FuseGen: PLM Fusion for Data-generation based Zero-shot Learning

Tianyuan Zou[1], Yang Liu[1,2,†], Peng Li[1,2,†], Jianqing Zhang[1,3], Jingjing Liu[1], Ya-Qin Zhang[1]
[1]Institute for AI Industry Research (AIR), Tsinghua University,
[2]Shanghai Artificial Intelligence Laboratory, [3]Shanghai Jiao Tong University
[†]**Correspondence:** liuy03@air.tsinghua.edu.cn, lipeng@air.tsinghua.edu.cn

Abstract

Data-generation based zero-shot learning, although effective in training Small Task-specific Models (STMs) via synthetic datasets generated by Pre-trained Language Models (PLMs), is often limited by the low quality of such synthetic datasets. Previous solutions have primarily focused on single PLM settings, where synthetic datasets are typically restricted to specific sub-spaces and often deviate from real-world distributions, leading to severe distribution bias. To mitigate such bias, we propose FuseGen, a novel data-generation based zero-shot learning framework that introduces a new criteria for subset selection from synthetic datasets via utilizing multiple PLMs and trained STMs. The chosen subset provides in-context feedback to each PLM, enhancing dataset quality through iterative data generation. Trained STMs are then used for sample re-weighting as well, further improving data quality. Extensive experiments across diverse tasks demonstrate that FuseGen substantially outperforms existing methods, highly effective in boosting STM performance in a PLM-agnostic way.[1]

1 Introduction

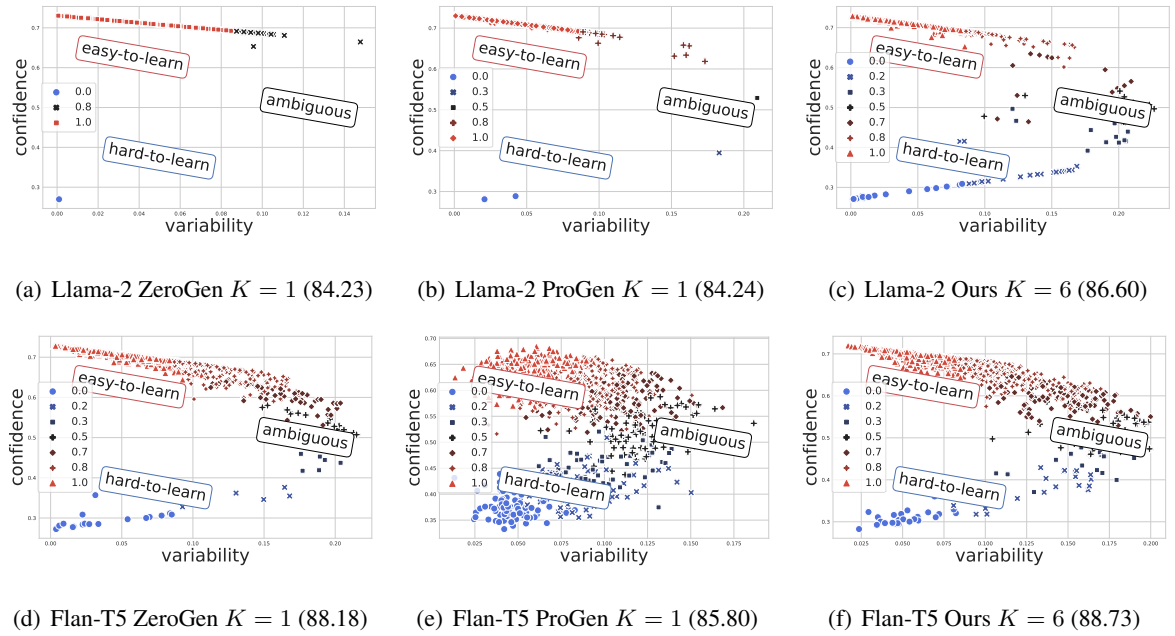
Despite the prevalence of powerful Pre-trained Language Models (PLMs) (Achiam et al., 2023; Team et al., 2023; Devlin et al., 2019) such as GPT-4, Small Task-specific Models (STMs) are indispensable due to their compact size and efficiency, especially for resource-constrained environments (Bommasani et al., 2021). To compensate for the scarcity of high-quality training data, synthetic data generated by PLMs has been widely applied for STM training (Ye et al., 2022a; Wang et al., 2023). In particular, *data-generation based zero-shot learning* (Ye et al., 2022a; Meng et al., 2022; Gao et al., 2023; Ye et al., 2022b) trains STM using the dataset synthesized by one PLM through task-related label-descriptive prompts, requiring only the task name (*e.g.* movie review sentiment analysis) and label categories (*e.g.* positive/negative). This zero-shot trained STM is significantly smaller than the original PLM with comparable performance (Ye et al., 2022a), thus is particularly advantageous for domains with limited computational resources (*e.g.* on mobile devices) or strict data privacy constraints (*e.g.* in finance applications).

However, the long-standing low-quality issue of synthetic data impedes the practical application of STMs to a wider range (Gao et al., 2023; Ye et al., 2022b). Previous works on improving synthetic data quality mainly focus on enhancing data diversity (Fan et al., 2018; Holtzman et al., 2020; Su and Collier, 2022; Yu et al., 2024), reducing redundancy (Bolón-Canedo et al., 2013; Deng et al., 2023), and implementing data-importance-guided in-context feedback (Ye et al., 2022b) or sample re-weighting (Gao et al., 2023). Despite notable advancements, they primarily rely on one single PLM as source, inevitably overlooking the inherent distribution biases of synthetic datasets.

To thoroughly investigate these biases and their impact on STM performance, we conduct two pilot studies. As illustrated in Figure 1, we use the dataset cartography approach (Swayamdipta et al., 2020) to plot the cartography of synthetic datasets given by different PLMs. Dataset samples are categorized into *easy-to-learn* (marked in red), *ambiguous* (marked in black) and *hard-to-learn* (marked in blue) based on their confidence and variability, defined as the mean and standard deviation of model probabilities for their labels across training epochs. Since *easy-to-learn* samples aid convergence and *ambiguous* samples are vital for boosting performance (Swayamdipta et al., 2020), an ideal dataset should predominantly contain diverse *easy-to-learn* and *ambiguous* samples, with fewer *hard-to-learn* samples which are often mislabeled (Swayamdipta

[1]The code is available at https://github.com/LindaLydia/FuseGen.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2172–2190
November 12-16, 2024



(a) Llama-2 ZeroGen $K = 1$ (84.23) (b) Llama-2 ProGen $K = 1$ (84.24) (c) Llama-2 Ours $K = 6$ (86.60)

(d) Flan-T5 ZeroGen $K = 1$ (88.18) (e) Flan-T5 ProGen $K = 1$ (85.80) (f) Flan-T5 Ours $K = 6$ (88.73)

Figure 1: Synthetic dataset cartography (Swayamdipta et al., 2020) using $1,000$ samples generated by Llama-2 and Flan-T5 for movie review semantic analysis. ZeroGen (Ye et al., 2022a) uses zero-shot prompt for generation, while ProGen (Ye et al., 2022b) and FuseGen (Ours) use few-shot prompt with feedback, with ProGen relying on a single PLM and FuseGen leveraging multiple PLMs. K is the number of PLMs. Numbers within parentheses are the results of STM trained with Self-boosting Weight Adjustment (see Section 3.4) and evaluated over IMDb (Maas et al., 2011) dataset. Results for more PLMs are provided in Figure 8 in Appendix C.1.

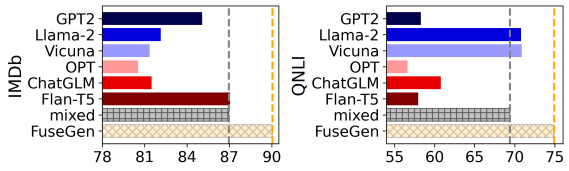


Figure 2: Performance of STM trained using $6,000$ synthetic data samples generated by various PLMs. "mixed" uses a dataset comprising $6,000$ total samples given by the 6 listed PLMs ($1,000$ samples per PLM). "FuseGen" (Ours) uses the 6 listed PLMs and $6,000$ samples.

et al., 2020). This composition of diverse samples promises better STM performance. In a second study, we provide the comparison between STMs trained with different datasets that vary in sources and generation methods, as illustrated in Figure 2.

These visualization analyses reveal three key observations: (1) Synthetic datasets from different PLMs exhibit significant distribution biases. For example, Figures 1(a) and 1(d) show that the zero-shot synthetic dataset produced by Llama-2 (Touvron et al., 2023) primarily includes *easy-to-learn* samples, whereas that of Flan-T5 (Chung et al., 2022) contains a more balanced mixture of all 3 categories. (2) Distribution biases are difficult to overcome by only relying on a single PLM. ProGen (Ye et al., 2022b), an advanced single-PLM generation method, only slightly improves the ratio of *easy-to-learn* and *ambiguous* samples (Figure 1(b)), while adversely increases the proportion of *hard-to-learn* samples in some cases (Figure 1(e)). (3) Simply mixing samples from multiple PLMs is ineffective. As demonstrated in Figure 2, plainly combining data generated by multiple PLMs improves STM performance compared to most single-PLM cases, but is still worse than the best single PLM.

To tackle these challenges, we propose FuseGen, a smart data-generation based zero-shot learning framework that mitigates inherent dataset distribution bias by harnessing the diversity of a PLM cluster. In FuseGen, given a specific task and its label categories, synthetic datasets are initially generated by various PLMs in a zero-shot manner, which are then used to train their respective STMs. To alleviate distribution bias, FuseGen selects superior samples generated by multiple PLMs as shared in-context feedback, and prompts each PLM to accumulate higher-quality data iteratively. To select relevant in-context samples, FuseGen pivots on an efficient cross-model criteria that considers both dataset composition and individual sample importance. To mitigate the negative impact of poor-quality samples, FuseGen further uses a self-boosting method to dynamically adjust sample

weights to optimize STM in training. As demonstrated in Figures 1(c), 1(f) and 2, with these novel techniques, FuseGen effectively reduces distribution biases and achieves better STM performance than state-of-the-art methods.

Our contributions can be summarized as follows:

(1) We introduce a novel data-generation based zero-shot learning framework, FuseGen, which collaboratively leverages multiple PLMs to generate higher-quality synthetic dataset without incurring any additional queries to PLMs themselves. Further, FuseGen neither requires access to nor fine-tunes the parameters of PLMs.

(2) We propose a novel cross-model criteria for selecting in-context samples, which then serves as generation feedback, and a self-boosting method for improving STM performance.

(3) Extensive evaluations on 9 Natural Language Inference (NLI), Understanding (NLU) and Generation (NLG) tasks with 6 open-source and 2 closed-source PLMs demonstrate the consistent superiority of FuseGen over single-PLM methods. This PLM-agnostic nature eliminates the reliance on specific PLMs for downstream tasks.

2 Related Work

Data-generation based Zero-shot Learning. A recent line of research focuses on exploiting the data generation capabilities of PLMs (Ye et al., 2022a; Meng et al., 2022; Ye et al., 2022b; Gao et al., 2023) to generate synthetic data for training a target model (Meng et al., 2022; Ye et al., 2022a,b; Gao et al., 2023). The dataset is generated by prompting PLM with task and label descriptions. A critical challenge for this approach is that generated datasets often contain low-quality samples. Recent attempts to address this include techniques to enhance dataset diversity (*e.g.* Top-k sampling (Fan et al., 2018), nucleus sampling (Holtzman et al., 2020), diversely attributed prompts (Yu et al., 2024), and contrastive search decoding (Su and Collier, 2022)). Additionally, feature selection (Bolón-Canedo et al., 2013) helps eliminate redundant information within the dataset. Finally, methods like progressive generation with in-context feedback (Ye et al., 2022b) and sample re-weighting (Ye et al., 2022b) focus on identifying and amplifying the influence of high-quality samples. Despite significant progress, existing studies often overlook the inherent data distribution bias in synthetic datasets generated by a single PLM. In contrast, our work explores avoiding this bias by leveraging diverse multiple PLMs.

Fusion of PLMs. Recent studies suggest that it is possible to combine the capabilities of multiple PLMs to obtain a model with stronger performance (Wan et al., 2024a,b; Li et al., 2024). Existing PLM knowledge-fusion techniques can be grouped into *training-time fusion* and *test-time fusion* (Mavromatis et al., 2024). *Training-time fusion* methods (Wan et al., 2024a,b) fuse PLMs' token-level predictions produced during training time to fine-tune a target PLM, requiring abundant computational resources. *Test-time fusion* methods do not fine-tune PLMs, but utilize methods such as logits averaging (Mavromatis et al., 2024) and majority voting (Li et al., 2024) to fuse the knowledge of PLMs at test time. In addition, interactions and collaborations among PLM agents (Liu et al., 2024; Du et al., 2023) have been investigated.

All these works demonstrate that collaboration among diverse PLMs helps. However, all existing works require direct access to training samples, which means they are not applicable to the setting of data-generation based zero-shot learning, the problem we aim to solve.

3 FuseGen

3.1 Preliminaries

In *data-generation based zero-shot learning* (Ye et al., 2022a; Gao et al., 2023) with *a single PLM*, given a downstream task like text classification, a PLM $\mathcal{P}$ with parameter $\Phi_{\mathcal{P}}$ first generates a synthetic dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of size N. This is accomplished by using a proper task-related label-descriptive prompt $\mathcal{T}(\cdot)$ (examples are provided in Appendix A.1) as follows:

$$\mathbf{x}_i \sim \mathcal{P}(\cdot|\mathcal{T}(y_i), \Phi_{\mathcal{P}})\,. \tag{1}$$

$\mathcal{D}$ is then used to train an STM m with the following training objective:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(m(\mathbf{x}_i), y_i), \tag{2}$$

where ℓ is a common loss function, *e.g.* cross-entropy loss.

3.2 FuseGen Architecture Overview

Different from previous works, we focus on *multi-PLM setting* and propose FuseGen. The FuseGen workflow is illustrated in Figure 3. In a nutshell, FuseGen consists of two main components: Cross-model Dataset Generation (CDG) (Section 3.3)

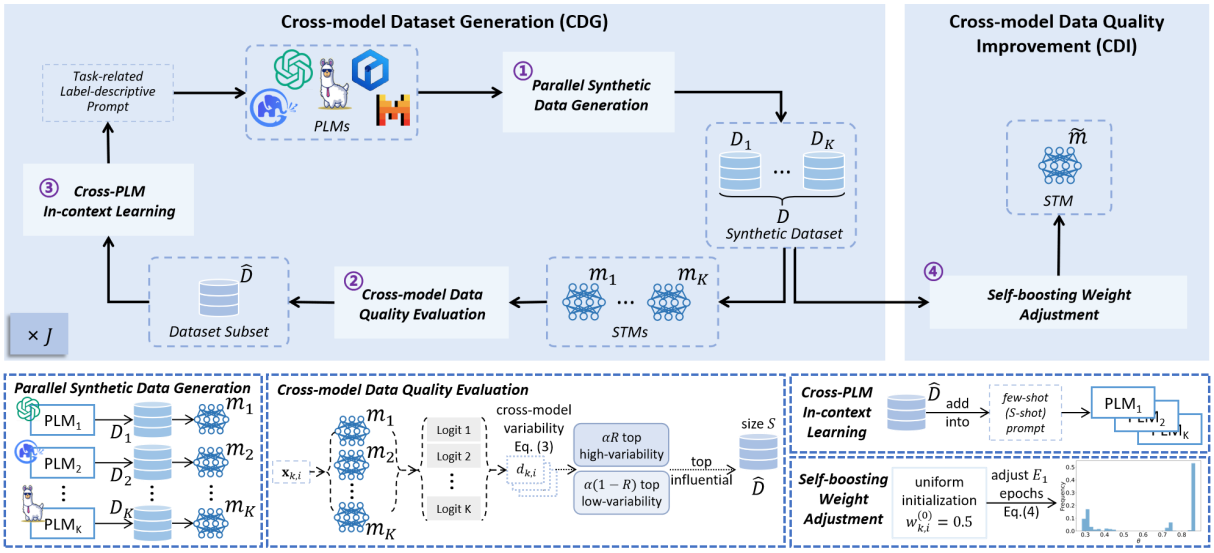


Figure 3: Illustrated Workflow of FuseGen with two components: Cross-model Data Generation (CDG) and Cross-model Data Quality Improvement (CDI). CDG iteratively executes parallel synthetic data generation, cross-model data quality evaluation and cross-PLM in-context learning. CDI implements self-boosting weight adjustment for sample-reweighted training of STM.

and Cross-model Data Quality Improvement (CDI) (Section 3.4). For CDG, given a fixed number of samples to generate in total, PLMs progressively generate datasets for multiple rounds, each round using an improved subset of samples generated from previous rounds as in-context examples. This is realized in three steps: (1) *Parallel Synthetic Data Generation*: each PLM generates its own dataset and trains a respective STM. (2) *Cross-model Data Quality Evaluation*: the quality of generated samples is evaluated using a cross-PLM criteria to select a desirable subset. (3) *Cross-PLM In-context Learning*: the cross-PLM subsets are used as in-context examples to prompt PLMs to generate new datasets. Step (1) is then repeated. After the required number of samples is reached, we perform CDI which re-weights samples with a self-boosting strategy. Algorithm 1 provides an overview of the above steps, with each function detailed in Appendix B.

Algorithm 1 FuseGen

Input:
K PLMs, empty synthetic dataset $\{\mathcal{D}_k \leftarrow \emptyset\}_{k=1}^{K}$, target number of synthetic samples N for each PLM, sample selection hyper-parameter α, R, S, number of feedback steps J taken to obtain in total N synthetic samples, random initialized STM $m_{(0)}$, test dataset of downstream task $\mathcal{A}$, initialized sample weights $\left\{\{w_{k,i}^{(0)}\}_{i=1}^{N}\right\}_{k=1}^{K}$, learning rate η, number of weight adjustment epochs E_1, number of STM training epochs E_2.
Output: STM $\tilde{m}$ that obtains the effectively aggregated knowledge from K PLMs.

Initialize in-context feedback samples $\hat{\mathcal{D}} \leftarrow \emptyset$.
for $j = 0$ **to** J **do**
  for $k = 1$ **to** K **in parallel do**
    $\mathcal{D}_k \leftarrow$ `S_AccumulativeSynDataGeneration`$(\mathcal{D}_k, \hat{\mathcal{D}}, N, J, j)$.
    $m_k \leftarrow$ `S_STMTraining`$(\mathcal{D}_k, m_{(0)}, E_2)$.
  end for
  $\tilde{m} \leftarrow$ `S_STMTraining`$(\cup_{k=1}^{K}\mathcal{D}_k, m_{(0)}, E_2)$.
  $\hat{\mathcal{D}} \leftarrow$ `C_SampleSelection`$(\cup_{k=1}^{K}\mathcal{D}_k, \{m_k\}_{k=1}^{K}, \tilde{m}, \alpha, R, S)$.
end for
$\tilde{m} \leftarrow$ `S_WeightAdjustSTMTraining`$(\cup_{k=1}^{K}\mathcal{D}_k, m_{(0)}, \cup_{k=1}^{K}\left\{\{w_{k,i}^{(0)}\}_{i=1}^{N}\right\}, E_1, E_2)$.

3.3 Cross-model Dataset Generation

In FuseGen, each PLM iteratively generates a total of N samples across $J+1$ rounds, incorporating feedback from STMs after each of the first J rounds. In each round, a total of $\frac{N}{J+1}$ samples are generated using the accumulated knowledge of multiple PLMs from previous rounds as feedback. Specifically, the following steps are taken:

Parallel Synthetic Dataset Generation. In each round, each of K PLMs (denoted as $\{\mathcal{P}_k\}_{k=1}^{K}$) generates a synthetic dataset $\mathcal{D}_k = \{(\mathbf{x}_{k,i}, y_{k,i})\}_{i=1}^{\frac{N}{J+1}}$ of size $\frac{N}{J+1}$ in parallel with the same task-related label-descriptive prompt $\mathcal{T}(\cdot)$ as described in Section 3. Each dataset is then used to train a separate STM m_k following Equation (2). This step produces K separate STMs and K synthetic datasets.

Cross-model Data Quality Evaluation. In this step, we aim to select a desirable subset from $\mathcal{D} = \bigcup_{k=1}^{K}\mathcal{D}_k$ to guide data generation. To accomplish

this goal, we utilize the knowledge of trained STMs at hand and develop a simple yet efficient criteria for data-quality evaluation.

As discussed in Section 1 , *easy-to-learn* samples of low-variability and *ambiguous* samples of high-variability are both vital for constructing a desirable dataset, valuable for training convergence and model generalization ability, respectively. Inspired by this, we first use cross-model variability $d_{k,i}$ to categorize each sample, defined as:

$$d_{k,i} = \mathrm{STD}(p_{1,k,i}[y_{k,i}], ..., p_{k',k,i}[y_{k,i}], ..., p_{K,k,i}[y_{k,i}]) \tag{3}$$

where $p_{k',k,i}[y_{k,i}]$ denotes STM model $m_{k'}$'s predicted probability of synthetic label $y_{k,i}$ on that sample $\mathbf{x}_{k,i}$, and STD represents standard deviation[2]. To prompt the generation of a dataset that includes both low-variability (low $d_{k,i}$) and high-variability (high $d_{k,i}$) data, we select a small number of candidates (of size $R \ll N$) comprised of αR top high-variability and $(1-\alpha)R$ top low-variability samples, where α is a hyper-parameter that controls the percentage of high-variability samples. The goal here is to efficiently select a smaller and more manageable subset from a large set of candidates. The selected subset can then be processed by more computationally intensive ranking. To further identify samples that are vital for training, we train an STM $\tilde{m}$ using $\mathcal{D}$ and leverage the noise-resistant influence function proposed in ProGen (Ye et al., 2022b) to select the top-S influential samples from the R candidate samples ($S < R$). Our results validate that these selected samples originate from various PLMs (See Appendix C.4.)

Cross-PLM In-context Learning. After selecting S in-context samples (denoted as $\hat{\mathcal{D}}$), we add them to the original prompt $\mathcal{T}(\cdot)$, resulting in $\mathcal{T}(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_S; \cdot)$ (see examples in Appendix A.1). We then send the feedback prompt to each PLM to generate $\frac{N}{J+1}$ new samples following $\mathbf{x}_{k,i} \sim \mathcal{P}_k(\cdot|\mathcal{T}(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_S; y_{k,i}), \Phi_{\mathcal{P}_k})$, where $\Phi_{\mathcal{P}_k}$ denotes the parameter of $\mathcal{P}_k$. In this way, PLMs can learn from each other and generate datasets with improved quality.

3.4 Cross-model Data Quality Improvement

After CDG process that improves overall data distribution, we perform one last step of re-weighting samples by their quality, determined by a **Self-boosting Weight Adjustment (SWA)** approach.

As *hard-to-learn* samples (refer to Figures 1(c) and 1(f)) and low-quality samples (*e.g.* meaningless or irrelevant) still exist post-CDG, we down-weight these samples in each training round of the final STM $\tilde{m}$. Specifically, a weight $w_{k,i}$ (uniformly initialized as 0.5) is assigned to each sample in $\mathcal{D} = \{\{(\mathbf{x}_{k,i}, y_{k,i})\}_{i=1}^{N}\}_{k=1}^{K}$. At the e_1-th weight-adjustment round of $\tilde{m}$, we update $w_{k,i}$ using the following boosting strategy inspired by TrAdaBoost (Dai et al., 2007):

$$w_{k,i}^{(e_1+1)} = w_{k,i}^{(e_1)} \beta^{-\mathrm{error}_{k,i}(1-\mathrm{correct}_{k,i})}, \quad k = 1, \ldots, K,\ i = 1, \ldots, N, \tag{4}$$

where $\beta = \frac{1}{1+\sqrt{\frac{2\ln(NK)}{E_1}}} > 0$ is a constant value for weight adjustment, E_1 is the number of total epochs for weight adjustment, $\mathrm{error}_{k,i} = 1 - p_{k,i}[y_{k,i}]$ is the prediction error of $\tilde{m}$ on data sample $\mathbf{x}_{k,i}$, and $\mathrm{correct}_{k,i} = 1$ if $\tilde{m}$ predicts sample $\mathbf{x}_{k,i}$ correctly, otherwise $\mathrm{correct}_{k,i} = 0$. Normalization is applied afterwards to guarantee that $\sum_{k=1}^{K}\sum_{i=1}^{N} w_{k,i}^{(e_1)} = 0.5NK$. After normalization, $w_{k,i}$ for correctly inferred samples increases while that for wrongly inferred samples decreases. A new STM is trained from scratch with the new weights after each adjustment step. Training details are provided in Algorithms 1 and 2. With SWA, the training objective for $\tilde{m}$ using all synthetic data $\mathcal{D}$ is given by:

$$\mathcal{L} = \sum_{k=1}^{K}\sum_{i=1}^{N} w_{k,i} \cdot \ell(\tilde{m}(\mathbf{x}_{k,i}), y_{k,i}) \,. \tag{5}$$

Unlike SunGen (Gao et al., 2023), which utilizes a self-guided sample re-weighting method with bi-level SGD optimization to enhance its STM performance, our SWA achieves comparable STM performance without requiring this computationally expensive optimization step (see Section 4 and Appendix C.6). This translates to a significantly smaller computational cost.

4 Experiments

4.1 Experimental Settings

Models. In our experiments, we evaluate on 6 open-source PLMs: GPT-2-xl (GPT-2) (Radford et al., 2019), Llama-2-7b-chat-hf (Llama-2) (Touvron et al., 2023), Vicuna-7b-1.5v (Vicuna) (Chiang et al., 2023), OPT-6.7b (OPT) (Zhang et al., 2022),

[2]Different from Swayamdipta et al. (2020), we do not include confidence (*i.e.* mean of predicted probability in our criteria, as the synthetic label is not used for in-context samples (see Appendix A.1 for in-context sample examples).

	IMDb						SST-2					
	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
ZeroGen ♠	$85.07_{\pm 1.49}$	$82.14_{\pm 0.83}$	$81.36_{\pm 2.98}$	$80.54_{\pm 3.63}$	$81.49_{\pm 3.33}$	$87.06_{\pm 0.64}$	$80.99_{\pm 2.25}$	$79.47_{\pm 3.65}$	$82.33_{\pm 3.16}$	$82.00_{\pm 2.29}$	$86.49_{\pm 0.13}$	$81.88_{\pm 1.86}$
SunGen ♠	$86.94_{\pm 0.99}$	$86.59_{\pm 1.20}$	$84.93_{\pm 1.17}$	$85.21_{\pm 0.64}$	$84.76_{\pm 2.67}$	$\underline{89.79}_{\pm 1.33}$	$83.45_{\pm 0.79}$	$84.30_{\pm 0.28}$	$84.04_{\pm 0.30}$	$83.49_{\pm 1.22}$	$\underline{87.18}_{\pm 0.08}$	$83.53_{\pm 0.86}$
ProGen ♠	$85.68_{\pm 2.68}$	$84.33_{\pm 0.26}$	$82.14_{\pm 2.30}$	$85.57_{\pm 0.19}$	$87.41_{\pm 1.01}$	$88.00_{\pm 0.53}$	$83.60_{\pm 1.54}$	$79.53_{\pm 1.72}$	$82.53_{\pm 1.69}$	$82.78_{\pm 0.44}$	$86.64_{\pm 1.03}$	$83.17_{\pm 1.12}$
FuseGen (Ours)	**90.06**$_{\pm 0.30}$						**87.51**$_{\pm 0.23}$					
	Yelp						QNLI					
	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
ZeroGen ♠	$89.73_{\pm 0.43}$	$89.74_{\pm 0.76}$	$85.67_{\pm 3.21}$	$87.13_{\pm 3.36}$	$82.00_{\pm 3.32}$	$92.41_{\pm 0.48}$	$58.30_{\pm 1.35}$	$70.79_{\pm 1.72}$	$70.88_{\pm 0.22}$	$56.64_{\pm 0.63}$	$60.77_{\pm 0.18}$	$57.95_{\pm 1.84}$
SunGen ♠	$91.85_{\pm 0.56}$	$89.30_{\pm 0.55}$	$89.06_{\pm 0.88}$	$91.22_{\pm 0.38}$	$88.86_{\pm 1.78}$	$\underline{93.13}_{\pm 0.31}$	$62.26_{\pm 0.63}$	$74.20_{\pm 0.13}$	$\underline{74.35}_{\pm 0.38}$	$57.50_{\pm 0.88}$	$65.64_{\pm 1.04}$	$58.21_{\pm 1.17}$
ProGen ♠	$91.26_{\pm 2.88}$	$89.82_{\pm 1.59}$	$88.55_{\pm 0.18}$	$89.00_{\pm 0.83}$	$88.81_{\pm 1.69}$	$91.71_{\pm 0.58}$	$58.38_{\pm 1.78}$	$69.56_{\pm 0.79}$	$70.29_{\pm 1.70}$	$57.46_{\pm 1.46}$	$61.08_{\pm 0.38}$	$69.44_{\pm 0.31}$
FuseGen (Ours)	**93.47**$_{\pm 0.32}$						**74.92**$_{\pm 0.36}$					
	MNLI-matched						MNLI-mismatched					
	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
ZeroGen ♠	$41.99_{\pm 1.63}$	$48.52_{\pm 1.12}$	$45.87_{\pm 0.30}$	$36.16_{\pm 0.18}$	$32.65_{\pm 0.07}$	$47.37_{\pm 1.81}$	$46.38_{\pm 1.93}$	$50.04_{\pm 1.27}$	$48.10_{\pm 0.97}$	$36.74_{\pm 0.47}$	$33.00_{\pm 0.09}$	$49.95_{\pm 1.17}$
SunGen ♠	$44.66_{\pm 0.35}$	$\underline{49.43}_{\pm 0.04}$	$46.27_{\pm 0.65}$	$37.44_{\pm 0.12}$	$32.71_{\pm 0.07}$	$49.04_{\pm 0.70}$	$47.45_{\pm 0.42}$	$\underline{51.67}_{\pm 0.27}$	$48.63_{\pm 0.55}$	$38.35_{\pm 0.31}$	$33.02_{\pm 0.06}$	$51.66_{\pm 0.67}$
ProGen ♠	$43.35_{\pm 1.28}$	$48.69_{\pm 1.51}$	$47.50_{\pm 0.99}$	$36.79_{\pm 2.48}$	$32.81_{\pm 0.08}$	$48.56_{\pm 1.10}$	$46.57_{\pm 1.84}$	$50.57_{\pm 1.19}$	$49.65_{\pm 1.48}$	$40.27_{\pm 1.55}$	$33.01_{\pm 0.10}$	$50.24_{\pm 1.50}$
FuseGen (Ours)	**49.76**$_{\pm 0.55}$						**51.70**$_{\pm 0.50}$					
	AgNews						SQuAD					
	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
ZeroGen ♠	$77.86_{\pm 3.31}$	$83.40_{\pm 0.07}$	$81.25_{\pm 2.25}$	$84.81_{\pm 0.33}$	$83.17_{\pm 0.33}$	$81.87_{\pm 2.91}$	$9.32_{\pm 0.99}$	$7.37_{\pm 1.46}$	$5.05_{\pm 0.05}$	$7.72_{\pm 0.60}$	$8.60_{\pm 0.79}$	$5.95_{\pm 0.98}$
SunGen ♠	$80.94_{\pm 0.33}$	$84.44_{\pm 0.31}$	$82.50_{\pm 2.90}$	$\underline{85.68}_{\pm 0.03}$	$84.12_{\pm 0.59}$	$85.57_{\pm 1.69}$	$\underline{9.66}_{\pm 1.20}$	$7.55_{\pm 1.62}$	$5.09_{\pm 0.05}$	$8.92_{\pm 0.85}$	$8.60_{\pm 0.69}$	$6.97_{\pm 1.44}$
ProGen ♠	$78.68_{\pm 1.91}$	$83.93_{\pm 1.23}$	$81.46_{\pm 1.57}$	$85.66_{\pm 0.97}$	$84.74_{\pm 0.43}$	$84.59_{\pm 0.37}$	$8.08_{\pm 0.58}$	$7.42_{\pm 1.39}$	$6.96_{\pm 0.79}$	$7.51_{\pm 1.83}$	$9.43_{\pm 0.55}$	$6.60_{\pm 1.16}$
FuseGen (Ours)	**86.89**$_{\pm 0.23}$						**10.09**$_{\pm 0.64}$					

Table 1: Comparison of FuseGen and baselines with $K = 6$. Methods marked by ♠ are single-PLM methods. $\tilde{m}_G$, $\tilde{m}_L$, $\tilde{m}_V$, $\tilde{m}_O$, $\tilde{m}_C$, $\tilde{m}_F$ represents the final STM performance with single PLM GPT-2, Llama-2, Vicuna, OPT, ChatGLM3 and Flan-T5, respectively. Best result is marked as **bold**, and the second best is marked with underline.

	MarkedNews					
	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
ZeroGen ♠	$77.16_{\pm 0.71}$	$74.49_{\pm 2.07}$	$74.10_{\pm 0.47}$	$77.80_{\pm 0.76}$	$80.33_{\pm 1.23}$	$76.12_{\pm 1.27}$
SunGen ♠	$78.01_{\pm 0.61}$	$76.75_{\pm 0.82}$	$76.39_{\pm 1.19}$	$78.15_{\pm 0.27}$	$82.16_{\pm 0.60}$	$77.85_{\pm 0.51}$
ProGen ♠	$77.17_{\pm 2.24}$	$76.51_{\pm 2.00}$	$76.14_{\pm 1.59}$	$77.93_{\pm 2.12}$	$\underline{82.70}_{\pm 0.73}$	$78.75_{\pm 1.09}$
FuseGen (Ours)	**83.85**$_{\pm 0.48}$					

Table 2: Results of FuseGen and baselines with $K = 6, N = 1,000$ using Markednews dataset. Best result is marked as **bold**, and the second best is marked with underline.

ChatGLM3-6b-base (ChatGLM3) (Du et al., 2022) and Flan-T5-xl (Flan-T5) (Chung et al., 2022). 2 closed-source PLMs are also used for generating synthetic datasets: GPT-3.5-turbo-instruct (GPT-3.5) (OpenAI, 2021) and GPT-4-turbo-preview (GPT-4) (OpenAI, 2023). For the choice of STM, we use bert-base-uncased (BERT), a pre-trained model, to perform downstream classification tasks. The trained STM is evaluated over a real-world human-annotated dataset (test dataset) $\mathcal{A}$ that is never used during training.

Datasets. We select 8 well-developed datasets to evaluate our framework: 1) IMDb (Maas et al., 2011) and SST-2 (Socher et al., 2013; Wang et al., 2019) for movie review semantic analysis task, 2) Yelp-polarity (Zhang et al., 2015) for restaurant review semantic analysis task, 3) AgNews (Zhang et al., 2015) for news category classification task, 4) QNLI (Wang et al., 2019) for question-information entailment classification task, 5) MNLI (both matched and mismatched) (Williams et al., 2018) for sentence-pair relation classification task, 6) SQuAD (Rajpurkar et al., 2016) for question answering task. To test the effectiveness of FuseGen on unseen tasks, we further create a new dataset named MarkedNews from AgNews. MarkedNews categorizes articles containing the symbol "$" as "Money with $ included", and all other articles retain their original AgNews categories. This creates a new 5-class classification task: "World", "Sports", "Business", "Technology", and "Money with $ included". We adopt the original test dataset as $\mathcal{A}$ except for QNLI and MNLI, where ground-truth labels are unavailable. In these cases, we use the validation sets instead. The experiments run on A100-80G.

Baselines. We compare our framework with several existing data-generation based zero-shot learning methods, including 1) ZeroGen (Ye et al., 2022a) which directly trains an STM using the generated synthetic data, 2) SunGen (Gao et al., 2023) which recovers a robust synthetic dataset through sample-level weight optimization, and 3) ProGen (Ye et al., 2022b) which progressively generates data using self-given in-context feedback through prompt. To ensure a fair comparison, all methods generate the same number of samples. In other words, each single-PLM method produces a total of $N \times K$ samples.

Implementation Details. Unless otherwise stated, the following setting is applied: $N = 1,000$

synthetic data samples generated by each PLM are used for FuseGen; the BERT models (STMs) are trained with Adam optimizer with a learning rate of 2×10^{-5} and training epochs (E_2) of 3. When training STMs, weight adjustment is performed for 30 iterations ($E_1 = 30$). Each experiment is repeated 3 times using different random seeds, and averaged accuracy is reported. $\alpha = 0.5, R = 40, S = 8$ is used to select in-context samples for constructing feedback prompt, except for QNLI and MNLI datasets, where $R = 20, S = 4$ is used in order to fit the maximum input length of each PLM. $J = 4$ is used for iterative generation (both FuseGen and ProGen). For SunGen, 50 samples are used for sample-weight backward gradient estimation.

4.2 Main Results

Tables 1 and 2 summarizes the main results of our FuseGen framework and compared baseline methods. To ensure comprehensive evaluation, each single-PLM baseline method is evaluated using samples generated from each of the PLMs. F1 score is reported for SQuAD while classification accuracy (ACC) is reported for other datasets.

Open-source PLMs. Tables 1 and 2 show that FuseGen consistently outperforms all baselines using the same number of generated samples. Our method achieves up to 1.2% increase in STM performance over the best-performing single-PLM baseline, which exploits the optimal PLM for each task. SunGen performs consistently well among single-PLM baselines, but the ideal PLM varies by task. However, in zero-shot setting, where no task-specific samples are available, pre-selecting a PLM for optimal training performance is impractical. FuseGen is free from such pre-selection. Results for SQuAD with more synthetic samples are included in Appendix C.5.

Unseen Tasks. Evaluation results for FuseGen and baselines over our new dataset MarkedNews are shown in Table 2, with synthetic data generation prompts detailed in Appendix A.1. FuseGen outperforms all baselines consistently, demonstrating its ability to enhance downstream STM performance even when PLMs lack prior knowledge of the unseen classification task.

Closed-source PLMs. We also conduct experiments on the fusion of two popular closed-source models (GPT-3.5 and GPT-4) using QNLI dataset with $K = 2$. Results in Table 3 (each $\tilde{m}_k$ is trained with $6,000$ samples) demonstrate the superior performance of FuseGen compared to baselines.

	QNLI	
	$\tilde{m}_{GPT-3.5}$	$\tilde{m}_{GPT-4}$
ZeroGen ♠	$74.25_{\pm1.11}$	$72.11_{\pm2.97}$
SunGen ♠	$\underline{76.66}_{\pm0.84}$	$75.46_{\pm0.96}$
ProGen ♠	$74.84_{\pm1.09}$	$74.83_{\pm2.00}$
FuseGen (Ours)	**77.59**$_{\pm0.53}$	

Table 3: Comparison of FuseGen and baseline methods on closed-source PLMs with QNLI dataset and $K = 2$.

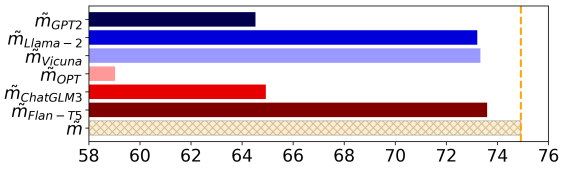


Figure 4: Comparison of FuseGen between using multi-PLM (last bar) and single-PLM with QNLI dataset.

FuseGen's consistent superiority across diverse tasks and models underscores its PLM-agnostic nature. This eliminates the need of relying on specific models for downstream tasks, making it a more flexible and efficient solution.

4.3 Ablation Study

4.3.1 Multi-PLM v.s. Single-PLM

We evaluate the impact of multi-PLM fusion by comparing FuseGen between using multi-PLM ($K = 6$) and single-PLM ($K = 1$). Results are provided in Figure 4. Since cross-model variability evaluation in CDG can not be performed for $K = 1$, random selection is applied here to select R candidate samples, whereas CDI is applied to both cases. Figure 4 shows that *multi-PLM collaboration is vital for further improving the quality of synthetic dataset, yielding better STM performance than relying on single-PLM*. Detailed results on more datasets are provided in Appendix C.7.

We further study the impact of K on the performance of FuseGen. Figure 5 shows the average and standard deviation (STD) of the performance of FuseGen with $K = 1, 2, 3, 4, 5$ across all C_6^K possible combinations. Each run is repeated with 3 different seeds and a constant total synthetic sample budget, $N \times K = 6,000$ is used for all the runs. These results demonstrate that, as K increases, the expectation of the final STM performance improves, while the randomness (STD) decreases. This indicates that FuseGen is able to mitigate the degree of randomness on the final performance by incorporating a greater number of

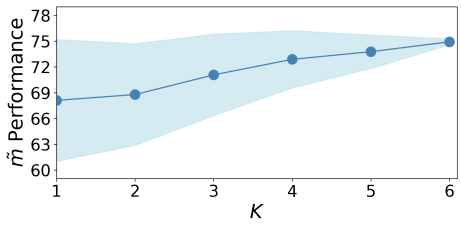


Figure 5: Change in performance of FuseGen with the change of K.

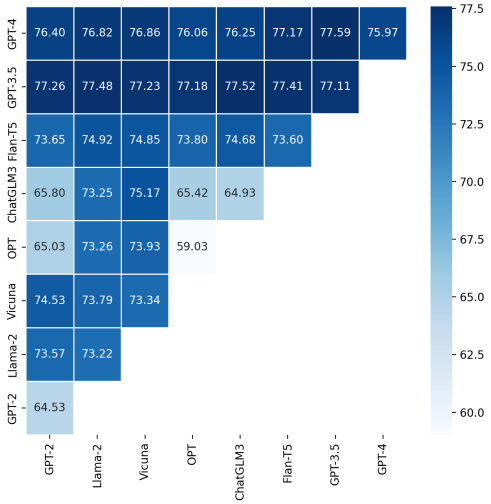


Figure 6: STM performance of FuseGen ($\tilde{m}$) with $K = 1$ (diagnose) and $K = 2$ (others) and $N \times K = 6,000$ using QNLI dataset.

PLMs in the collaboration.

4.3.2 Pair-wise PLM Fusion

We additionally perform experiments for every possible pairing of the 8 PLMs ($K = 2, N = 3,000$) to investigate the pair-wise collaboration between PLMs. Results are included in Figure 6. By comparing the pair-wise fusion results with single-PLM performance (diagnose in Figure 6), we show that even the strongest single PLMs, i.e. GPT-4 and GPT-3.5, benefit from FuseGen through collaboration with other (weaker) PLMs, resulting in enhanced STM performance. This highlights that FuseGen's enhancements are PLM-agnostic, requiring no prior knowledge of PLM performance. This flexibility is particularly important given the plethora of open-source and closed-source PLMs available today.

4.3.3 In-context Sample Selection

In-context sample selection is a critical component of the FuseGen framework, as it influences the quality of feedback from STMs to PLMs, which in turn affects the generation quality of PLMs. In this section, we compare various in-context sample selection strategies, including random selection, high-variability and low-variability selection. The latter two exclusively select top-R high-variability or low-variability samples, respectively. We also evaluate each strategy with and without fine-grained influence-based selection. The results are shown in Table 4. We also report the performance of each m_k trained with SWA using the corresponding $\mathcal{D}_k$ during the FuseGen process in Table 4. Our in-context sample selection strategy surpasses other alternatives consistently, not just in the final STM performance, but also for each intermediate small model m_k produced during FuseGen. This underscores *the efficacy of our selection approach* and *FuseGen's ability to produce higher-quality datasets for all PLMs involved.*

Variability Low	Variability High	Influence	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
Rand.		✗	52.47	67.48	65.90	50.52	56.68	67.66	72.89
✓	✗	✗	53.77	66.18	61.33	50.96	53.37	66.13	73.76
✗	✓	✗	54.98	65.48	60.76	49.79	54.28	65.47	73.81
✓	✓	✗	58.59	70.85	66.31	50.38	55.23	67.83	74.14
Rand.		✓	54.25	70.44	70.74	51.19	56.68	68.84	74.07
✓	✗	✓	54.00	70.07	67.75	51.12	55.70	66.49	74.08
✗	✓	✓	54.85	66.47	64.46	50.08	56.50	70.50	74.16
✓	✓	✓	**59.68**	**71.48**	**72.37**	**52.37**	**57.33**	**72.12**	**74.92**
FuseGen (Ours)									

Table 4: Comparison of different in-context sample selection methods with QNLI as test dataset. "Variability" is cross-model variability, and "Rand." stands for random sampling for in-context sample candidate selection. $m_G, m_L, m_V, m_O, m_C, m_F$ each represents $m_{GPT-2}, m_{Llama-2}, m_{Vicuna}, m_{OPT}, m_{ChatGLM3}, m_{Flan-T5}$ and $\tilde{m}$ is the final STM trained using $\mathcal{D}$. Best result is marked as **bold** and the second best marked with underline for each STM (each column).

	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
FuseGen (Ours)	59.68	71.48	72.37	52.37	57.33	72.12	74.92
w/o SWA	56.72	69.99	70.94	51.98	56.39	68.65	73.41
w/o CDG & SWA	51.24	65.81	70.61	50.83	53.01	55.73	69.41
SDG+mixed	52.13	69.22	70.11	51.79	54.87	68.58	70.20

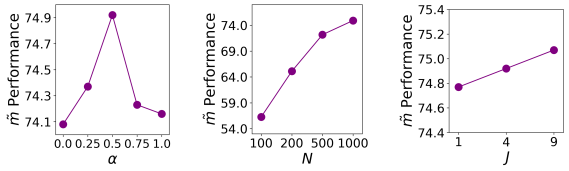
Table 5: Comparison between FuseGen and its ablations using $N = 1,000$ with QNLI as test dataset.

4.3.4 Effectiveness of SWA and CDG

As FuseGen consists of 2 components, CDG and CDI (mainly achieved by SWA), we perform ablation study by removing SWA and CDG step by step from FuseGen, resulting in 2 ablations: "w/o SWA" and "w/o CDG & SWA". Note when both CDI and CDG are removed, datasets are generated from multiple PLMs using zero-shot prompt and naively combined (the "mixed" case in Figure 2). We further add ablation "SDG+mixed" (also with-

		time [s]	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
1k	SunGen	43.3	**57.46**	**72.01**	72.14	50.71	**55.45**	57.31
	SWA	0.1	56.95	71.13	**72.21**	**51.96**	55.12	**57.43**
6k	SunGen	240.8	62.26	74.20	**74.35**	57.50	**65.64**	58.21
	SWA	0.5	**62.59**	**74.58**	**74.35**	**58.42**	64.81	**58.47**

Table 6: Comparison on running time for each weight adjustment epoch and STM performance between SunGen and SWA with QNLI as test dataset. Best result is marked as **bold**.



(a) Effect of α (b) Effect of N (c) Effect of J

Figure 7: Ablation results on different hyper-parameters used for FuseGen with QNLI as test dataset.

out SWA) which naively combines datasets given by multiple PLMs using self-guided data generation (SDG) for in-context feedback (same as $K = 1$ in Section 4.3.1). Results are summarized in Table 5 and Table 10 in Appendix C.6. From Table 5, we observe a 1.51% drop in $\tilde{m}$ performance when removing SWA, and another 5.51% drop when further removing CDG, demonstrating that *SWA is effective in boosting knowledge transfer from synthetic dataset to STM* and *CDG is effective in fusing the knowledge of multiple PLMs*. Also, CDG ("w/o SWA") outperforms "SDG+mixed" by a huge margin (3.21%), verifying the superiority of collaborative feedback over self-guided feedback.

As SunGen (Gao et al., 2023) also re-weights samples to boost STM performance, we further compare the performance of SWA with SunGen (using 50 samples for estimating gradients of sample weights), with results shown in Table 6. We observe that, SunGen's computational cost is two orders-of-magnitude higher than SWA, yet delivers comparable performance. This underscores the effectiveness and efficiency of SWA, demonstrating that our FuseGen framework is much more computationally effective.

4.3.5 Effect of Hyper-parameters

We further study the impact of hyper-parameters α (ratio of high-variability samples within the R in-context sample candidates), N (sample generation budget), and J (feedback times) of FuseGen with $K = 6$ in Figure 7. Detailed results with each m_k are included in Tables 12 to 14 in Appendix C.8.

Effect of α. Figure 7(a) shows that, too many or too few high-variability samples in the candidate set both hurt the synthetic dataset quality, resulting in lower STM performance, whereas a balanced mix ($\alpha = 0.5$) yields the highest STM results.

Effect of N. Figure 7(b) demonstrates that STM performance improves with the increase of N. Additionally, the performance improvement rate decelerates at larger values of N.

Effect of J. From Figure 7(c), we observe that increasing J results in a slight but consistent improvement in performance, likely due to the fact that more precise guidance is given to PLMs by a more frequent feedback during the process.

5 Conclusion

We propose a novel data-generation based zero-shot learning framework FuseGen that harnesses the collaborative capability of multiple PLMs to improve synthetic data generation of PLMs. We first integrate multiple PLMs to alleviate distribution bias of synthetic datasets through cross-PLM in-context samples selection, for constructing better feedback recursively. To further improve STM performance, we employ a self-boosting weight adjustment strategy to down-weight low-quality samples. Extensive experiments and ablation studies on various NLI and NLU tasks demonstrate that FuseGen is highly effective, query-efficient and PLM-agnostic without the reliance on specific PLMs for downstream tasks, making it a more flexible and resource-efficient solution.

Limitations

This work sheds lights on the possibility of multi-PLM collaboration in the field of zero-shot learning. However, it does not delve deeply into the interrelationships between pairs of PLMs. A more thorough investigation could yield insightful conclusions regarding which PLMs are most complementary to one another. Meanwhile, aside from seeding the same feedback to all PLMs, more personalized feedback can be constructed to better suit the inherit distribution bias of each PLM, which may further boost STM performances.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No.2022ZD0160504, the Tsinghua University (AIR)-Asiainfo Technologies (China) Inc. Joint Research Center.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. 2013. A Review of Feature Selection Methods on Synthetic Data. *Knowledge and information systems*, 34:483–519.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling Instruction-finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for Transfer Learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200.

Yongheng Deng, Ziqing Qiao, Ju Ren, Yang Liu, and Yaoxue Zhang. 2023. Mutual Enhancement of Large and Small Language Models with Cross-silo Knowledge Transfer. *arXiv preprint arXiv:2312.05842*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv preprint arXiv:2305.14325*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Jiahui Gao, Renjie Pi, Lin Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided Noise-free Data Generation for Efficient Zero-shot Learning. In *Proceedings of The Eleventh International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the Eighth International Conference on Learning Representations*.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More Agents is All You Need. *arXiv preprint arXiv:2402.05120*.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024. A dynamic llm-powered agent network for task-oriented agent collaboration. In *Proceedings of the 1st Conference on Language Modeling (COLM 2024)*.

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

Costas Mavromatis, Petros Karypis, and George Karypis. 2024. Pack of LLMs: Model Fusion at Test-Time via Perplexity Optimization. *arXiv preprint arXiv:2404.11531*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating Training Data with Language Models: Towards Zero-shot Language Understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

OpenAI. 2021. GPT-3.5-Turbo.

OpenAI. 2023. GPT-4-Turbo and GPT-4.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yixuan Su and Nigel Collier. 2022. Contrastive Search is What You Need for Neural Text Generation. *arXiv preprint arXiv:2210.14140*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. 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.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-tuned Chat Models, 2023. *URL https://arxiv.org/abs/2307.09288*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024a. Knowledge Fusion of Large Language Models. In *Proceedings of The Twelfth International Conference on Learning Representations*.

Fanqi Wan, Ziyi Yang, Longguang Zhong, Xiaojun Quan, Xinting Huang, and Wei Bi. 2024b. FuseChat: Knowledge Fusion of Chat Models. *arXiv preprint arXiv:2402.16107*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In the Proceedings of The Seventh International Conference on Learning Representation.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing Open-source Language Models with Mixed-quality Data. *arXiv preprint arXiv:2309.11235*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022a. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022b. ProGen: Progressive Zero-shot Dataset Generation via In-context Feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias. *Advances in Neural Information Processing Systems*, 36.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *Preprint*, arXiv:2205.01068.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-Level Convolutional Networks for Text Classification. *Advances in neural information processing systems*, 28.

A Prompts Used in Experiments

A.1 Task-related Label-descriptive Prompts

We present the prompts used for synthetic dataset generation in Table 7. For information-question entailment analysis task (QNLI) and sentence pair relation analysis task (MNLI), we leverage the open-source Wikipedia-short (https://github.com/yumeng5/SuperGen/tree/main/pretrain_corpus) dataset, which contains short Wikipedia sequences (5 to 30 words) extracted from sentences in Wikipedia. We use these sentences as the information source for the prompts. In other words, each occurrence of <*information*> or <*sentence1*> within the prompt is replaced with a randomly-chosen Wikipedia-short sequence before feeding it to PLMs.

Below we also provide 2 examples of the few-shot prompts used in FuseGen . We need to clarify that, label information is not included in the in-context samples.

Few-shot prompt for movie review semantic analysis

The movie review is: This is an excellent romantic comedy that relies more on wit and character than on silly, typical formula. A lot of people I know walked away from this movie disappointed, but I found it an enjoyable experience. I also don't understand why Hollywood thinks that 'quirkiness' is more important than story, or why they can't seem to create movies in which the plot is interesting and makes sense.
The movie review is: There's a lot of talent wasted here. Haggis overuses his themes and is unable to let his characters go in this soapy melodrama.
The movie review is: The movie is not fast paced and some of the drama was a bit too much for me, but I did like it.
The movie review is: There is a certain helplessness in allowing ourselves to be tricked by the tricky cuts that grace the first half of the film. It allows us to suspend our disbelief and see what we want to see. It's not a movie I'd love to watch again, but it is one I'm glad I got to see.
The movie review is: I will be the first to admit that the animation is crude in some parts. What I liked about the movie is that it had a very fun story line and I loved the songs. The movie review is: There's no reason you shouldn't enjoy this semi-tangential off-shoot of a popular video game; it's a fun, goofy movie that doesn't rely on the whole 'cinematic universe' concept
The movie review is: engaging and entertaining, with excellent performances from David Niven and Barbara Stanwyck. 2.Sheila is stunning in the movie, a lady obsessed with the detective, especially when working in an area with limited light. 3.The climax is shocking - but it's entirely appropriate, as the plot's terrible.
The movie review is: Many don't like the hero, and still others were glad they saw it and it was good. With that said, there are some surprising plot holes, inconsistencies and potential points of plot-holes that also need to be addressed before anyone can put their money into the film. If anyone was wondering how people like things and don't like other people like things, this movie is a great example.

The new movie review in negative sentiment which is diverse in the expression compared to the above given samples is:

Few-shot prompt for information-question entailment analysis

The Information-Question pair is: Soon after, the account began to go viral, attracting the attention of reddit streams, content aggregators, art critics, and Renoir\u2019s own descendants.[SEP]and Renoir's own accounts suggests that they met in early November 1881 when the baron stopped at their boardinghouse. "Below a quadriga in the Louvre courtyard, Henri left his easel with his model and ran up the stairway to Duret with the idea of showing him what he had accomplished." (from Renoir's biography by Fr?
The Information-Question pair is: She made her American debut in 1910, with the New York Symphony Orchestra, under conductor Walter Damrosch.[SEP]If this photo were to depict a specific moment in history, or an individual's life, which historical period or individual would it most closely resemble?
The Information-Question pair is: The Fall Line is an American true crime podcast that covers lesser-known cases of murder and disappearance from minority communities in Georgia.[SEP]The founder is the founder. If the owner owns the club, is it the 'Alamo' of crime blogs (or is it an 'evil bar')?
The Information-Question pair is: She was a Member of the Supreme Council of the Uzbek SSR.[SEP]Who was the head of the Uzbek SSR during her time on the Supreme Council?

The new Information-Question pair which is diverse in the expression compared to the above given samples is: Information: "<information>" Question (answer not in above information):

B Detailed Algorithms

We provide the detailed algorithms for each function used in Algorithm 1 here in Algorithm 2.

C Additional Experimental Results

C.1 Dataset Cartography of More Synthetic Datasets

C.1.1 Dataset Cartography Visualization

Dataset cartography (Swayamdipta et al., 2020) approach characterizes each sample by its confidence and variability, which are defined as the mean and

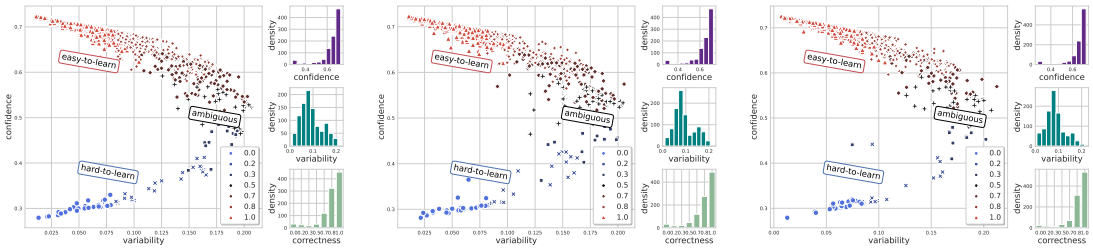
Dataset (task)	type	prompt	label
IMDb and SST2 (semantic analysis of movie review)	zero-shot	"The movie review in ***positive/negative*** sentiment for a movie is: "	***positive/negative***
	few-shot	"The movie review is: <*sample_1*> The movie review is: <*sample_2*> ... The movie review is: <*sample_S*> The movie review in ***positive/negative*** sentiment which is diverse in the expression compared to the above given samples is: "	***positive/negative***
Yelp (semantic analysis of restaurant review)	zero-shot	"The restaurant review in ***positive/negative*** sentiment is:"	***positive/negative***
	few-shot	"The restaurant review is: <*sample_1*> The restaurant review is: <*sample_2*> ... The restaurant review is: <*sample_S*> The new restaurant review in ***positive/negative*** sentiment which is diverse in the expression compared to the above given samples is: "	***positive/negative***
QNLI (information-question entailment analysis)	zero-shot	"Information: <*information*> Question (answer ***in/not in*** above information): "	***entailment/not_entailment***
	few-shot	"The Information-Question pair is: <*sample_1*> The Information-Question pair is: <*sample_2*> ... The Information-Question pair is: <*sample_S*> The new Information-Question pair which is diverse in the expression compared to the above given samples is: Information: <*information*> Question (answer ***in/not in*** above information): "	***entailment/not_entailment***
MNLI (matched and mismatched) (sentence pair relation analysis)	zero-shot	"<sentence1> ***In other words,*** / <sentence1> ***Furthermore,*** / ***There is a rumor that*** <sentence1> ***However, the truth is:*** "	***entailment/ neutral/ contradiction***
	few-shot	"The sentence pair is: <*sample_1*> The sentence pair is: <*sample_2*> ... The sentence pair is: <*sample_S*> The new sentence pair which is diverse in the expression compared to the above given samples is: <sentence1> ***In other words,*** / <sentence1> ***Furthermore,*** / ***There is a rumor that*** <sentence1> ***However, the truth is:*** "	***entailment/ neutral/ contradiction***
AgNews (news articles classification)	zero-shot	"The news articles is in the category of ***World/Sports/Business/Technology***: "	***World/Sports/ Business/Technology***
	few-shot	"The news article is: <*sample_1*> The news article is: <*sample_2*> ... The news article is: <*sample_S*> The new news article in the category of ***World/Sports/Business/Technology*** which is diverse in the expression compared to the above given samples is: "	***World/Sports/ Business/Technology***
MarkedNews (self-defined news articles classification)	zero-shot	"A news article in the category of ***World that does not include '$'/Sports that does not include '$'/Business that does not include '$'/Technology that does not include '$'/Money with '$' included***: "	***World/Sports/ Business/Technology/ Money with $ included***
	few-shot	"The news article is: <*sample_1*> The news article is: <*sample_2*> ... The news article is: <*sample_S*> The new news article in the category of ***World that does not include '$'/ Sports that does not include '$'/Business that does not include '$'/ Technology that does not include '$'/Money with '$' included*** which is diverse in the expression compared to the above given samples is: "	***World/Sports/ Business/Technology/ Money with $ included***

Table 7: Prompt used for synthetic dataset generation.

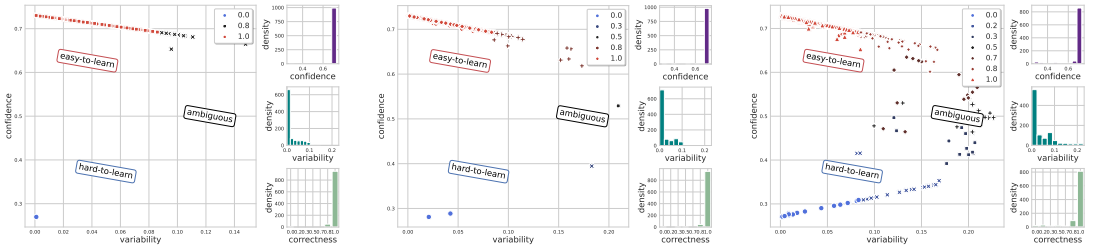
standard deviation of the model probability of its related label across all training epochs. For example, if the model correctly predict a sample's label across training epochs, it will have high confidence and low variability. These samples are regarded as *easy-to-learn* samples , whereas those with low variability yet low confidence are identified as *hard-to-learn* samples. Conversely, samples with high variability are deemed *ambiguous*.

We provide dataset cartography of synthetic datasets generated by 6 different PLMs (GPT-2, Llama-2, Vicuna, OPT, ChatGLM3 and Flan-T5) in Figure 8 . In left-subplot of each sub-figure in Figure 8, we display the variability (x-axis) and confidence (y-axis) of all samples. The right subplots depict histograms detailing the distributions of confidence, variability, and correctness. Notice that exactly $1,000$ samples are scattered onto each plot, although samples may overlap with each other, creating a visually sparser impression.

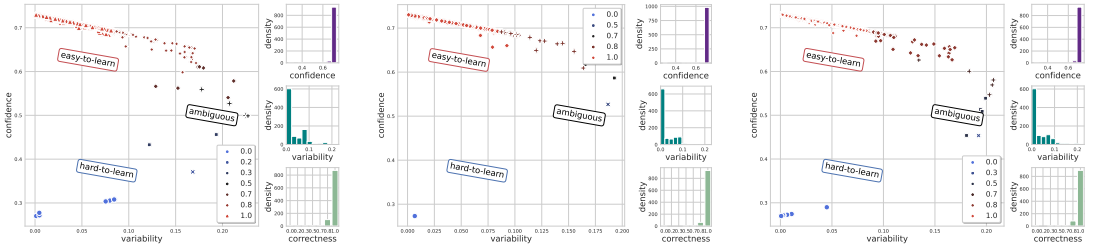
Comparing dataset cartography generated by the same PLM, we can see that FuseGen helps to improve the dataset composition by introducing more ambiguous samples to balance the prevalence of the easy-to-learn samples, while ensuring hard-to-learn samples remain a minority.



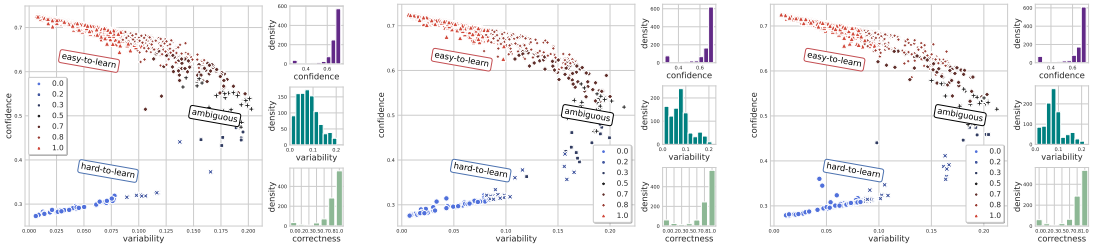
(a) GPT-2 ZeroGen K=1 (84.83) (b) GPT-2 ProGen K=1 (85.74) (c) GPT-2 Ours K=6 (87.85)



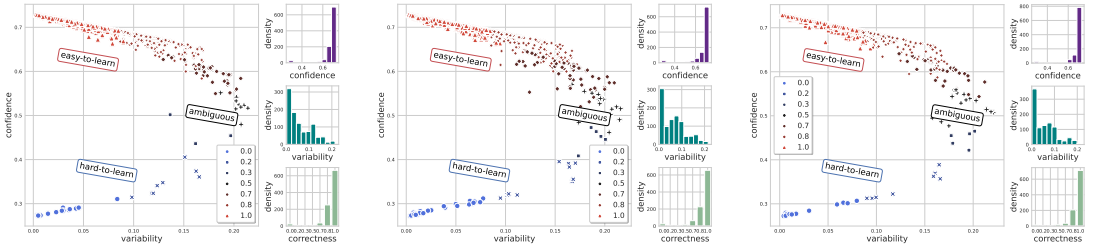
(d) Llama-2 ZeroGen K=1 (84.23) (e) Llama-2 ProGen K=1 (84.24) (f) Llama-2 Ours K=6 (86.60)



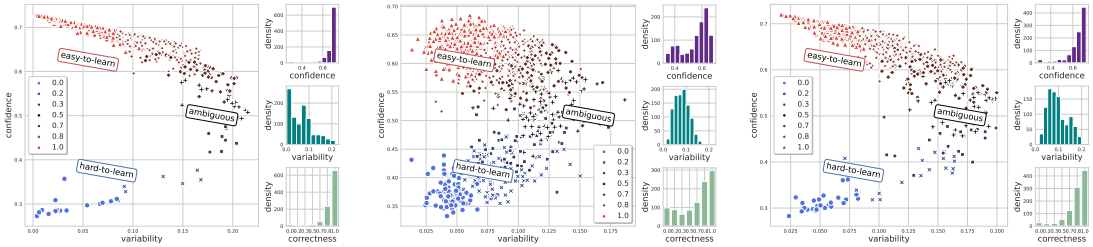
(g) Vicuna ZeroGen K=1 (82.37) (h) Vicuna ProGen K=1 (83.60) (i) Vicuna Ours K=6 (87.50)



(j) OPT ZeroGen K=1 (84.97) (k) OPT ProGen K=1 (87.57) (l) OPT Ours K=6 (88.47)



(m) ChatGLM3 ZeroGen K=1 (86.43) (n) ChatGLM3 ProGen K=1 (87.07) (o) ChatGLM3 Ours K=6 (88.56)



(p) Flan-T5 ZeroGen K=1 (88.18) (q) Flan-T5 ProGen K=1 (85.80) (r) Flan-T5 Ours K=6 (88.73)

Figure 8: Synthetic dataset cartography (Swayamdipta et al., 2020) using $1,000$ generated samples for movie review semantic analysis. ZeroGen uses zero-shot prompt for generation, while ProGen and FuseGen (Ours) use few-shot prompt with feedback but with different K, the number of PLMs involved. Numbers within parentheses are STM performance evaluated using IMDb after training on the generated dataset, with SWA applied during training.

Algorithm 2 Functions used in Algorithm 1 for FuseGen

function `S_AccumulativeSynDataGeneration`($\mathcal{D}_k$, $\hat{\mathcal{D}}$, N, J, j):
  if $j = 0$ **then**
    Use zero-shot prompt as working prompt $\mathcal{T}$.
  else
    Use $\hat{\mathcal{D}}$ to create few-shot prompt as working prompt $\mathcal{T}$.
  end if
  Generate $\frac{N}{J+1}$ samples using $\mathcal{T}$ and add them to $\mathcal{D}_k$.
  return $\mathcal{D}_k$.

function `S_STMTraining`($\mathcal{D}$, $m_{(0)}$, E_2):
  Initialize a trainable STM $m \leftarrow m_{(0)}$ and train m using $\mathcal{D}_k$ for E_2 epochs with Equation (2).
  return m.

function `C_SampleSelection`($\mathcal{D}$, $\{m_k\}_{k=1}^K$, $\tilde{m}$, α, R, S):
  Reset $\hat{\mathcal{D}} \leftarrow \emptyset$.
  for $k' = 1$ **to** K **do**
    for Each sample $(\mathbf{x}_{k,i}, y_{k,i})$ in $\mathcal{D}$ **do**
      Obtain the prediction vector $p_{k',k,i} = m_{k'}(\mathbf{x}_{k,i}) \in \mathbb{R}^C$ and predicted label-position probability $p_{k',k,i}[y_{k,i}] \in \mathbb{R}^1$.
      Calculate disagreement score $d_{k,i} = \mathrm{STD}(p_{1,k,i}[y_{k,i}], ..., p_{k',k,i}[y_{k,i}], ..., p_{K,k,i}[y_{k,i}])$.
    end for
  end for
  Sort all the samples within $\mathcal{D}$ and add the top-$(1-\alpha)R$ samples with the lowest score and top-αR samples with the highest samples into $\hat{\mathcal{D}}$.
  Calculate the influence score of each sample in $\hat{\mathcal{D}}$ with $\tilde{m}$ using Eq.(3) in Ye et al. (2022b).
  $\hat{\mathcal{D}} \leftarrow$ {top-S samples with the highest influence score}.
  return $\hat{\mathcal{D}}$.

function `S_WeightAdjustSTMTraining`($\mathcal{D}$, $m_{(0)}$, $\{w_i^{(0)}\}_{i=1}^N$, E_1, E_2):
  for $e_1 = 0$ **to** $E_1 - 1$ **do**
    Initialize a trainable STM $m \leftarrow m_{(0)}$ and train m using $\mathcal{D}$ for E_2 epochs with weighted loss using $\{w_i^{(e_1)}\}_{i=1}^N$ and Equation (5).
    Adjust sample-level weight $w_i^{(e_1+1)} \leftarrow w_i^{(e_1)}$ with m using Equation (4) for each sample $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$.
  end for
  return m.

C.1.2 Relationship Between Synthetic Dataset Distribution Biases and Performance

We examine 2 statistical metrics, namely the mean and standard deviation (STD), of the variability (defined in Swayamdipta et al. (2020)) of each sample in the synthetic dataset plotted in Figure 8 that are each given by a single PLM$_k$ ($\mathcal{D}_k$). The results are presented in Table 8. We further conduct a Pearson Correlation Coefficient test to evaluate the correlation between these two metrics and the final STM performance separately. Considering all values in Table 8, the correlation coefficient and p-value between the Mean of variability and STM performance are 0.494 and 0.037 (< 0.050) respectively, while that between the STD of variability and STM performance are 0.500 and 0.035 (< 0.050). These results support the hypothesis that there is a statistically significant positive correlation between both the Mean and STD of sample variability and the final STM performance.

Method		GPT-2	Llama-2	Vicuna	OPT	ChatGLM3	Flan-T5
ZeroGen	Mean	0.094	0.021	0.033	0.080	0.061	0.068
	STD	0.045	0.029	0.042	0.044	0.052	0.052
	ACC	84.83	84.23	82.37	84.97	86.43	88.18
ProGen	Mean	0.095	0.021	0.025	0.077	0.066	0.082
	STD	0.044	0.031	0.033	0.046	0.052	0.031
	ACC	85.74	84.24	83.60	87.57	87.07	85.80
FuseGen (Ours)	Mean	0.087	0.041	0.030	0.083	0.058	0.096
	STD	0.041	0.050	0.039	0.042	0.051	0.042
	ACC	87.85	86.60	87.50	88.47	88.56	88.73

Table 8: Mean and standard deviation (STD) of the variability for each synthetic dataset in Figure 8 with corresponding STM performance (trained with Self-boosting Weight Adjustment) using IMDb. For FuseGen, results are the performance of each $\tilde{m}_k$.

C.2 T-SNE Visualization of Sample Distributions

We also visualize the t-distributed Stochastic Neighbor Embedding (t-SNE) of synthetic samples ($N = 1,000$) in Figure 9. All samples are embedded with a pre-trained bert-base-uncased encoder model.

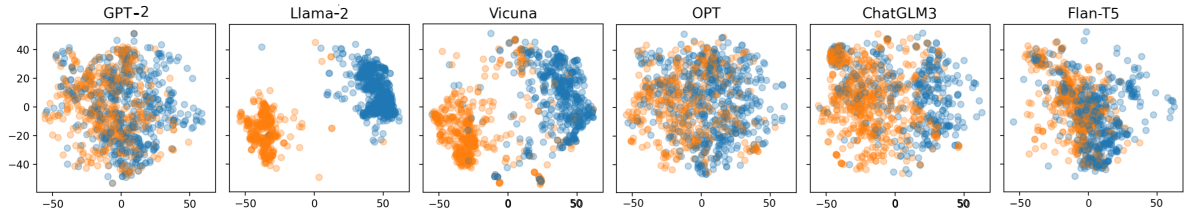
Consistent with the dataset cartography in Figures 1 and 8, FuseGen generates a higher proportion of ambiguous samples, which pulls the distribution of samples from different semantic classes closer to each other compared to ZeroGen and ProGen. This effect is particularly pronounced for synthetic datasets given by Llama-2 and Vicuna.

C.3 Low-quality Synthetic Dataset Samples

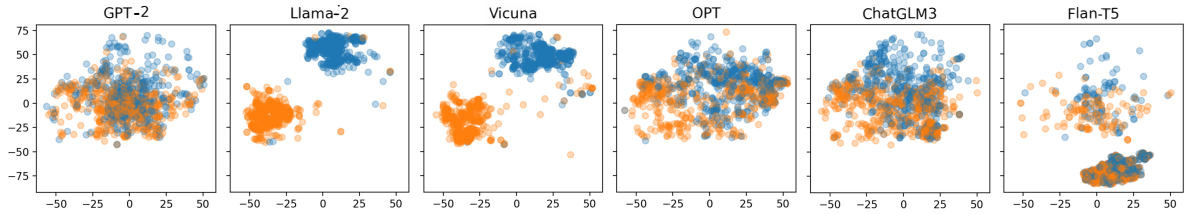
In Table 9, we show examples of low-quality samples, including samples that are "mislabeled", of "low-relevancy", and of "low-text-quality". Samples are selected from synthetic datasets generated by individual PLMs using zero-shot prompt for the movie review semantic analysis task. This demonstrates the importance for improving the overall data quality of synthesic datasets.

C.4 Source of Selected In-context Samples

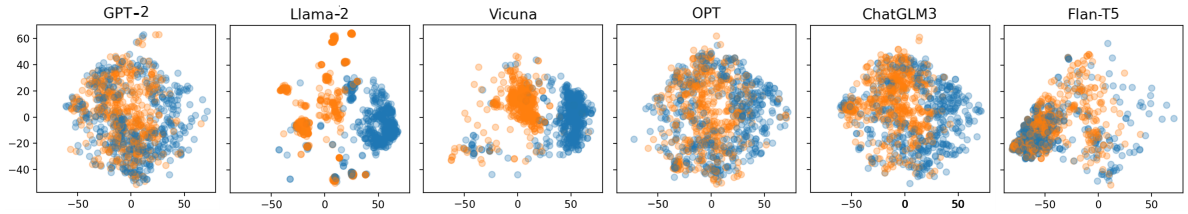
We show in Figure 10 that, the selected in-context samples (desirable subset) and its candidates during CDG originate from various PLMs. However, the



(a) ZeroGen



(b) ProGen



(c) FuseGen (Ours)

Figure 9: t-SNE visualization of each synthetic sample generated by 6 PLMs for movie review task. Different colors, blue and orange, represents embeddings from different class, positive and negative respectively.

proportion of samples contributed by each PLM can fluctuate across iterations. This verifies that knowledge from different PLMs are fused and fed to each PLM through the feedback prompt, which further boosts the generation quality of each PLM.

C.5 Larger Synthetic Datasets for Question Answering Tasks

Note that, as the NLG task is harder than NLI and NLU tasks, training a BERT with a total of $6,000$ samples does not result in high performance (see Table 1). Therefore, we additionally performed experiments with a total of $K \times N = 6 \times 6,000 = 36,000$ samples. FuseGen achieves an F1 score of 15.79. For the baselines, the best and second-best performing baselines under the smaller synthetic dataset size setting, i.e., SunGen using GPT-2 and ProGen using ChatGLM3, result in F1 scores of 13.57 and 13.12 respectively.

C.6 Ablations on More Tasks

We include the ablation results of "w/o SWA", "w/o CDG & SWA" and "SDG+mixed"(also w/o SWA) for more tasks and here due to space limitation. We also elaborate the explanation of "SDG+mixed" here. In "SDG+mixed", SWA is removed and CDG is replaced with self-based feedback, i.e. random selection is applied to select R candidate samples from each $\mathcal{D}_k$. K in-context samples subsets are than selected based on sample importance from the K candidate sample sets of size R and are further fed to respective PLM $\mathcal{P}_k$ to generate samples.

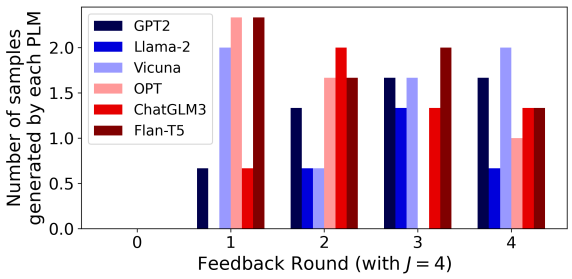
As shown in Table 10, the application of SWA significantly boosts the performance of all STMs, particularly for $\{m_k\}_{k=1}^{K}$. This improvement highlights the efficacy of SWA in enhancing the quality of synthetic datasets through the up-weighting of higher-quality samples and the down-weighting of lower-quality samples, thereby reducing the impact of the latter. Furthermore, the application of CDG also significantly boosts the performance of all STMs to a greater extent than applying SDG. This underscores the superiority of cross-model feedback over the combination of self-guided feedback and highlights the efficacy of CDG in harnessing the capabilities of multiple PLMs.

PLM	Sample Text	Label	Type
GPT-2	A touching portrayal of friendship in the troubled 21st Century.	negative	mislabeled
GPT-2	We believe the movie <C> is decent but it's none too unexpected for someone like me who has fallen asleep watching movies and has seen this film before.	positive	mislabeled
GPT-2	The 'Godzilla' of the year thing is back. Say hello to The Amazing Spider-Man 2	negative	low-relevancy
GPT-2	Broken Hearts Way was, on average, better than <T> was	positive	low-text-quality
GPT-2	Help! My name is Mark Hammer and I have always wanted to write a movie review. Now, after hours and hours of research, the truth is finally out.	negative	low-text-quality
Llama-2	It was a complete waste of time and money, I would not recommend it to anyone. This movie was just plain bad, with uninteresting characters and a plot that was predictable from the start. The acting was also subpar and the dialogue was cringe-worthy. I would give it a 0/10 if I could.	positive	mislabeled
Llama-2	After waiting for what felt like an eternity, I finally watched	negative	low-text-quality
Vicuna	The film, directed by a renowned auteur, is a story of a group of people navigating through a dystopian society.**** It is a thought-provoking piece that leaves the audience pondering on the state of society and what it takes to survive in a world that is controlled by a powerful corporation. The cinematography is beautiful and the score is hauntingly good.	negative	mislabeled
Vicuna	A great movie that will leave you thinking for days to come. No need to spend a lot of money to see it. It is a must watch for everyone.	negative	mislabeled
Vicuna	C++ is a difficult and complex language, but with time and practice, it becomes easy and manageable.1234567890	positive	low-relevancy
OPT	But the movie begins with this big, action-packed explosion, which carries right into the more thoughtful segments of the movie. But wait! There's more! It continues through those thoughtful segments and the entire rest of the movie, not just a handful of sequences.	negative	mislabeled
OPT	The movie is very disappointing. The story line is nothing but a shameless rip-off. Even the main cast of the movie is not worth any praise at all. This is a movie to just go and waste your money. Just don't!	positive	mislabeled
OPT	When my teenage self in France was presented with this movie on VHS, I was in love with the soundtrack, with the French style of acting (something totally alien to my home in the States), and with the idea that there was a world like this.	negative	mislabeled
OPT	The packaging is nothing more than an anonymous brown paper bag, and the theater provided stale popcorn.	positive	low-relevancy
OPT	\n\n- a negative movie review\n\nThe movie review in negative sentiment for movie	positive	low-text-quality
ChatGLM3	Very disappointing. There was not one LOL moment. No wonder the movie was not a box office hit.	positive	mislabeled
ChatGLM3	Perhaps a crime movie and is interesting to watch .	negative	mislabeled
ChatGLM3	i'm not the most romantic person and i'm not a chick.	positive	low-relevancy
ChatGLM3	even a bad magician should be able to catch the rabbit	positive	low-relevancy
Flan-T5	He works in audio-visual technique and the end product is often flawed.	positive	mislabeled
Flan-T5	When a thing is a fantasy, it just become real, whether it was imagined or just played out. When they put on a performance in this movie, it has to be one of the best, most inspired moments.	negative	mislabeled
Flan-T5	if the time has come to say goodbye to Dick Van Patten.	positive	low-relevancy
Flan-T5	perverse creatures know they should be ashamed to exist. for human beings to walk around dressed like cannibals in a heavy jungle set up camp.	negative	low-relevancy
Flan-T5	And this is just another (incomplete) list of things that	negative	low-text-quality

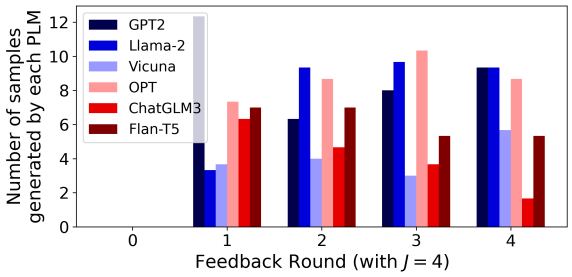
Table 9: Examples of low-quality samples in generated synthetic dataset for movie review.

C.7 Multi-PLM v.s. single-PLM on More Tasks

We provided additional results on the comparison of multi-PLM ($K = 6$) and single-PLM ($K = 1$) across 8 datasets for various tasks in Table 11. As multi-PLM consistently outperforms all single-PLM under the each task, we conclude that multi-PLM collaboration is more effective than relying



(a) Samples in selected desirable subset of size $S = 8$



(b) Selected candidates of size $R = 40$

Figure 10: Proportion of samples in S in-context samples and R sample candidates that originate from each PLM at each feedback time (J) in FuseGen with $J = 4, R = 40, S = 8, N = 1,000, K = 6$ for movie review sentiment analysis task. Results are averaged using 3 different seeds.

	IMDb						
	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
FuseGen (Ours)	**87.85**	**86.60**	**87.50**	**88.47**	**88.56**	**88.73**	**90.19**
w/o SWA	82.90	78.98	74.34	85.17	85.77	85.43	89.07
w/o CDG & SWA	80.71	75.73	59.41	81.37	81.14	84.35	87.06
SDG+mixed	80.72	76.18	65.05	84.19	84.56	81.19	87.41
	SST-2						
	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
FuseGen (Ours)	**86.38**	**84.36**	**85.52**	**86.50**	**86.96**	**86.32**	**87.35**
w/o SWA	81.87	79.22	82.43	80.99	85.73	80.99	85.38
w/o CDG & SWA	80.68	76.42	76.46	80.80	84.58	78.44	85.01
SDG+mixed	80.75	77.53	79.52	80.86	85.69	80.89	85.71
	Yelp						
	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
FuseGen (Ours)	**91.94**	**90.30**	**90.81**	**92.50**	**92.98**	**92.21**	**93.54**
w/o SWA	90.87	88.09	84.99	87.19	91.72	90.71	92.84
w/o CDG & SWA	89.13	79.17	81.97	86.78	81.50	89.48	92.16
SDG+mixed	89.63	82.39	83.80	86.84	86.32	87.48	92.23
	QNLI						
	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
FuseGen (Ours)	**60.55**	**72.48**	**74.10**	**57.39**	**69.89**	**72.13**	**74.95**
w/o SWA	56.72	69.99	70.94	51.98	56.39	68.65	73.41
w/o CDG & SWA	51.24	65.81	70.61	50.83	53.01	55.73	69.41
SDG+mixed	52.13	69.22	70.11	51.79	54.87	68.58	70.20

Table 10: Comparison between FuseGen and its ablations with $K = 6, N = 1,000, J = 4$. Each m_k is trained on $\mathcal{D}_k$ of size $1,000$ while $\tilde{m}$ is trained on $\mathcal{D}$ of size $6,000$. Best result is marked as **bold** for each STM (each column).

on a single PLM for enhancing STM performance.

	multi	single					
	$\tilde{m}$	$\tilde{m}_G$	$\tilde{m}_L$	$\tilde{m}_V$	$\tilde{m}_O$	$\tilde{m}_C$	$\tilde{m}_F$
IMDb	**89.96**	87.60	86.14	85.42	87.59	88.84	89.74
SST-2	**87.51**	84.81	84.39	85.22	85.88	87.43	85.38
Yelp	**93.27**	93.03	91.07	91.69	92.72	92.08	92.07
QNLI	**74.92**	64.52	73.22	73.34	59.03	64.93	73.60
MNLI-m	**49.76**	44.93	49.61	49.11	37.40	32.82	49.34
MNLI-mm	**51.70**	48.53	51.62	50.76	42.32	33.05	51.47
AgNews	**86.89**	82.21	85.34	85.36	86.75	86.27	86.36
MarkedNews	**83.85**	79.98	80.04	79.36	78.60	83.54	80.86

Table 11: Comparison between FuseGen using multi-PLM ($K = 6$) and single-PLM ($K = 1$) with 4 datasets. MNLI-m and MNLI-mm each stands for MNLI-matched and MNLI-mismatched. Best result is marked as **bold** with the second best marked with underline for each dataset (each row).

α	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
0.0	54.00	70.07	67.75	51.12	55.70	66.49	74.08
0.25	56.12	70.22	70.45	52.10	56.90	71.12	74.37
0.5	**59.68**	**71.48**	**72.37**	**52.37**	**57.33**	**72.12**	**74.92**
0.75	55.27	69.13	69.53	52.19	56.59	70.91	74.23
1.0	54.85	66.47	64.46	50.08	56.50	70.50	74.16

Table 12: Comparison of different α used for FuseGen with QNLI as test dataset. Best result is marked as **bold** with the second best marked with underline for each STM (each column).

N	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
100	51.33	53.16	53.79	50.62	51.20	51.11	56.27
200	52.23	60.42	60.06	50.71	53.07	59.09	65.11
500	53.53	67.36	67.90	51.67	54.95	64.72	72.18
$1,000$	**59.68**	**71.48**	**72.37**	**52.37**	**57.33**	**72.12**	**74.92**

Table 13: Comparison of different N used for FuseGen with QNLI as test dataset. Best result is marked as **bold** with the second best marked with underline for each STM (each column).

C.8 Detailed Results for Hyper-parameters

Due to space limitation, we provide detailed results of hyper-parameters α (ratio of high-variability samples within the R in-context sample candidates), N (sample generation budget), and J (feedback times) here in Tables 12 to 14. We additionally include the performance of each m_k as well (SWA applied). These results indicate that employing a more balanced mix of high-variability and low-variability samples ($\alpha = 0.5$), a larger sample budget N and more feedback times J all help to achieve a better STM performance. This enhancement is observed not only for the final STM $\tilde{m}$, but also for each $\{m_k\}_{k=1}^K$.

J	m_G	m_L	m_V	m_O	m_C	m_F	$\tilde{m}$
0	56.95	71.13	72.21	51.96	55.12	58.43	74.44
1	57.11	<u>71.50</u>	72.25	52.07	56.53	64.81	74.77
4	<u>59.68</u>	71.48	**72.37**	**52.37**	<u>57.33</u>	<u>72.12</u>	<u>74.92</u>
9	**59.71**	**71.60**	**72.37**	<u>52.34</u>	**57.70**	**72.14**	**75.07**

Table 14: Comparison of different J used for FuseGen with QNLI as test dataset. Best result is marked as **bold** with the second best marked with <u>underline</u> for each STM (each column).