraction of nodes in V that are neighbors of some node in S under threshold τ . We proceed to show that threshold tuning on the subsample generalizes well to the full dataset, in the following proposition.

Proposition 1. *Let $\tau \in [0, 1]$ be fixed. For any subset $S' \subset V'$ of size k , let $\text{Cov}_\tau(S'; V') = \frac{|\bigcup_{v \in S'} \{u \in V : \text{sim}(u, v) \geq \tau\}|}{|V|}$ be the normalized coverage. Let $S \subset V$ be the greedy max coverage selection of size K using the same threshold. Then, with high probability over the choice of V' , we have:*

$$|\text{Cov}_\tau(S; V) - \text{Cov}_\tau(S'; V')| \leq \varepsilon$$

for some $\varepsilon \in O\left(\sqrt{\frac{\log(1/\delta)}{N'}}\right)$, where δ is the failure probability.

Proof. Let $G = (V, E)$ be our original similarity graph and let $V' \subseteq V$ be a subset of vertices chosen uniformly at random, with $|V'| = N'$. Consider the induced subgraph $G' = (V', E')$. For a fixed threshold $\tau \in [0, 1]$, we define coverage for a subset $S \subseteq V$ as

$$\text{Cov}_\tau(S; V) = \frac{|\bigcup_{v \in S} \{u \in V : \text{sim}(u, v) \geq \tau\}|}{|V|}.$$

Let $S \subseteq V$ and $S' \subseteq V'$ each be greedy max coverage selections of size k using threshold τ on graphs G and G' , respectively. Our goal is to show that, with high probability,

$$|\text{Cov}_\tau(S; V) - \text{Cov}_\tau(S'; V')| \leq \varepsilon,$$

for $\varepsilon \in O\left(\sqrt{\frac{\log(1/\delta)}{N'}}\right)$, with failure probability at most δ .

We first compute expected relationships. The expected number of edges in the induced subgraph G' is

$$\mathbb{E}[|E'|] = \frac{\binom{|V'|}{2}}{\binom{|V|}{2}} |E|,$$

and the expected degree on G' , d' , is

$$\mathbb{E}[d'] = \frac{|V'| - 1}{|V| - 1} d_{\text{avg}}.$$

Thus, for a fixed sets $S \subseteq V$ and $S' \subseteq V'$ of size k , we have

$$\mathbb{E}[\text{Cov}_\tau(S'; V')] = \frac{|V|(|V'| - 1)}{|V'|(|V| - 1)} \text{Cov}_\tau(S; V).$$

Define the indicator random variables

$$X_u = \mathbf{1}[\exists v \in S', \text{sim}(u, v) \geq \tau], \quad u \in V',$$

so that coverage can be expressed as

$$\text{Cov}_\tau(S'; V') = \frac{1}{|V'|} \sum_{u \in V'} X_u.$$

Applying Hoeffding's inequality, we have for any $t > 0$,

$$\Pr(|\text{Cov}_\tau(S'; V') - \mathbb{E}[\text{Cov}_\tau(S'; V')]| \geq t) \leq 2 \exp(-2|V'|t^2).$$

548 Choosing $t = \sqrt{\frac{\log(2/\delta)}{2|V'|}}$, we get with probability at least $1 - \delta$,

$$|\text{Cov}_\tau(S'; V') - \mathbb{E}[\text{Cov}_\tau(S'; V')]| \leq \sqrt{\frac{\log(2/\delta)}{2|V'|}}.$$

549 Therefore, using these inequalities, we bound

$$\begin{aligned} |\text{Cov}_\tau(S; V) - \text{Cov}_\tau(S'; V')| &\leq |\text{Cov}_\tau(S; V) - \mathbb{E}[\text{Cov}_\tau(S'; V)]| + |\mathbb{E}[\text{Cov}_\tau(S'; V)] - \text{Cov}_\tau(S'; V')| \\ &\leq \sqrt{\frac{\log(2/\delta)}{2|V'|}} + O\left(\frac{|V| - |V'|}{|V'|(|V| - 1)}\right). \end{aligned}$$

550 Assuming $|V'|$ sufficiently large relative to $|V|$, the second term becomes negligible compared to the
551 Hoeffding term. Hence, we conclude that with probability at least $1 - \delta$,

$$|\text{Cov}_\tau(S; V) - \text{Cov}_\tau(S'; V')| \leq \varepsilon, \quad \text{where} \quad \varepsilon = O\left(\sqrt{\frac{\log(1/\delta)}{|V'|}}\right).$$

552 This completes the proof. □

553 This proposition establishes that threshold selection on a small sample yields an accurate coverage
554 estimate on the full dataset, with the accuracy improving for larger datasets.

555 To validate this claim empirically, we conducted a series of experiments across the datasets used in
556 the main text (sentiment analysis, relation extraction, and named entity recognition). In each setting,
557 we selected a random subset of the data at varying proportions, ranging from very small to nearly the
558 full dataset. For each subset, we used binary search to identify the threshold τ^* such that the greedy
559 ACS procedure on the subset achieved a fixed target of 90% coverage with k examples. We then
560 applied this same threshold τ^* to construct the similarity graph for the full dataset and ran the greedy
561 max coverage to select a size- K subset, measuring the resulting coverage over all data points.

562 Figure 10 summarizes the results. Each plot corresponds to a different dataset (SST2, FewRel, or
563 CrossNER). The x-axis represents the fraction of the dataset used to compute the optimal threshold,
564 and the y-axis shows the actual coverage obtained on the full dataset using that threshold. A shaded
565 band indicates an ε -envelope centered at the target coverage of 90% where ε is set to be 5×10^{-3} .
566 Across all settings, we observe that even small subsamples, often less than 20% of the full dataset,
567 yield thresholds that generalize well. As the sample size increases, the coverage rapidly converges to
568 the target, and variance remains low throughout.

569 These results provide strong empirical support for the scalable ACS approach. By selecting a
570 threshold on a small, randomly drawn subset, we can achieve nearly identical coverage behavior on
571 the full dataset, enabling efficient and accurate training data selection in large-scale scenarios without
572 repeated expensive graph construction or threshold tuning.

573 We note that the above experiments, in line with Proposition 1 do not impose any max degree
574 constraints on the similarity graph. We demonstrate that even when such constraints are imposed, the
575 scalability of optimal threshold remains. In Figure 10, we again impose the max degree constraint of
576 $2 \cdot c \cdot N/k$ and set a target coverage of 0.5.

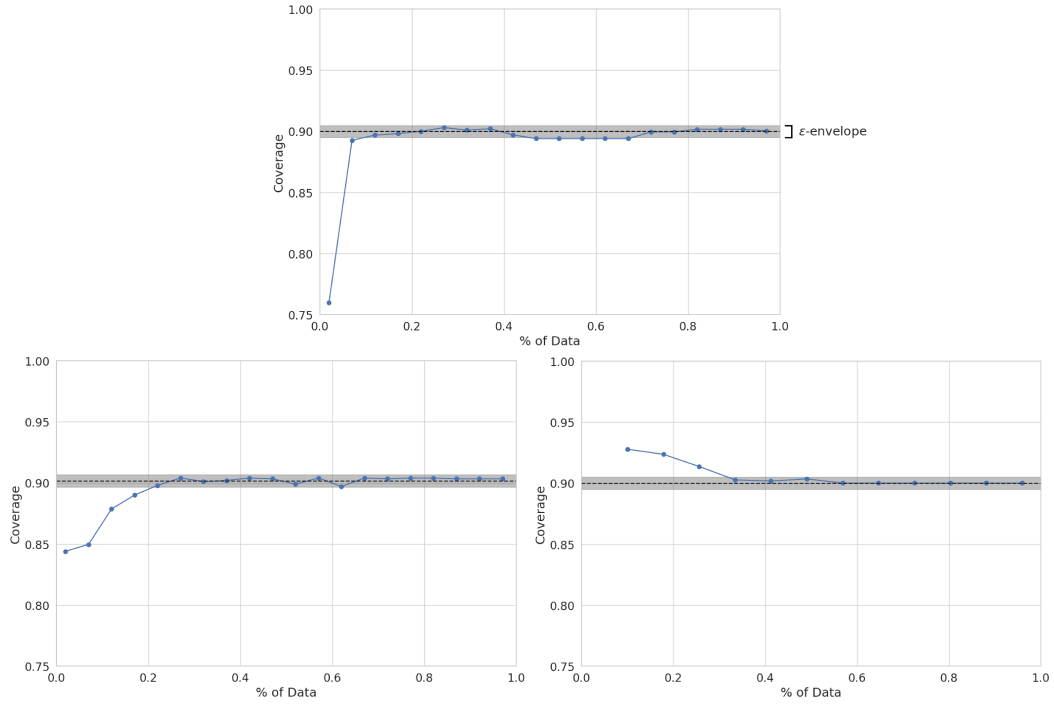


Figure 8: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 90% target. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Top Left) SST2, (Bottom Left), CrossNER, (Bottom Right) FewRel.

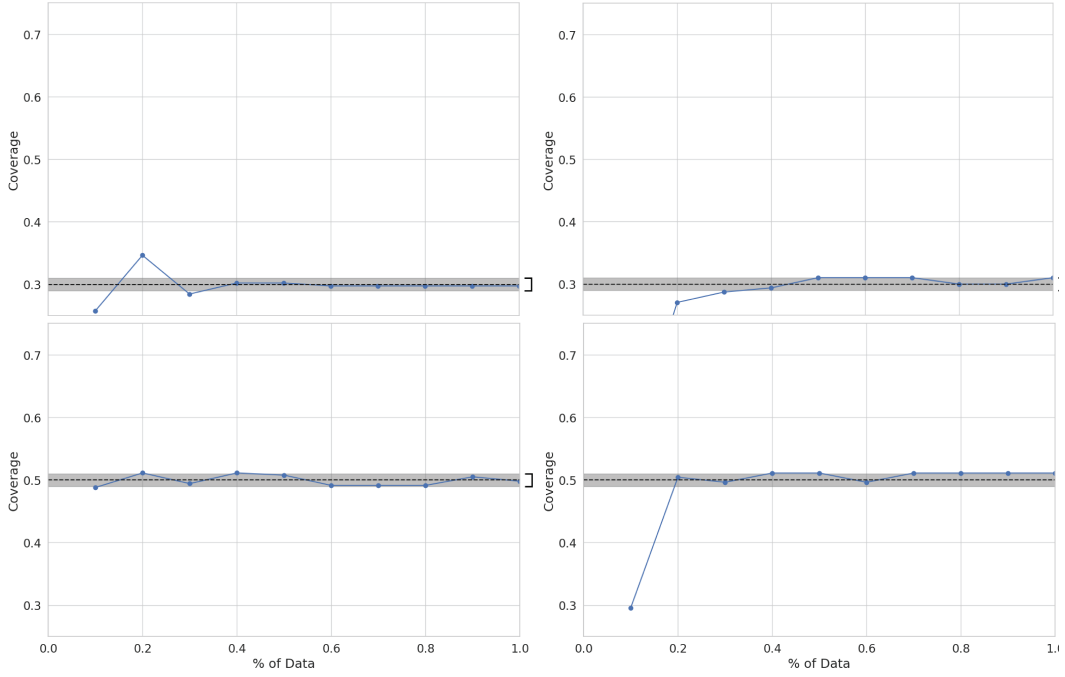


Figure 9: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 30% and 50% targets. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Left) SST2, and (Right) CrossNER.

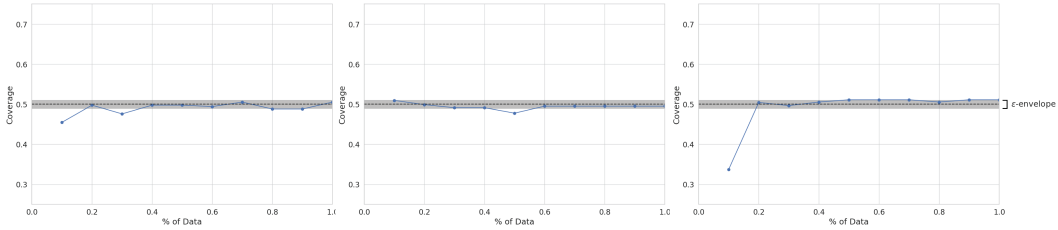


Figure 10: Coverage transfer from subsample to full dataset. Each point corresponds to a threshold τ^* optimized on a random subset with max degree constraint of a given size and evaluated for coverage on the full dataset. The gray band denotes a small tolerance range around the 50% target. Results show threshold transfer achieves accurate and stable coverage across various dataset sizes. (Left) SST2, (Middle) FewRel and (Right) CrossNER.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Yes. Methodology's validity is demonstrated in Section 4, with demonstration of its performance in Section 5.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, limitations as compared to other methods discussed in related work, as well as Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Main theoretical claim of Section 3 is proven on page 5.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, all methods are clearly delineated and source codes provided as supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All codes, random seeds, and notebooks to run are provided as supplementary. Codes are well documented and easy to run.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, ablation and study of optimal parameters discussed in main text. Random seeds for direct replication of model training provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Results averaged with error reported.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Compute resources discussed in Section 3.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Anonymity preserved.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- 838 • If this information is not available online, the authors are encouraged to reach out to
839 the asset’s creators.

840 **13. New assets**

841 Question: Are new assets introduced in the paper well documented and is the documentation
842 provided alongside the assets?

843 Answer: [NA]

844 Justification:

845 Guidelines:

- 846 • The answer NA means that the paper does not release new assets.
- 847 • Researchers should communicate the details of the dataset/code/model as part of their
848 submissions via structured templates. This includes details about training, license,
849 limitations, etc.
- 850 • The paper should discuss whether and how consent was obtained from people whose
851 asset is used.
- 852 • At submission time, remember to anonymize your assets (if applicable). You can either
853 create an anonymized URL or include an anonymized zip file.

854 **14. Crowdsourcing and research with human subjects**

855 Question: For crowdsourcing experiments and research with human subjects, does the paper
856 include the full text of instructions given to participants and screenshots, if applicable, as
857 well as details about compensation (if any)?

858 Answer: [NA]

859 Justification:

860 Guidelines:

- 861 • The answer NA means that the paper does not involve crowdsourcing nor research with
862 human subjects.
- 863 • Including this information in the supplemental material is fine, but if the main contribu-
864 tion of the paper involves human subjects, then as much detail as possible should be
865 included in the main paper.
- 866 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
867 or other labor should be paid at least the minimum wage in the country of the data
868 collector.

869 **15. Institutional review board (IRB) approvals or equivalent for research with human**
870 **subjects**

871 Question: Does the paper describe potential risks incurred by study participants, whether
872 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
873 approvals (or an equivalent approval/review based on the requirements of your country or
874 institution) were obtained?

875 Answer: [NA]

876 Justification:

877 Guidelines:

- 878 • The answer NA means that the paper does not involve crowdsourcing nor research with
879 human subjects.
- 880 • Depending on the country in which research is conducted, IRB approval (or equivalent)
881 may be required for any human subjects research. If you obtained IRB approval, you
882 should clearly state this in the paper.
- 883 • We recognize that the procedures for this may vary significantly between institutions
884 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
885 guidelines for their institution.
- 886 • For initial submissions, do not include any information that would break anonymity (if
887 applicable), such as the institution conducting the review.

888 **16. Declaration of LLM usage**

889 Question: Does the paper describe the usage of LLMs if it is an important, original, or
890 non-standard component of the core methods in this research? Note that if the LLM is used
891 only for writing, editing, or formatting purposes and does not impact the core methodology,
892 scientific rigorousness, or originality of the research, declaration is not required.

893 Answer: [Yes]

894 Justification: LLMs only used editing the writing.

895 Guidelines:

- 896 • The answer NA means that the core method development in this research does not
897 involve LLMs as any important, original, or non-standard components.
- 898 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
899 for what should or should not be described.