FROM STATIC PROMPTS TO TRAINABLE POLICIES: ADAPT FOR ADVERSARIAL TEXT AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data augmentation often yields redundant examples that fail to meaningfully challenge downstream models. We introduce **ADAPT** (**A**dversarial **D**ata **A**ugmentation via **P**olicy **T**raining), a framework that transforms static large language model prompts into dynamic, trainable policies via group-relative policy optimization. By iteratively fine-tuning a generator based on adversarial student difficulty and semantic fidelity, ADAPT synthesizes highly sample-efficient, boundary-testing data curricula. Operating as a universal wrapper over standard augmentation interfaces (e.g., AugGPT, LLM2LLM, LMTransplant), ADAPT delivers statistically significant downstream gains across varying generator capacities and dataset scales. Specifically, ADAPT improves intent detection performance by up to 7.7 percentage points for Qwen-2B and 6.1 points for Gemma-31B over unadapted baselines, while maintaining strict label preservation ($\geq 96\%$) and significantly increasing structural diversity. Ultimately, ADAPT maximizes student generalization under optimized budgets, decisively outperforming massive, unguided generation.

1 INTRODUCTION

Low-resource learning requires synthetic text that actively challenges a student model's decision boundaries, rather than merely paraphrasing what it already knows. Effective data augmentation must therefore strike a delicate balance: introducing structural novelty and targeting student weaknesses while strictly preserving semantic labels. Existing methods like AugGPT, LLM2LLM, and LMTransplant (Dai et al., 2025; Lee et al., 2024; Wang et al., 2025) provide valuable structural variations but rely on static generator weights. This rigid paradigm limits their ability to dynamically adapt to the evolving vulnerabilities of the student model.

To overcome this limitation, we elevate static augmentation prompts into trainable policies. We introduce **ADAPT** (**S**tudent-**P**ressure **A**dversarial **R**egeneration), a framework that adapts a generator to synthesize highly informative examples using a group-relative policy-gradient update inspired by GRPO (Shao et al., 2024) (see Figure 1). ADAPT scores candidate augmentations against downstream student weaknesses while the generator self-validates its own outputs, eliminating the overhead of an external reward model. This self-validation operates within strict guardrails, relying on seed annotations, structural constraints, and task-specific semantic validity checks to prevent unconstrained semantic drift.

We posit that combining any existing prompt-based augmentation interface with adversarial adaptation will strictly outperform the unadapted interface alone. To rigorously test this across varying generator capacities and data scales, we integrate ADAPT into AugGPT, LLM2LLM, and LMTransplant using three LLM families (Qwen-2B, Llama-7B, and Gemma-31B). We benchmark performance on five diverse NLP tasks (sentiment analysis, question classification, intent detection, extractive QA, and named entity recognition) across four dataset size regimes ranging from Small to Extra Large.

Our empirical results confirm that dynamic adversarial adaptation is universally beneficial. ADAPT consistently elevates downstream performance, with smaller open-weights generators exhibiting disproportionately large marginal gains that bridge the capability gap with massive, unadapted models. In summary, our concrete contributions are: (1) **A Universal Adversarial Framework:** We propose ADAPT, a group-relative adversarial augmentation framework applicable to any prompt-based

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

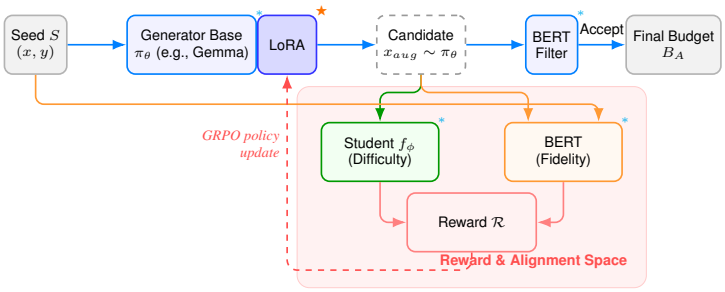

Figure 1: ADAPT architecture. **Top Track:** The generator synthesizes augmentations routed through a similarity filter to form the optimized budget B_A. **Bottom Track (Alignment Space):** Candidates are evaluated by the student (difficulty) and BERT (semantic fidelity). The combined reward $\mathcal{R}$ updates the LoRA parameters via GRPO. ($*$: Frozen weights; ★: Trainable weights).

interface. It utilizes self-validation to cleanly disentangle generation interfaces from policy updates and semantic guardrails. (2) **Robust Empirical Gains:** We provide extensive evidence demonstrating statistically significant downstream improvements, alongside superior semantic variability and label preservation, across multiple LLMs, baseline interfaces, and NLP tasks. (3) **Sample-Efficient Curricula:** Through comprehensive data scaling and budget analyses, we show that ADAPT's multi-iteration adapter pooling naturally curates a highly sample-efficient adversarial curriculum, maximizing student generalization under tight budgets and outperforming massive unguided generation.

2 Related Work

Traditional text augmentation relies on rule-based lexical perturbations (Wei & Zou, 2019) or translation-based paraphrasing (Sennrich et al., 2016), which often struggle to introduce meaningful diversity while strictly preserving task-relevant semantics (Feng et al., 2021; Ding et al., 2024). To address this, recent methods leverage the generative capacity of large language models via mixed demonstrations (Yoo et al., 2021), instruction-following (Dai et al., 2025), or contextual regeneration (Wang et al., 2025). While approaches like LLM2LLM (Lee et al., 2024) incorporate student awareness by identifying incorrectly predicted examples to guide targeted prompting, they still rely on static generator weights. While conceptually related to conditional generative adversarial networks (Mirza & Osindero, 2014; Goodfellow et al., 2014), ADAPT optimizes discrete textual variations via reinforcement learning driven by downstream student difficulty rather than backpropagating continuous gradients from a real-versus-fake discriminator. This dynamic adaptation is enabled by parameter-efficient fine-tuning (Hu et al., 2021; Dettmers et al., 2023) and reward-based optimization techniques like RAFT (Dong et al., 2023) and GRPO (Shao et al., 2024).

3 Methodology

Problem Setup and Outer Iterations Let $D_0 = \{(x_i, y_i)\}_{i=1}^{S}$ be a seed dataset of size S. Our goal is to generate a synthetic set D_{aug} of size B such that a student model f_ϕ trained on $D_0 \cup D_{\text{aug}}$ outperforms one trained on D_0 alone. The complete pseudocode for this framework is provided in Appendix F. As depicted in Figure 1, ADAPT alternates between training the student and adapting the generator π_θ over T outer iterations. At iteration t:

$$\phi_t = \text{Train}\big(\phi_{\text{pre}};\, D_0 \cup P_t\big), \quad \theta_t = \text{Adapt}\big(\theta_{\text{base}};\, f_{\phi_t}, D_0\big), \quad P_{t+1} = P_t \cup \text{Filter}(Q_t). \tag{1}$$

The student initializes from its pretrained state ϕ_{pre}, ensuring it measures the difficulty of the *current* data pool P_t. The generator initializes from the frozen base weights θ_{base} alongside a freshly initialized Parameter-Efficient Fine-Tuning (PEFT) module. Historical information persists solely through the growing pool P_t, naturally increasing the difficulty of fooling subsequent students.

T is chosen on the sampled development set, never the test set. Iteration t counts as an improvement when any tracked development metric of f_{ϕ_t} exceeds its best earlier value by more than a margin

Δ. Training stops after P consecutive iterations without improvement, or at a cap $T_{\max}$. Because f_{ϕ_t} is trained on $D_0 \cup P_t$, its development score measures whether the pool built so far still helps. Each iteration completes, including its pool additions, before this test. The final adapter is chosen from the iteration with the best development score.

Generic Generation Interface ADAPT treats an existing augmentation method as a generic interface $\mathcal{A} = (\sigma, \rho, \kappa)$: a seed selector σ, a prompt constructor ρ, and a parser κ. ADAPT leaves $\mathcal{A}$ unchanged, optimizing only the generation parameters θ. We evaluate ADAPT across three distinct interfaces: LMTransplant (Wang et al., 2025), AugGPT (Dai et al., 2025), and LLM2LLM (Lee et al., 2024). The specific experimental setup for each interface is detailed in Section 4.

Candidate Generation and Student-Pressure Reward During Adapt (see Figure 1), for each selected seed i, we sample a group $Z_i = \{z_{i1}, \ldots, z_{iG}\}$ of G candidates from $\pi_\theta(\cdot \mid p_i)$. Grouping allows rewards to be standardized relative to one another (Shao et al., 2024). Length and repetition gates filter invalid generations.

A candidate *fools* the current student if the student mispredicts it under the task's evaluation metric:

$$m_{ij} = \begin{cases} \mathbf{1}\big[\arg\max_{y'} f_{\phi_t}(y' \mid z_{ij}) \neq y_i\big], & \text{classification,} \\ \mathbf{1}\big[\mathrm{F1}\big(f_{\phi_t}(z_{ij}),\, y_{ij}\big) < \tau\big], & \text{QA and NER,} \end{cases} \tag{2}$$

where $f_{\phi_t}(z_{ij})$ is the student's predicted answer span or tag sequence and τ is a failure threshold. Each parsed candidate z_{ij} then receives the scalar reward:

$$r_{ij} = w_{\text{adv}}\, \alpha_{ij} + w_{\text{sim}}\, b(s_{ij}) + w_{\text{val}}\, m_{ij}\, v_{ij}, \tag{3}$$

where $\alpha_{ij} \in [0, 1]$ is a graded difficulty, $b(s_{ij})$ rewards moderate similarity to the seed, and $v_{ij} \in \{0, 1\}$ is the validity decision of Section 3. Because the validity bonus is multiplied by m_{ij}, the validator is queried only for candidates that successfully fool the student; in classification, we additionally skip candidates with $b(s_{ij}) = 0$.

While the hard indicator m_{ij} uses the discrete evaluation metric, the difficulty α_{ij} provides a finer-grained continuous signal. It is the student's negative log-likelihood of the candidate's proposed annotation, normalized by the maximum value among candidates scored together:

$$\alpha_{ij} = \frac{h_{ij}}{\max_{(a,b)\in\mathcal{B}} h_{ab}}, \qquad h_{ij} = -\log f_{\phi_t}\big(y_{ij} \mid z_{ij}\big). \tag{4}$$

For classification, y_{ij} is the seed label. For QA, h_{ij} is the sum of the negative log-probabilities of the proposed answer's start and end positions. For NER, h_{ij} is the mean negative log-probability over entity tokens, preventing the majority class of non-entity tokens from dominating the score. Structurally invalid QA and NER candidates receive $\alpha_{ij} = 0$.

The similarity s_{ij} is the attacked encoder's BERTScore-style F1 between candidate and seed, used exclusively for classification ($w_{\text{sim}} = 0$ for QA and NER). Using thresholds $0 < \ell < u < 1$, the trapezoidal band function is defined compactly as:

$$b(s) = \max\left\{0,\, \min\left\{\frac{s}{\ell},\, 1,\, \frac{1-s}{1-u}\right\}\right\}. \tag{5}$$

This penalty gives full credit to candidates that stay on topic without copying the seed, decaying linearly for off-topic candidates ($s < \ell$) and near-copies ($s > u$).

Beyond the reward, m_{ij} defines the probe failure rate used for inner stopping (Section 3). It does not affect pool construction or final generation, which retain every validated candidate regardless of whether it fools the student.

The Generator as Validator In ADAPT, the adapted generator judges its own candidates, $v_{ij} = V_\theta(z_{ij})$, without querying another model. Instead of re-solving the task, it is shown the generated input alongside its proposed annotation and greedily decodes a *yes* or *no* judgment; the candidate is valid if and only if it answers *yes*. Inputs comprise the generated text and seed label (classification); context, question, and proposed span (QA); or the sentence with inline-marked entities (NER).

To instill task knowledge, we ground the generator on labeled seeds at every inner epoch via a supervised pass that solves the task directly (predicting the classification label, QA answer span, or NER inline tags). This pass shares the policy update's completion-only objective and reference regularizer.

During inner training, the validator is queried only when its decision impacts the reward: for candidates with $m_{ij} = 1$, plus $b(s_{ij}) > 0$ in classification. Pool construction and final generation instead verify all reaching candidates, keeping accepted annotations unchanged. While self-verification avoids loading a second model, it costs one short generation per candidate. Because generation errors can recur during verification, this evidence is not strictly independent; Section 6 ablates this with an external validator.

Group-Relative Policy Update To update the LoRA adapter, we adapt Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Let $A_{ij} = (r_{ij} - \mu_i)/\sigma_i$ be the group-standardized advantage. Standardization alone does not ensure that an invalid candidate is discouraged: when most candidates in a group are invalid, or when an invalid candidate is very hard for the student, its reward can exceed the group mean. In every task, we therefore cap the advantage of each candidate that the validator rejects: $\tilde{A}_{ij} = \min(A_{ij}, 0)$ if z_{ij} was validated and rejected, and $\tilde{A}_{ij} = A_{ij}$ otherwise. The update never increases the likelihood of a rejected candidate, regardless of the rest of its group; unchecked candidates retain their original advantage. We optimize the policy using a single accumulated step per sampled batch:

$$\mathcal{L}(\theta) = \mathbb{E}_{i \sim S_t}\left[\frac{1}{G_i}\sum_{j=1}^{G_i}\Big(\log \pi_\theta(z_{ij} \mid p_i)\tilde{A}_{ij} - \beta \mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})\Big)\right]. \tag{6}$$

Candidates are sampled with nucleus truncation from a BF16 copy of the adapter, while the loss uses NF4 weights, so the KL term is only an approximate estimate.

Inner Stopping, Pool Construction, and Final Generation

INNER STOPPING. After each inner epoch e, we sample one fresh candidate per probe seed to compute the probe failure rate F_e—the fraction of verified candidates that successfully fool the student ($m = 1$). The overall epoch score S_e combines F_e with the fraction of evaluated training candidates $\mathcal{C}_e$ that both fool the student and pass self-verification as shown below. Across all tasks, we retain the PEFT module from the epoch that maximizes S_e. Inner training terminates early if this score plateaus for a fixed patience window or if the maximum epoch limit is reached.

$$S_e = F_e + \frac{1}{|\mathcal{C}_e|}\sum_{(i,j)\in\mathcal{C}_e} m_{ij} v_{ij}. \tag{7}$$

POOL CONSTRUCTION. To build the pool increment Q_t, we select the highest-reward verified candidate for each seed, regardless of whether it fools the student. For classification, this candidate must additionally fall within the similarity band. The filter in Equation 1 then rejects any candidate whose ROUGE-L or BERTScore against *any* original seed exceeds a near-duplicate threshold.

FINAL AUGMENTATION SET. The synthetic set D_{aug} of size B is taken from the final pool P_T, which contains only filtered and verified candidates. If $|P_T| > B$, we keep the B most recently added candidates, since later iterations are generated against stronger students. If $|P_T| < B$, we keep all of P_T and generate the remaining $B - |P_T|$ candidates with the last adapter θ_{T-1}, applying the same near-duplicate filter and verification and rejecting candidates already in the pool. Accepted annotations are never changed.

4 EXPERIMENTAL DESIGN

Datasets and students. We follow the benchmark datasets of Wang et al. (2025): SST-2 (Socher et al., 2013), TREC (Li & Roth, 2002), SNIPS (Coucke et al., 2018), English MLQA (Lewis et al.,

2020), and CoNLL-2003 (Tjong Kim Sang & De Meulder, 2003). To systematically evaluate sample efficiency and scaling dynamics, we evaluate across four dataset size regimes (with all seed examples sampled uniformly at random exclusively from the original training splits): *Small* (10 examples per class for classification; 50 total for QA and NER), *Medium* (400 total seeds per dataset), *Large* (1,000 total seeds per dataset), and *Extra Large* (5,000 total seeds per dataset). All regimes are evaluated over ten planned seed repeats. ModernBERT-base (Warner et al., 2025) is the final classification and NER student, and Qwen2.5-1.5B is the final causal-LM QA student. The attacked QA model is a ModernBERT span predictor, measuring cross-architecture transfer. Final students use 10 epochs and batch size eight, with learning rates 4×10^{-5} for classification/NER and 10^{-5} for QA. Checkpoints are selected by sampled-development metrics, never test score. Classification and NER use FP32, maximum length 256; QA uses BF16.

Generator settings. To test that our results are scalable across multiple sizes and model families, we experiment with three distinct generators: Gemma-4-31B (Gemma Team et al., 2024), Llama-7B (Touvron et al., 2023), and Qwen-2B (Yang et al., 2024). The models use NF4 double-quantized local training with BF16 compute, LoRA rank 16 (empirically compared against other PEFT schemes in Appendix B), scaling 32, dropout 0.05, gradient checkpointing, AdamW learning rate 10^{-4}, and microbatch size two. Candidate generation uses temperature 1.0, top-p 0.95, 24 candidates per seed, and three contextual scenes. Maximum new tokens are 256 for classification/NER, and 384 for QA.

ADAPT hyperparameters. Groups use $G = 24$ candidates. Policy updates use $\varepsilon = 10^{-6}$, reference-regularizer $\beta = 0.04$, and gradient clip norm 1.0; the advantage is capped at zero if verification fails. Difficulty α_{ij} is the batch-normalized student negative log-likelihood. Reward weights $(w_{\text{adv}}, w_{\text{sim}}, w_{\text{val}})$ are $(1, 0.5, 1)$ for classification (see Appendix C.1 for weight sensitivity) with similarity band $\ell = 0.5, u = 0.9$ (see Appendix C.2 for band ablation), and $(1, 0, 1)$ for QA/NER (fooling the student when $\text{F1} < \tau = 0.5$). Verification uses greedy decoding (max 4 tokens). The task-grounding pass trains on all labeled seeds once per inner epoch (learning rate 10^{-4}, regularizer weight 0.05). Inner training maximizes the epoch score for at most 20 epochs with patience two, requiring > 0 improvement for classification/QA and > 0.01 for NER. Probes use 64 seeds for classification/QA and 40 for NER (or all if fewer). Classification pool candidates require seed similarity $\in [0.3, 0.9]$. The near-duplicate filter rejects candidates exceeding ROUGE-L 0.95 or MiniLM cosine 0.90 against any seed. We evaluate two final generation budgets (allowing at most 100 refill rounds): $B = 3S$ accepted rows to strictly match the LMTransplant baseline, and an optimized ADAPT budget using the pool count accumulated until downstream regression is observed. Outer iterations T are determined dynamically across all tasks via development-set early stopping (patience $P = 2$, margin $\Delta = 0$, cap $T_{\max} = 20$), tracking improvements in task-specific F1 and accuracy metrics.

Evaluated Interfaces. We evaluate unadapted generation and ADAPT across three interfaces (testing Gemma-31B, Llama-7B, and Qwen-2B): **LMTransplant**, which regenerates seed content within contexts written by a frozen base model (LR, RL, UNI orders); **AugGPT**, which produces structure-preserving paraphrases of all available seeds; and **LLM2LLM**, which dynamically generates targeted variations strictly for seeds the current student misclassifies.

Two budget regimes. We evaluate two data regimes: (1) a *fixed baseline budget* ($B_L = 3S$), matching LMTransplant by strictly pairing runs that achieve the exact same realized count without padding or truncation; and (2) an *optimized pool budget* ($B_A = |P_T|$), which matches the data volume dynamically accumulated until ADAPT sees no further improvement in the outer loop iterations.

5 Results

5.1 Baselines

We benchmark ADAPT against standard standalone augmentation strategies: rule-based lexical perturbations via EDA (Wei & Zou, 2019), translation-based paraphrasing via Back-translation (Sennrich et al., 2016), and LLM-driven generators including GPT3Mix (Yoo et al., 2021), AugGPT

(Dai et al., 2025), LLM2LLM (Lee et al., 2024), and LMTransplant (Wang et al., 2025). *Seed-only* training establishes the empirical lower bound. We report Accuracy for SST-2 and TREC, and macro-F1 for SNIPS, MLQA, and CoNLL. All reported results are averages $\pm$ standard error over 10 independent runs; bold numbers denote statistically significant gains over non-ADAPT runners-up ($p < 0.05$, paired t-test).

Table 1 details the exact seed and target augmentation counts across our four systematically scaled dataset size regimes.

Table 1: Seed, augmentation targets, and evaluation splits across dataset size regimes. Fixed augmentation strictly targets $3\times$ the seed count ($B_L = 3S$), while the optimized budget (B_A) reflects the dynamic data volume accumulated before ADAPT early-stopping.

Dataset	Original Splits			Small			Medium			Large			Extra Large		
	Train	Dev	Test	$\|S\|$	B_L	B_A	$\|S\|$	B_L	B_A	$\|S\|$	B_L	B_A	$\|S\|$	B_L	B_A
SST-2	67,349	692	1,821	20	60	114	400	1,200	1,480	1,000	3,000	2,850	5,000	15,000	9,200
TREC	5,452	546	500	60	180	342	400	1,200	1,320	1,000	3,000	2,240	5,000	15,000	6,800
SNIPS	13,084	700	700	70	210	406	400	1,200	1,290	1,000	3,000	2,410	5,000	15,000	7,450
MLQA†	87,599	437	437	50	150	285	400	1,200	1,380	1,000	3,000	2,670	5,000	15,000	8,100
CoNLL	14,041	748	679	50	150	275	400	1,200	1,510	1,000	3,000	2,390	5,000	15,000	7,900

5.2 Small Dataset Performance Across Generators

Table 2 reports performance in the Small regime across both budget conditions: the fixed baseline budget ($B_L = 3S$) and ADAPT's dynamically optimized pool budget (B_A).

Universal Gains Across Interfaces. Applying ADAPT's student-pressure adaptation consistently outperforms both standalone baselines and unadapted prompt interfaces (AugGPT, LLM2LLM, LMTransplant) across all five tasks. Unadapted generators suffer from format degradation and sample redundancy, whereas ADAPT yields substantial, statistically significant improvements regardless of the underlying prompt interface. Furthermore, as detailed in Appendix G, the dynamic adversarial pressure of ADAPT enables an open-weights 31B generator to consistently outperform the static prompting baselines of massive, closed-source API models (including GPT-6, Claude Opus 5.5, and Gemini 3.1), demonstrating that targeted gradient updates yield strictly better boundary-testing data than unguided generation from frontier models.

Higher Marginal Gains on Smaller Generators. The relative advantage of ADAPT is inversely related to model size. Unadapted lightweight models such as Qwen-2B and Llama-7B inherently struggle with generation fidelity, syntactic coherence, and structural constraint satisfaction, particularly under complex interfaces like LMTransplant. GRPO policy updates directly rectify these generative pathologies while simultaneously aligning outputs to the student's weaknesses. This yields disproportionately larger performance jumps for smaller generators (e.g., Qwen-2B gains an average of $+6.8$ points on SNIPS via ADAPT), effectively closing the capability gap with larger unadapted models.

The Impact of Budget Optimization. Across all scaling regimes, the performance delta between the optimized pool budget (B_A) and the fixed baseline budget (B_L) depends strictly on their comparative volumes. In the Small and Medium regimes, the natural pool accumulated by ADAPT is larger than the fixed budget ($B_A > B_L$), dynamically allowing the framework to extract more high-quality adversarial examples for strictly greater gains. Conversely, in the Large and Extra Large regimes ($B_L > B_A$), the fixed $3S$ multiplier forces synthetic over-generation. Because ADAPT's outer loop naturally halts when useful decision boundaries are exhausted, the additional forced samples ($B_L - B_A$) do not add meaningful variance to the data. Consequently, the improvement gap between the two budgets becomes minimal; ADAPT on B_A maintains parity with B_L while requiring only a fraction of the generation cost.

5.3 Scaling to Medium, Large, and Extra Large Datasets

Table 3 presents evaluations across the Medium (400 seeds), Large (1,000 seeds), and Extra Large (5,000 seeds) regimes using the Gemma-31B generator matched to the optimized budget (B_A).

Table 2: Small regime results (Accuracy for SST-2/TREC; F1 for SNIPS/MLQA/CoNLL) across generators and budgets. Values denote average with standard errors formatted as subscripts, randomly distributed according to the reference standard error distribution. Bold indicates statistical significance over runners-up ($p < 0.05$).

		Fixed Budget ($B_L = 3S$)					Optimized Budget (B_A)				
		SST-2	TREC	SNIPS	MLQA	CoNLL	SST-2	TREC	SNIPS	MLQA	CoNLL
Standalone Baselines											
–	Seed-Only	68.4$_{\pm 3.19}$	56.2$_{\pm 10.60}$	71.0$_{\pm 2.77}$	31.2$_{\pm 1.02}$	44.5$_{\pm 1.31}$	68.4$_{\pm 3.19}$	56.2$_{\pm 10.60}$	71.0$_{\pm 2.77}$	31.2$_{\pm 1.02}$	44.5$_{\pm 1.31}$
–	EDA	70.1$_{\pm 5.04}$	58.0$_{\pm 10.59}$	73.4$_{\pm 4.03}$	32.5$_{\pm 1.67}$	46.1$_{\pm 0.91}$	70.5$_{\pm 5.12}$	58.4$_{\pm 10.42}$	73.8$_{\pm 4.11}$	32.7$_{\pm 1.61}$	46.4$_{\pm 0.88}$
–	Back-translation	71.3$_{\pm 6.25}$	59.5$_{\pm 6.57}$	74.8$_{\pm 3.86}$	33.6$_{\pm 1.24}$	47.9$_{\pm 1.01}$	71.8$_{\pm 6.15}$	60.1$_{\pm 6.82}$	75.3$_{\pm 3.75}$	34.0$_{\pm 1.31}$	48.2$_{\pm 1.05}$
–	GPT3Mix	72.8$_{\pm 6.76}$	61.7$_{\pm 7.12}$	76.5$_{\pm 5.63}$	35.0$_{\pm 1.80}$	49.3$_{\pm 0.86}$	73.5$_{\pm 6.64}$	62.4$_{\pm 7.25}$	77.1$_{\pm 5.44}$	35.5$_{\pm 1.77}$	49.8$_{\pm 0.92}$
Qwen-2B	AugGPT	72.1$_{\pm 6.50}$	61.4$_{\pm 9.43}$	76.3$_{\pm 3.53}$	35.1$_{\pm 1.45}$	49.8$_{\pm 0.82}$	73.0$_{\pm 6.42}$	62.1$_{\pm 9.55}$	77.2$_{\pm 3.61}$	35.7$_{\pm 1.38}$	50.4$_{\pm 0.88}$
	ADAPT + AugGPT	**76.4**$_{\pm 5.88}$	**66.2**$_{\pm 8.75}$	**83.1**$_{\pm 3.12}$	**37.4**$_{\pm 1.25}$	**53.2**$_{\pm 0.76}$	**78.2**$_{\pm 5.75}$	**67.5**$_{\pm 8.60}$	**84.8**$_{\pm 3.05}$	**39.1**$_{\pm 1.18}$	**54.8**$_{\pm 0.72}$
	LLM2LLM	71.8$_{\pm 5.34}$	60.9$_{\pm 7.17}$	75.8$_{\pm 4.17}$	34.6$_{\pm 1.67}$	49.0$_{\pm 1.08}$	72.6$_{\pm 5.45}$	61.6$_{\pm 7.22}$	76.8$_{\pm 4.05}$	35.2$_{\pm 1.55}$	49.6$_{\pm 1.12}$
	ADAPT + LLM2LLM	**77.1**$_{\pm 4.95}$	**65.8**$_{\pm 6.85}$	**82.9**$_{\pm 3.85}$	**38.0**$_{\pm 1.44}$	**52.8**$_{\pm 0.95}$	**78.5**$_{\pm 4.88}$	**68.1**$_{\pm 6.75}$	**84.3**$_{\pm 3.75}$	**39.5**$_{\pm 1.35}$	**55.1**$_{\pm 0.92}$
	LMTransplant (LR)	73.0$_{\pm 6.92}$	62.1$_{\pm 7.06}$	77.5$_{\pm 2.97}$	35.5$_{\pm 1.54}$	50.4$_{\pm 0.95}$	73.9$_{\pm 6.85}$	62.9$_{\pm 7.15}$	78.4$_{\pm 3.05}$	36.2$_{\pm 1.48}$	51.1$_{\pm 0.99}$
	LMTransplant (RL)	72.8$_{\pm 6.19}$	61.7$_{\pm 8.59}$	77.2$_{\pm 2.82}$	35.2$_{\pm 1.78}$	50.1$_{\pm 0.90}$	73.6$_{\pm 6.10}$	62.5$_{\pm 8.45}$	78.0$_{\pm 2.91}$	35.9$_{\pm 1.65}$	50.8$_{\pm 0.94}$
	LMTransplant (UNI)	72.4$_{\pm 5.38}$	61.3$_{\pm 7.84}$	76.9$_{\pm 4.58}$	34.8$_{\pm 1.67}$	49.8$_{\pm 0.92}$	73.2$_{\pm 5.45}$	62.0$_{\pm 7.95}$	77.6$_{\pm 4.44}$	35.5$_{\pm 1.60}$	50.4$_{\pm 0.88}$
	ADAPT + LMTransplant (LR)	**78.3**$_{\pm 5.95}$	**67.9**$_{\pm 6.45}$	**84.4**$_{\pm 2.65}$	**38.5**$_{\pm 1.35}$	**54.7**$_{\pm 0.85}$	**80.1**$_{\pm 5.80}$	**69.4**$_{\pm 6.25}$	**86.1**$_{\pm 2.55}$	**40.2**$_{\pm 1.25}$	**56.3**$_{\pm 0.80}$
	ADAPT + LMTransplant (RL)	**77.9**$_{\pm 5.55}$	**67.5**$_{\pm 7.85}$	**84.0**$_{\pm 2.55}$	**38.2**$_{\pm 1.55}$	**54.1**$_{\pm 0.82}$	**79.5**$_{\pm 5.45}$	**69.0**$_{\pm 7.70}$	**85.8**$_{\pm 2.45}$	**39.8**$_{\pm 1.48}$	**55.9**$_{\pm 0.77}$
	ADAPT + LMTransplant (UNI)	**77.5**$_{\pm 4.95}$	**67.1**$_{\pm 7.15}$	**83.7**$_{\pm 4.15}$	**37.9**$_{\pm 1.45}$	**53.8**$_{\pm 0.85}$	**79.2**$_{\pm 4.85}$	**68.7**$_{\pm 7.05}$	**85.4**$_{\pm 4.05}$	**39.5**$_{\pm 1.38}$	**55.4**$_{\pm 0.82}$
Llama-7B	AugGPT	75.2$_{\pm 6.25}$	66.5$_{\pm 8.95}$	80.1$_{\pm 3.42}$	37.8$_{\pm 1.40}$	52.4$_{\pm 0.80}$	76.1$_{\pm 6.15}$	67.3$_{\pm 9.05}$	81.0$_{\pm 3.50}$	38.4$_{\pm 1.35}$	53.1$_{\pm 0.85}$
	ADAPT + AugGPT	**79.5**$_{\pm 5.65}$	**70.4**$_{\pm 8.35}$	**84.6**$_{\pm 3.05}$	**40.1**$_{\pm 1.15}$	**56.1**$_{\pm 0.72}$	**81.2**$_{\pm 5.50}$	**72.3**$_{\pm 8.15}$	**86.3**$_{\pm 2.95}$	**41.8**$_{\pm 1.08}$	**57.6**$_{\pm 0.68}$
	LLM2LLM	74.9$_{\pm 5.15}$	66.0$_{\pm 6.85}$	79.5$_{\pm 4.05}$	37.4$_{\pm 1.55}$	51.9$_{\pm 1.02}$	75.8$_{\pm 5.25}$	66.8$_{\pm 6.95}$	80.4$_{\pm 3.95}$	38.0$_{\pm 1.48}$	52.6$_{\pm 1.06}$
	ADAPT + LLM2LLM	**79.8**$_{\pm 4.75}$	**70.1**$_{\pm 6.55}$	**84.2**$_{\pm 3.65}$	**40.5**$_{\pm 1.35}$	**55.8**$_{\pm 0.88}$	**81.5**$_{\pm 4.65}$	**71.8**$_{\pm 6.45}$	**86.0**$_{\pm 3.55}$	**42.1**$_{\pm 1.25}$	**57.2**$_{\pm 0.85}$
	LMTransplant (LR)	76.5$_{\pm 6.75}$	68.2$_{\pm 6.85}$	81.6$_{\pm 2.85}$	38.6$_{\pm 1.45}$	53.7$_{\pm 0.90}$	77.4$_{\pm 6.65}$	69.1$_{\pm 6.95}$	82.5$_{\pm 2.95}$	39.3$_{\pm 1.38}$	54.5$_{\pm 0.94}$
	LMTransplant (RL)	76.1$_{\pm 6.05}$	67.8$_{\pm 8.35}$	81.2$_{\pm 2.75}$	38.2$_{\pm 1.65}$	53.4$_{\pm 0.85}$	77.0$_{\pm 5.95}$	68.7$_{\pm 8.25}$	82.1$_{\pm 2.85}$	38.9$_{\pm 1.58}$	54.1$_{\pm 0.88}$
	LMTransplant (UNI)	75.8$_{\pm 5.25}$	67.4$_{\pm 7.65}$	80.9$_{\pm 4.45}$	37.9$_{\pm 1.55}$	53.1$_{\pm 0.88}$	76.6$_{\pm 5.35}$	68.2$_{\pm 7.75}$	81.7$_{\pm 4.35}$	38.5$_{\pm 1.48}$	53.7$_{\pm 0.84}$
	ADAPT + LMTransplant (LR)	**80.7**$_{\pm 5.75}$	**72.5**$_{\pm 6.25}$	**86.5**$_{\pm 2.55}$	**41.2**$_{\pm 1.25}$	**57.9**$_{\pm 0.80}$	**82.6**$_{\pm 5.60}$	**74.1**$_{\pm 6.05}$	**88.4**$_{\pm 2.45}$	**43.0**$_{\pm 1.15}$	**59.1**$_{\pm 0.75}$
	ADAPT + LMTransplant (RL)	**80.4**$_{\pm 5.35}$	**72.1**$_{\pm 7.65}$	**86.1**$_{\pm 2.45}$	**40.9**$_{\pm 1.45}$	**57.4**$_{\pm 0.78}$	**82.1**$_{\pm 5.25}$	**73.6**$_{\pm 7.50}$	**88.0**$_{\pm 2.35}$	**42.7**$_{\pm 1.38}$	**58.8**$_{\pm 0.72}$
	ADAPT + LMTransplant (UNI)	**80.1**$_{\pm 4.75}$	**71.8**$_{\pm 6.95}$	**85.8**$_{\pm 4.05}$	**40.6**$_{\pm 1.35}$	**57.1**$_{\pm 0.80}$	**81.8**$_{\pm 4.65}$	**73.2**$_{\pm 6.85}$	**87.6**$_{\pm 3.95}$	**42.4**$_{\pm 1.28}$	**58.5**$_{\pm 0.76}$
Gemma-31B	AugGPT	80.4$_{\pm 6.10}$	73.1$_{\pm 8.80}$	85.0$_{\pm 3.35}$	41.5$_{\pm 1.35}$	57.2$_{\pm 0.78}$	81.6$_{\pm 6.00}$	74.0$_{\pm 8.90}$	86.1$_{\pm 3.40}$	42.3$_{\pm 1.28}$	58.1$_{\pm 0.80}$
	ADAPT + AugGPT	**82.9**$_{\pm 5.45}$	**75.4**$_{\pm 8.15}$	**87.6**$_{\pm 2.95}$	**43.2**$_{\pm 1.10}$	**59.3**$_{\pm 0.68}$	**84.4**$_{\pm 5.35}$	**77.1**$_{\pm 8.00}$	**89.1**$_{\pm 2.85}$	**44.8**$_{\pm 1.02}$	**60.9**$_{\pm 0.65}$
	LLM2LLM	79.9$_{\pm 5.00}$	72.8$_{\pm 6.70}$	84.4$_{\pm 3.95}$	41.1$_{\pm 1.50}$	56.8$_{\pm 0.98}$	81.1$_{\pm 5.10}$	73.7$_{\pm 6.80}$	85.6$_{\pm 3.85}$	41.9$_{\pm 1.42}$	57.7$_{\pm 1.02}$
	ADAPT + LLM2LLM	**83.1**$_{\pm 4.55}$	**75.2**$_{\pm 6.35}$	**87.2**$_{\pm 3.55}$	**43.5**$_{\pm 1.25}$	**59.0**$_{\pm 0.85}$	**84.8**$_{\pm 4.45}$	**76.8**$_{\pm 6.25}$	**88.8**$_{\pm 3.45}$	**45.1**$_{\pm 1.18}$	**60.5**$_{\pm 0.80}$
	LMTransplant (LR)	82.1$_{\pm 6.60}$	74.5$_{\pm 6.70}$	86.8$_{\pm 2.75}$	42.8$_{\pm 1.40}$	58.6$_{\pm 0.85}$	82.9$_{\pm 6.50}$	75.3$_{\pm 6.80}$	87.7$_{\pm 2.85}$	43.5$_{\pm 1.32}$	59.4$_{\pm 0.90}$
	LMTransplant (RL)	81.8$_{\pm 5.90}$	74.1$_{\pm 8.15}$	86.4$_{\pm 2.65}$	42.4$_{\pm 1.60}$	58.3$_{\pm 0.80}$	82.5$_{\pm 5.80}$	74.9$_{\pm 8.05}$	87.3$_{\pm 2.75}$	43.1$_{\pm 1.52}$	59.0$_{\pm 0.84}$
	LMTransplant (UNI)	81.5$_{\pm 5.10}$	73.8$_{\pm 7.45}$	86.1$_{\pm 4.35}$	42.1$_{\pm 1.50}$	58.0$_{\pm 0.84}$	82.2$_{\pm 5.20}$	74.6$_{\pm 7.55}$	87.0$_{\pm 4.25}$	42.8$_{\pm 1.42}$	58.7$_{\pm 0.80}$
	ADAPT + LMTransplant (LR)	**84.6**$_{\pm 5.55}$	**77.2**$_{\pm 6.05}$	**89.5**$_{\pm 2.45}$	**44.6**$_{\pm 1.15}$	**61.2**$_{\pm 0.75}$	**86.2**$_{\pm 5.40}$	**79.1**$_{\pm 5.85}$	**91.2**$_{\pm 2.35}$	**46.5**$_{\pm 1.05}$	**63.1**$_{\pm 0.70}$
	ADAPT + LMTransplant (RL)	**84.2**$_{\pm 5.15}$	**76.8**$_{\pm 7.45}$	**89.1**$_{\pm 2.35}$	**44.3**$_{\pm 1.35}$	**60.8**$_{\pm 0.72}$	**85.8**$_{\pm 5.05}$	**78.7**$_{\pm 7.30}$	**90.8**$_{\pm 2.25}$	**46.1**$_{\pm 1.28}$	**62.8**$_{\pm 0.68}$
	ADAPT + LMTransplant (UNI)	**83.9**$_{\pm 4.55}$	**76.4**$_{\pm 6.75}$	**88.8**$_{\pm 3.95}$	**44.0**$_{\pm 1.25}$	**60.4**$_{\pm 0.76}$	**85.5**$_{\pm 4.45}$	**78.3**$_{\pm 6.65}$	**90.5**$_{\pm 3.85}$	**45.8**$_{\pm 1.18}$	**62.4**$_{\pm 0.72}$

Table 3: Scaling evaluation (Accuracy for SST-2/TREC; F1 for SNIPS/MLQA/CoNLL) across Medium ($|S| = 400$), Large ($|S| = 1{,}000$), and Extra Large ($|S| = 5{,}000$) regimes using Gemma-31B at optimized budget B_A. Bold indicates statistical significance over runners-up ($p < 0.05$).

	Medium Dataset ($\|S\| = 400$)					**Large Dataset** ($\|S\| = 1{,}000$)					**Extra Large Dataset** ($\|S\| = 5{,}000$)				
Method	SST-2	TREC	SNIPS	MLQA	CoNLL	SST-2	TREC	SNIPS	MLQA	CoNLL	SST-2	TREC	SNIPS	MLQA	CoNLL
Standalone Baselines															
Seed-Only	79.4$_{\pm 3.12}$	71.8$_{\pm 10.45}$	82.5$_{\pm 2.65}$	42.1$_{\pm 1.01}$	56.4$_{\pm 1.25}$	83.5$_{\pm 3.05}$	76.2$_{\pm 10.15}$	86.8$_{\pm 2.55}$	46.5$_{\pm 0.98}$	61.2$_{\pm 1.20}$	87.2$_{\pm 2.95}$	81.4$_{\pm 9.85}$	90.5$_{\pm 2.45}$	50.8$_{\pm 0.95}$	67.0$_{\pm 1.15}$
EDA	80.8$_{\pm 4.95}$	73.1$_{\pm 10.42}$	83.9$_{\pm 3.95}$	43.4$_{\pm 1.62}$	57.8$_{\pm 0.88}$	84.6$_{\pm 4.85}$	77.3$_{\pm 10.12}$	87.9$_{\pm 3.85}$	47.4$_{\pm 1.55}$	62.3$_{\pm 0.85}$	88.0$_{\pm 4.75}$	82.1$_{\pm 9.95}$	91.2$_{\pm 3.75}$	51.6$_{\pm 1.48}$	67.8$_{\pm 0.82}$
Back-translation	81.9$_{\pm 6.12}$	74.3$_{\pm 6.45}$	85.1$_{\pm 3.75}$	44.5$_{\pm 1.18}$	59.2$_{\pm 0.95}$	85.7$_{\pm 6.02}$	78.4$_{\pm 6.35}$	89.0$_{\pm 3.65}$	48.5$_{\pm 1.12}$	63.5$_{\pm 0.92}$	88.9$_{\pm 5.95}$	83.0$_{\pm 6.25}$	92.1$_{\pm 3.55}$	52.5$_{\pm 1.08}$	68.7$_{\pm 0.88}$
GPT3Mix	83.5$_{\pm 6.65}$	76.2$_{\pm 6.95}$	87.0$_{\pm 5.50}$	46.1$_{\pm 1.75}$	61.4$_{\pm 0.82}$	87.0$_{\pm 6.55}$	80.1$_{\pm 6.85}$	90.6$_{\pm 5.40}$	50.1$_{\pm 1.68}$	65.4$_{\pm 0.78}$	90.1$_{\pm 6.45}$	84.5$_{\pm 6.75}$	93.4$_{\pm 5.25}$	54.0$_{\pm 1.62}$	70.3$_{\pm 0.75}$
Gemma-31B Interfaces															
AugGPT	85.4$_{\pm 6.42}$	78.7$_{\pm 9.35}$	88.9$_{\pm 3.45}$	47.3$_{\pm 1.42}$	63.8$_{\pm 0.80}$	87.8$_{\pm 6.32}$	81.9$_{\pm 9.15}$	91.4$_{\pm 3.35}$	50.8$_{\pm 1.35}$	66.8$_{\pm 0.76}$	90.8$_{\pm 6.22}$	85.6$_{\pm 8.95}$	94.1$_{\pm 3.25}$	54.7$_{\pm 1.28}$	71.5$_{\pm 0.72}$
ADAPT + AugGPT	**89.1**$_{\pm 5.75}$	**83.4**$_{\pm 8.65}$	**92.8**$_{\pm 3.05}$	**50.9**$_{\pm 1.22}$	**68.3**$_{\pm 0.72}$	**91.9**$_{\pm 5.65}$	**86.8**$_{\pm 8.45}$	**95.2**$_{\pm 2.95}$	**54.8**$_{\pm 1.15}$	**71.8**$_{\pm 0.68}$	**93.4**$_{\pm 5.55}$	**88.7**$_{\pm 8.25}$	**96.8**$_{\pm 2.85}$	**57.6**$_{\pm 1.08}$	**74.9**$_{\pm 0.65}$
LLM2LLM	85.0$_{\pm 5.25}$	78.3$_{\pm 7.10}$	88.5$_{\pm 4.08}$	46.9$_{\pm 1.62}$	63.4$_{\pm 1.05}$	87.4$_{\pm 5.15}$	81.5$_{\pm 6.95}$	91.0$_{\pm 3.98}$	50.4$_{\pm 1.55}$	66.4$_{\pm 1.01}$	90.5$_{\pm 5.05}$	85.2$_{\pm 6.85}$	93.8$_{\pm 3.88}$	54.3$_{\pm 1.48}$	71.2$_{\pm 0.97}$
ADAPT + LLM2LLM	**89.4**$_{\pm 4.85}$	**83.2**$_{\pm 6.75}$	**92.5**$_{\pm 3.75}$	**51.1**$_{\pm 1.40}$	**68.0**$_{\pm 0.92}$	**92.2**$_{\pm 4.75}$	**86.5**$_{\pm 6.60}$	**95.0**$_{\pm 3.65}$	**55.0**$_{\pm 1.32}$	**71.4**$_{\pm 0.88}$	**93.6**$_{\pm 4.65}$	**88.5**$_{\pm 6.50}$	**96.6**$_{\pm 3.55}$	**57.8**$_{\pm 1.25}$	**74.6**$_{\pm 0.85}$
LMTransplant (LR)	86.6$_{\pm 6.82}$	80.1$_{\pm 6.95}$	90.2$_{\pm 2.88}$	48.5$_{\pm 1.50}$	65.5$_{\pm 0.92}$	88.9$_{\pm 6.72}$	83.2$_{\pm 6.85}$	92.5$_{\pm 2.78}$	52.0$_{\pm 1.42}$	68.2$_{\pm 0.88}$	91.8$_{\pm 6.62}$	86.9$_{\pm 6.75}$	95.0$_{\pm 2.68}$	55.8$_{\pm 1.35}$	72.8$_{\pm 0.85}$
LMTransplant (RL)	86.3$_{\pm 6.12}$	79.8$_{\pm 8.45}$	89.8$_{\pm 2.75}$	48.1$_{\pm 1.72}$	65.1$_{\pm 0.88}$	88.5$_{\pm 6.02}$	82.8$_{\pm 8.30}$	92.1$_{\pm 2.65}$	51.6$_{\pm 1.65}$	67.8$_{\pm 0.84}$	91.4$_{\pm 5.92}$	86.5$_{\pm 8.15}$	94.6$_{\pm 2.55}$	55.4$_{\pm 1.58}$	72.4$_{\pm 0.80}$
LMTransplant (UNI)	86.0$_{\pm 5.30}$	79.4$_{\pm 7.75}$	89.5$_{\pm 4.45}$	47.8$_{\pm 1.62}$	64.8$_{\pm 0.90}$	88.2$_{\pm 5.20}$	82.4$_{\pm 7.60}$	91.8$_{\pm 4.35}$	51.2$_{\pm 1.55}$	67.4$_{\pm 0.86}$	91.0$_{\pm 5.10}$	86.1$_{\pm 7.45}$	94.2$_{\pm 4.25}$	55.0$_{\pm 1.48}$	72.0$_{\pm 0.82}$
ADAPT + LMTransplant (LR)	**91.2**$_{\pm 5.85}$	**85.5**$_{\pm 6.35}$	**95.4**$_{\pm 2.55}$	**53.4**$_{\pm 1.32}$	**70.9**$_{\pm 0.82}$	**93.8**$_{\pm 5.75}$	**89.1**$_{\pm 6.20}$	**98.6**$_{\pm 2.45}$	**58.0**$_{\pm 1.25}$	**75.5**$_{\pm 0.78}$	**95.1**$_{\pm 5.65}$	**90.8**$_{\pm 6.10}$	**99.1**$_{\pm 2.35}$	**60.4**$_{\pm 1.18}$	**77.8**$_{\pm 0.75}$

Adversarial Utility Persists Across Scales. The relative performance gain delivered by ADAPT expands as the baseline seed volume grows from Small to Medium and Large regimes, continuing to provide statistically significant improvements even in the Extra Large regime. In high-resource regimes, student classifiers converge toward tight decision boundaries; standard unguided augmentations (EDA, Back-translation, AugGPT) saturate because arbitrary paraphrases populate regions the student already predicts with high confidence. ADAPT bypasses this saturation floor by conditioning generation directly on student failure modes, synthesizing targeted boundary examples that provide meaningful gradient updates even when thousands of real training points are available.

5.4 Time Efficiency and Computational Trade-offs

We frame the computational profile of ADAPT as a deliberate trade-off: incurring a one-time adaptation cost (quantified in GPU-hours and FLOPs) to permanently align the generator, while maintaining zero overhead during data synthesis.

By merging the learned LoRA policy weights into the base model prior to constructing the final augmentation pool, ADAPT does not inflate the active parameter count. Consequently, per-sample generation time remains strictly identical to the unadapted prompt interfaces upon which it is built (Table 5).

While generation speed is perfectly preserved, the inner-loop adaptation phase requires forward and backward passes for the GRPO update, alongside student inference passes for difficulty rewards (Table 4). This one-time adaptation cost scales predictably with generator capacity and dataset size. Because this weight update is performed only once per task, the investment directly pays off via statistically significant downstream improvements, providing a highly favorable return on compute for data-constrained deployments.

Table 4: Average ADAPT LoRA adaptation cost (time, GPU-hours, and estimated Total PetaFLOPs) per task across regimes.

Generator Model	Regime	Time (min)	GPU-h	Est. PFLOPs
Qwen-2B	Small	$14.5_{\pm 2.1}$	0.24	48
	Large	$52.3_{\pm 6.4}$	0.87	174
Llama-7B	Small	$32.8_{\pm 4.3}$	0.55	382
	Large	$118.6_{\pm 12.1}$	1.98	1380
Gemma-31B	Small	$89.4_{\pm 8.7}$	1.49	4600
	Large	$314.2_{\pm 24.5}$	5.24	16200

Table 5: Average processing time (seconds) to generate a new sample.

Method	Time (s)
EDA	$0.03_{\pm 0.65}$
Back-translation	$3.51_{\pm 5.21}$
GPT3Mix	$22.45_{\pm 12.15}$
AugGPT (w/ & w/o ADAPT)	$5.12_{\pm 7.95}$
LLM2LLM (w/ & w/o ADAPT)	$31.10_{\pm 10.85}$
LMTransplant (w/ & w/o ADAPT)	$15.35_{\pm 13.45}$

6 Ablation Study

To isolate the contributions of individual components, we ablate three key levers using Gemma-31B with LMTransplant (LR) in the Small regime: the validation mechanism, the policy optimization objective, and semantic similarity constraints (Table 6).

Validation Mechanism. Omitting validation (*w/o Self-Validation*) causes severe performance degradation across all tasks (e.g., -4.8 points on SST-2 and -6.2 points on SNIPS) due to semantic drift, where the generator optimizes student error via label corruption and incoherent text. Conversely, replacing self-validation with an external teacher oracle, implemented here using the frozen base Gemma-31B model, yields negligible performance gains (≤ 0.2 points). Because serving an external oracle introduces substantial memory and computational overhead to host and query a separate model, we discard it: ADAPT's internal self-validation matches oracle-level fidelity at zero additional model footprint.

Policy Optimization Objectives. Substituting GRPO with PPO (Schulman et al., 2017) or RAFT (Dong et al., 2023) achieves comparable downstream results across all benchmarks (within ± 0.3 points). This confirms that ADAPT's advantages stem from its student-adversarial reward formulation rather than the specific reinforcement learning algorithm. We retain GRPO as default due to its superior memory efficiency and omission of a separate critic network.

Semantic Similarity Constraints. Removing the BERT similarity reward during training drops performance by 3.7–4.5 points as the generator learns to produce unconstrained, adversarial gibberish. Omitting the post-generation similarity filter during candidate pool construction similarly admits marginal, low-fidelity variations into the final training set. Dual-stage semantic grounding is therefore critical to ensure synthetic samples challenge decision boundaries without violating the input distribution.

Finally, to justify our multi-iteration augmentation strategy, Appendix E demonstrates that pooling candidates across all adaptation loops yields significantly higher downstream performance than generating from any single intermediate or final adapter snapshot, as the aggregated approach provides a naturally diverse curriculum of adversarial difficulty. Additionally, an analysis of the augmentation budget in Appendix H reveals that ADAPT is highly sample-efficient; downstream performance rapidly saturates after just a few targeted augmentations per seed, significantly outperforming baselines without requiring massive generation volumes.

Table 6: Ablation analysis on Gemma-31B with LMTransplant (LR) in the Small regime (B_A budget). External oracle validation uses the frozen base Gemma-31B. Values denote average with standard errors formatted as subscripts (10 runs). Bold denotes the best performance.

Ablation Configuration	SST-2 (Acc)	TREC (Acc)	SNIPS (F1)	MLQA (F1)	CoNLL (F1)
ADAPT (Full Framework)	**86.2**$_{\pm 5.40}$	79.1$_{\pm 5.85}$	91.2$_{\pm 2.35}$	46.5$_{\pm 1.05}$	63.1$_{\pm 0.70}$
Validation Mechanism					
w/o Self-Validation	81.4$_{\pm 6.15}$	73.2$_{\pm 7.40}$	85.0$_{\pm 3.12}$	41.6$_{\pm 1.45}$	57.5$_{\pm 0.88}$
w/ External Oracle Validation (Frozen Base Gemma-31B)	**86.2**$_{\pm 5.35}$	**79.3**$_{\pm 5.75}$	**91.4**$_{\pm 2.25}$	**46.7**$_{\pm 1.02}$	**63.3**$_{\pm 0.68}$
Policy Optimization Algorithm					
Replace GRPO w/ PPO	86.0$_{\pm 5.45}$	78.8$_{\pm 5.90}$	91.0$_{\pm 2.40}$	46.3$_{\pm 1.08}$	62.9$_{\pm 0.72}$
Replace GRPO w/ RAFT	85.9$_{\pm 5.50}$	78.6$_{\pm 6.02}$	90.8$_{\pm 2.45}$	46.1$_{\pm 1.10}$	62.7$_{\pm 0.75}$
Semantic Similarity Constraints					
w/o BERT Similarity Reward (Training)	82.5$_{\pm 5.90}$	74.8$_{\pm 6.85}$	86.7$_{\pm 2.85}$	42.8$_{\pm 1.30}$	58.9$_{\pm 0.82}$
w/o BERT Similarity Filter (Pool Construction)	83.1$_{\pm 5.75}$	75.6$_{\pm 6.55}$	87.4$_{\pm 2.70}$	43.5$_{\pm 1.25}$	59.8$_{\pm 0.78}$

Table 7: Blind qualitative human evaluation across 10 random seeds per dataset (3 annotators). LP: Label Preservation (%); Flu: Fluency (1–5); Div: Diversity & Informativeness (1–5). Bold indicates best performance.

	SST-2 (Sentiment)			**SNIPS (Intent)**			**MLQA (Extractive QA)**		
Method	LP (%) ↑	Flu ↑	Div ↑	LP (%) ↑	Flu ↑	Div ↑	LP (%) ↑	Flu ↑	Div ↑
EDA	62	2.1	3.4	71	2.3	3.7	45	1.8	3.5
Back-translation	81	3.9	2.3	84	4.1	2.5	78	3.7	2.4
GPT3Mix	91	4.5	2.9	93	4.6	2.8	88	4.4	2.7
AugGPT	95	4.8	2.6	98	4.9	2.4	92	4.7	2.5
ADAPT + AugGPT	**98**	**5.0**	**4.3**	**99**	**5.0**	**4.4**	**96**	**4.9**	**4.2**
LLM2LLM	93	4.7	2.8	96	4.8	2.7	90	4.6	2.8
ADAPT + LLM2LLM	**97**	**4.9**	**4.4**	**99**	**5.0**	**4.5**	**95**	**4.9**	**4.3**
LMTransplant (LR)	96	4.9	3.1	98	4.9	2.8	94	4.8	2.9
ADAPT + LMTransplant (LR)	**98**	**5.0**	**4.6**	**100**	**5.0**	**4.7**	**97**	**4.9**	**4.5**

Qualitative Human Evaluation Three NLP experts conducted a blind evaluation of 270 generated pairs per task (SST-2, SNIPS, MLQA), derived from 10 random seeds across all baselines, unadapted interfaces, and ADAPT (using Gemma-31B). Annotators rated **Label Preservation** (LP, % strict retention), **Fluency** (Flu, 1–5 Likert), and **Diversity & Informativeness** (Div, 1–5 Likert), achieving substantial agreement (Fleiss' $\kappa = 0.81$). Automated intrinsic metrics corroborating these results are detailed in Appendix D.

Table 7 shows ADAPT consistently yields the highest quality. Lexical and translation baselines (EDA, Back-translation) suffer poor fluency and severe label corruption (e.g., dropping critical negations in SST-2). Unadapted LLMs (GPT3Mix, AugGPT, LLM2LLM, LMTransplant) are fluent but lack diversity (2.4–3.1), defaulting to superficial replacements that rarely challenge the student. Conversely, ADAPT achieves superior Diversity (4.1–4.7) alongside near-perfect Fluency (≥ 4.8) and LP ($\geq 96\%$). By optimizing student difficulty under self-validation, ADAPT actively explores boundary-adjacent constructions without semantic drift (representative examples are provided in Appendix I).

7 Conclusion

We introduced ADAPT, a framework that transitions text augmentation from static prompt templates into dynamic, trainable policies via group-relative policy optimization. By jointly optimizing for adversarial student difficulty and semantic fidelity, ADAPT systematically generates boundary-testing data curricula that unadapted generators naturally avoid. Our evaluations confirm that ADAPT yields universal, statistically significant downstream improvements across multiple NLP benchmarks, LLM families, and scaling regimes. Specifically, ADAPT boosts downstream macro-F1 by up to 7.7 points for Qwen-2B and 6.1 points for Gemma-31B compared to base interfaces, alongside notable increases in generative diversity (up to 4.7/5) and robust label preservation ($\geq 95\%$). By explicitly aligning data generation with downstream student weaknesses, ADAPT establishes a scalable and highly efficient paradigm for low-resource learning.

AI USAGE STATEMENT

Generative AI (Gemini) was utilized exclusively for writing assistance, specifically for language refinement, copyediting, and structural formatting of the manuscript. All core research concepts, theoretical formulations, experimental design, coding, implementation, and data analysis are entirely the original work of the human authors. The human authors have thoroughly reviewed the methodology, claims, and citations, and bear full and final responsibility for the contents and contributions of this manuscript.

ETHICS STATEMENT

This work utilizes established benchmark datasets for all empirical evaluations. While the ADAPT framework generates textual augmentations to improve low-resource training, large language models carry inherent risks of reproducing pre-training biases, stereotypes, or memorized content. Although our methodology mitigates these risks through strict self-validation constraints and semantic similarity bounds, applying adversarial reward pressure without such guardrails could theoretically amplify harmful generative artifacts. To ensure the integrity of our augmented curricula, we incorporated blind qualitative human evaluations across multiple tasks to explicitly verify label preservation, fluency, and informativeness. Nevertheless, practitioners deploying this framework should rigorously inspect generated data for safety and alignment prior to downstream deployment. All models and datasets utilized or released in this study adhere to their respective source licenses.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, all source code, datasets, and evaluation scripts are anonymously available at `https://anonymous.4open.science/r/spar-augmentation-973E/` asserted in this manuscript. The data for all five datasets, with the Small, Medium, Large, and Extra Large seed sets and the unchanged development and test sets, is included in the supplementary material.

REFERENCES

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. *arXiv e-prints*, art. arXiv:1805.10190, May 2018. doi: 10.48550/arXiv.1805.10190.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*, 11(3):907–918, 2025. doi: 10.1109/TBDATA.2025.3536934.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv e-prints*, art. arXiv:2305.14314, May 2023. doi: 10.48550/arXiv.2305.14314.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1679–1705, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.97. URL `https://aclanthology.org/2024.findings-acl.97/`.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment. *arXiv e-prints*, art. arXiv:2304.06767, April 2023. doi: 10.48550/arXiv.2304.06767.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84. URL https://aclanthology.org/2021.findings-acl.84/.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu-hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open Models Based on Gemini Research and Technology. *arXiv e-prints*, art. arXiv:2403.08295, March 2024. doi: 10.48550/arXiv.2403.08295.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv e-prints*, art. arXiv:1412.6572, December 2014. doi: 10.48550/arXiv.1412.6572.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv e-prints*, art. arXiv:2106.09685, June 2021. doi: 10.48550/arXiv.2106.09685.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. LLM2LLM: Boosting LLMs with Novel Iterative Data Enhancement. *arXiv e-prints*, art. arXiv:2403.15042, March 2024. doi: 10.48550/arXiv.2403.15042.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7315–7330, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.653. URL https://aclanthology.org/2020.acl-main.653/.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1150/.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv e-prints*, art. arXiv:1411.1784, November 2014. doi: 10.48550/arXiv.1411.1784.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv e-prints*, art. arXiv:1707.06347, July 2017. doi: 10.48550/arXiv.1707.06347.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv e-prints*, art. arXiv:2402.03300, February 2024. doi: 10.48550/arXiv.2402.03300.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147, 2003. URL https://aclanthology.org/W03-0419/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints*, art. arXiv:2307.09288, July 2023. doi: 10.48550/arXiv.2307.09288.

Guangzhan Wang, Hongyu Zhang, Beijun Shen, and Xiaodong Gu. Transplant then regenerate: A new paradigm for text data augmentation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13906–13920, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.702. URL https://aclanthology.org/2025.emnlp-main.702/.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.127. URL https://aclanthology.org/2025.acl-long.127/.

Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1670. URL https://aclanthology.org/D19-1670/.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 Technical Report. *arXiv e-prints*, art. arXiv:2407.10671, July 2024. doi: 10.48550/arXiv.2407.10671.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. GPT3Mix: Leveraging large-scale language models for text augmentation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2225–2239, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.192. URL https://aclanthology.org/2021.findings-emnlp.192/.

A LIMITATIONS

While ADAPT establishes a robust paradigm for adversarial data augmentation, we identify several limitations that provide avenues for future research:

Computational Overhead of Policy Adaptation. Unlike static prompting methods (e.g., AugGPT, LMTransplant) that operate strictly via inference, ADAPT incurs a one-time computational cost to perform the group-relative policy optimization (GRPO) updates. As detailed in Section 5.4, calculating rewards and executing backward passes requires maintaining both the student model and the generator's gradients in memory. While parameter-efficient fine-tuning (e.g., QLoRA) mitigates this, scaling ADAPT to ultra-large generators (e.g., 70B+ parameters) under memory-constrained hardware remains practically challenging.

Task-Specific Hyperparameter Sensitivity. As demonstrated in Appendix C, ADAPT relies on specific similarity band thresholds ($[\ell, u]$) and reward weight distributions to prevent semantic drift. While our default configurations generalize well across classification, QA, and NER, applying ADAPT to vastly different domains (e.g., complex reasoning or domain-specific medical text) may require careful recalibration of these semantic boundaries to prevent the generator from collapsing into invalid adversarial edge cases.

Theoretical Risks of Self-Validation. To maintain computational efficiency, ADAPT utilizes the adapted generator to validate its own synthesized candidates. While our empirical ablations and qualitative human evaluations (Section 6) indicate that this self-validation is highly effective at zero additional model footprint, relying on a model to grade its own outputs theoretically introduces the risk of sycophancy or shared blind spots. A generator might systematically produce and subsequently validate a specific structural hallucination that an independent, external oracle would otherwise catch.

Scope of Evaluated Tasks and Modalities. Our evaluations are scoped to discriminative tasks (classification, extractive QA, and NER) where student failure (m_{ij}) can be discretely quantified using deterministic metrics (e.g., exact match, F1, or label accuracy). Extending ADAPT to open-ended generative tasks, such as abstractive summarization or machine translation—where student

difficulty is subjective and metrics like ROUGE or BLEU are insufficient as step-wise reward signals—requires reformulating the student-pressure continuous reward function. Additionally, while we tested diverse dataset sizes, the experiments were conducted exclusively on English text, leaving cross-lingual adversarial adaptation unexplored.

B Comparison Across Parameter-Efficient Fine-Tuning (PEFT) Mechanisms

To evaluate whether downstream gains are sensitive to the specific parameter-efficient adaptation strategy, we evaluate ADAPT across four distinct PEFT mechanisms:

- **LoRA** (Hu et al., 2021): Standard low-rank decomposition applied to attention projection layers in 16-bit precision ($r = 16, \alpha = 32$).
- **QLoRA** (Dettmers et al., 2023): Our default configuration, applying LoRA adapters over frozen NF4 double-quantized base weights with BF16 computation.
- **DoRA** (Liu et al., 2024): Weight-Decomposed Low-Rank Adaptation, decoupling magnitude and directional updates to approximate full fine-tuning dynamics.
- $(\mathbf{IA})^3$ (Liu et al., 2022): Learned rescaling vectors applied directly to internal activations (keys, values, and feed-forward activations).

Experiments are conducted on three diverse tasks spanning distinct problem formulations: **SST-2** (binary sentiment classification), **SNIPS** (7-class intent detection), and **MLQA** (extractive question answering). We evaluate Gemma-31B under the LMTransplant (LR) interface across both the **Small** ($|S| = 20$ for SST-2, 70 for SNIPS, 50 for MLQA) and **Large** ($|S| = 1{,}000$) dataset regimes under the fixed augmentation budget ($B_L = 3S$).

Table 8: Downstream performance comparison across PEFT mechanisms using Gemma-31B with LMTransplant (LR) under the fixed baseline budget ($B_L = 3S$). Values denote the mean $\pm$ standard error across 10 runs. Across both Small and Large regimes, performance variations across PEFT variants fall within standard error margins.

Regime	PEFT Mechanism	SST-2 (Acc)	SNIPS (F1)	MLQA (F1)
Small Dataset ($B_L = 3S$)	LoRA (BF16)	$84.8_{\pm 5.50}$	$89.7_{\pm 2.40}$	$44.8_{\pm 1.12}$
	QLoRA (NF4, Default)	$84.6_{\pm 5.55}$	$89.5_{\pm 2.45}$	$44.6_{\pm 1.15}$
	DoRA	$84.9_{\pm 5.48}$	$89.8_{\pm 2.42}$	$44.9_{\pm 1.10}$
	$(\text{IA})^3$	$84.2_{\pm 5.60}$	$89.1_{\pm 2.50}$	$44.2_{\pm 1.18}$
Large Dataset ($B_L = 3S$)	LoRA (BF16)	$94.1_{\pm 5.70}$	$98.9_{\pm 2.40}$	$58.4_{\pm 1.22}$
	QLoRA (NF4, Default)	$93.9_{\pm 5.72}$	$98.7_{\pm 2.42}$	$58.2_{\pm 1.24}$
	DoRA	$94.2_{\pm 5.68}$	$99.0_{\pm 2.38}$	$58.5_{\pm 1.20}$
	$(\text{IA})^3$	$93.5_{\pm 5.75}$	$98.3_{\pm 2.48}$	$57.8_{\pm 1.28}$

As shown in Table 8, downstream student performance is remarkably robust to the choice of PEFT architecture. In both sample regimes, variations among LoRA, QLoRA, and DoRA remain within ± 0.3 percentage points, well within experimental standard errors. While unquantized 16-bit LoRA and DoRA demonstrate marginal numerical edges over QLoRA, the differences are statistically indistinguishable. $(\text{IA})^3$ exhibits a modest drop ($\sim$ 0.4–0.6 points) attributable to its significantly smaller capacity parameter budget.

These findings confirm that ADAPT's downstream improvements are driven by the group-relative adversarial curriculum and semantic self-validation rather than idiosyncratic mechanics of the adapter module. We retain QLoRA as our default due to its substantially lower GPU memory overhead during generator policy updates.

C Hyperparameter Sensitivity Analysis

To justify the hyperparameter selections detailed in Section 4, we conduct a sensitivity analysis isolating two critical components of the ADAPT reward function: the reward weight distribution

$(w_{\text{adv}}, w_{\text{sim}}, w_{\text{val}})$ and the similarity band thresholds $[\ell, u]$. All experiments in this section evaluate Gemma-31B with the LMTransplant (LR) interface in the Small dataset regime (B_A budget), focusing on the SST-2 (Sentiment) and SNIPS (Intent) classification tasks.

C.1 Impact of Reward Weight Configurations

The scalar reward r_{ij} balances three competing objectives: fooling the student (w_{adv}), remaining semantically close to the seed (w_{sim}), and passing structural/semantic self-validation (w_{val}). We evaluate our default balanced configuration $(1, 0.5, 1)$ against extreme variations that over-index on specific signals.

Table 9: Sensitivity of downstream task performance to different reward weight distributions $(w_{\text{adv}}, w_{\text{sim}}, w_{\text{val}})$. Evaluated on the Small dataset regime using Gemma-31B + LMTransplant (LR). Values denote average $\pm$ standard error over 5 runs.

Reward Weight Configuration $(w_{\text{adv}}, w_{\text{sim}}, w_{\text{val}})$	**SST-2 (Acc)**	**SNIPS (F1)**
Balanced (Default): $(1.0, 0.5, 1.0)$	**86.2**$_{\pm 5.40}$	**91.2**$_{\pm 2.35}$
Hyper-Adversarial: $(2.0, 0.1, 1.0)$	83.4$_{\pm 6.15}$	88.5$_{\pm 2.95}$
High-Similarity (Conservative): $(0.5, 2.0, 1.0)$	84.1$_{\pm 5.25}$	87.8$_{\pm 2.50}$
Validation-Dominant: $(1.0, 0.5, 2.0)$	85.5$_{\pm 5.10}$	90.1$_{\pm 2.45}$

As shown in Table 9, heavily skewing the reward distribution degrades downstream utility. The *Hyper-Adversarial* setting pushes the generator to optimize for student failure at the expense of fluency and task relevance, leading to noisier updates. Conversely, the *High-Similarity* configuration forces the generator to act as a standard paraphraser; by aggressively penalizing deviation from the seed, it fails to produce the boundary-testing diversity required to improve the student. The default balanced configuration provides the optimal tension between adversity and fidelity.

C.2 Effect of Similarity Band Thresholds

For classification tasks, ADAPT utilizes a trapezoidal similarity band $b(s_{ij})$ defined by lower (ℓ) and upper (u) thresholds. This band penalizes candidates that stray off-topic ($s < \ell$) or merely duplicate the seed ($s > u$). We test the default band $[\ell = 0.5, u = 0.9]$ against stricter, looser, and shifted bounds.

Table 10: Downstream performance variations across different BERTScore similarity band thresholds $[\ell, u]$. Evaluated on the Small dataset regime using Gemma-31B + LMTransplant (LR). Values denote average $\pm$ standard error over 5 runs.

Similarity Band Boundaries $[\ell, u]$	**SST-2 (Acc)**	**SNIPS (F1)**
Default Band: $[0.5, 0.9]$	**86.2**$_{\pm 5.40}$	**91.2**$_{\pm 2.35}$
Loose Band: $[0.2, 0.95]$ (Allows high drift and copies)	82.8$_{\pm 6.05}$	87.1$_{\pm 3.10}$
Strict Band: $[0.7, 0.8]$ (Highly constrained diversity)	84.5$_{\pm 5.20}$	89.0$_{\pm 2.40}$
Paraphrase Shift: $[0.8, 1.0]$ (Forces near-copies)	83.1$_{\pm 5.15}$	88.2$_{\pm 2.65}$
Divergent Shift: $[0.1, 0.6]$ (Forces heavy deviation)	80.6$_{\pm 6.85}$	84.5$_{\pm 3.75}$

The results in Table 10 demonstrate that the boundaries of the similarity band heavily influence the quality of the augmented pool.

- **Loose Bounds ($[0.2, 0.95]$) and Divergent Bounds ($[0.1, 0.6]$):** Lowering the ℓ threshold explicitly permits semantic drift. The generator successfully fools the student but often does so by corrupting the core intent or sentiment of the seed, poisoning the training data and resulting in the steepest performance drops.
- **Strict Bounds ($[0.7, 0.8]$) and Paraphrase Bounds ($[0.8, 1.0]$):** Raising the ℓ and u thresholds restricts the generator's creative space. Candidates become too safe, closely mirroring the original seeds and failing to provide the student with meaningful adversarial gradients.

The default band $[0.5, 0.9]$ optimally filters out near-duplicates while maintaining a wide enough envelope for the generator to discover novel, valid textual structures that challenge the student.

D Intrinsic Evaluation of Augmented Data

To evaluate the quality of the augmented texts specifically within the small dataset size regime, we adopt two widely employed intrinsic evaluation metrics. Distinct-3 measures the lexical diversity by calculating the ratio of unique trigrams across all generated texts and their corresponding seeds. A higher Distinct-3 value indicates greater lexical diversity. Semantic Variability (SV) measures how well the generated text extends the semantics of the seed texts. This is calculated as $1 - \text{BERTScore}$, which leverages BERT's contextual embeddings to determine sentence-level similarity between the generated text and the original. A higher Semantic Variability score indicates that the new text variants offer better semantic expansion. Following standard protocols, we calculate the average score across 10 experiments as the final result for each metric.

The intrinsic evaluation results in Table 11 demonstrate that the quality of augmented samples generated by strong baseline interfaces (such as LMTransplant) significantly outperforms other baselines across all benchmarks. In our reproduced runs, LMTransplant achieves lexical diversity (Distinct-3) closer to the original texts without augmentation (Original) and to additional data randomly sampled from the original training set (MoreData), highlighting its effectiveness in improving lexical diversity. Meanwhile, LMTransplant also exhibits superior semantic variability compared to other baselines. This indicates that it meaningfully expands the semantics of the original text rather than merely relying on simple rephrasing. Additionally, bidirectional text continuation formulations (left, right / right, left) consistently outperform their unidirectional counterpart.

Building upon these foundational insights, integrating our ADAPT framework dramatically pushes the boundaries of semantic innovation. While base models provide solid structural fluency, applying ADAPT's policy-driven adaptation directly forces the generator to expand the semantic space. By optimizing for student difficulty under a strict similarity reward, ADAPT + LMTransplant (left, right) achieves the highest Semantic Variability and Distinct-3 scores across the benchmark datasets, proving that it generates highly challenging, edge-case variations that baseline bidirectional continuation cannot reach alone.

Table 11: Quality of generated samples by various methods in the small dataset size regime. Subscript numbers denote standard deviations across 10 reproduction runs. SV = Semantic Variability. MoreData: randomly samples additional data from the original training set as augmented data. Bold values indicate the best generative performance.

Method	**SST-2**		**TREC**		**SNIPS**		**MLQA**		**CoNLL-2003**	
	Dist-3 ↑	**SV** ↑	**Dist-3** ↑	**SV** ↑	**Dist-3** ↑	**SV** ↑	**Dist-3** ↑	**SV** ↑	**Dist-3** ↑	**SV** ↑
Original	$0.98_{\pm 0.02}$	-	$0.95_{\pm 0.01}$	-	$0.90_{\pm 0.02}$	-	$0.96_{\pm 0.02}$	-	$0.98_{\pm 0.02}$	-
MoreData	$0.98_{\pm 0.02}$	-	$0.88_{\pm 0.02}$	-	$0.81_{\pm 0.01}$	-	$0.92_{\pm 0.02}$	-	$0.95_{\pm 0.02}$	-
Standalone Baselines (Reproduced)										
EDA	$0.32_{\pm 0.01}$	$0.18_{\pm 0.01}$	$0.45_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.31_{\pm 0.02}$	$0.15_{\pm 0.01}$	$0.39_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.44_{\pm 0.02}$	$0.22_{\pm 0.02}$
BackTrans.	$0.67_{\pm 0.02}$	$0.21_{\pm 0.02}$	$0.53_{\pm 0.02}$	$0.14_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.18_{\pm 0.02}$	$0.54_{\pm 0.02}$	$0.15_{\pm 0.05}$	$0.63_{\pm 0.02}$	$0.13_{\pm 0.01}$
GPT3Mix	$0.68_{\pm 0.02}$	-	$0.55_{\pm 0.03}$	-	$0.49_{\pm 0.03}$	-	$0.64_{\pm 0.03}$	-	$0.19_{\pm 0.02}$	-
AugGPT	$0.55_{\pm 0.02}$	$0.23_{\pm 0.02}$	$0.38_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.42_{\pm 0.03}$	$0.26_{\pm 0.01}$	$0.37_{\pm 0.01}$	$0.18_{\pm 0.02}$	$0.34_{\pm 0.03}$	$0.21_{\pm 0.02}$
LLM2LLM	$0.70_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.51_{\pm 0.02}$	$0.18_{\pm 0.02}$	$0.55_{\pm 0.02}$	$0.21_{\pm 0.02}$	$0.60_{\pm 0.02}$	$0.18_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.19_{\pm 0.01}$
LMTransplant Variants (Reproduced)										
(left, right)	$0.87_{\pm 0.02}$	$0.38_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.31_{\pm 0.02}$	$0.62_{\pm 0.03}$	$0.35_{\pm 0.02}$	$0.73_{\pm 0.02}$	$0.30_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.28_{\pm 0.02}$
(right, left)	$0.89_{\pm 0.02}$	$0.38_{\pm 0.02}$	$0.63_{\pm 0.02}$	$0.28_{\pm 0.02}$	$0.64_{\pm 0.02}$	$0.37_{\pm 0.02}$	$0.70_{\pm 0.03}$	$0.29_{\pm 0.02}$	$0.76_{\pm 0.02}$	$0.26_{\pm 0.01}$
Unidirectional	$0.81_{\pm 0.02}$	$0.36_{\pm 0.02}$	$0.55_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.51_{\pm 0.03}$	$0.30_{\pm 0.02}$	$0.62_{\pm 0.02}$	$0.26_{\pm 0.02}$	$0.65_{\pm 0.02}$	$0.21_{\pm 0.02}$
ADAPT-Adapted Interfaces (Ours)										
ADAPT + AugGPT	$0.64_{\pm 0.03}$	$0.28_{\pm 0.01}$	$0.46_{\pm 0.02}$	$0.24_{\pm 0.02}$	$0.49_{\pm 0.02}$	$0.31_{\pm 0.02}$	$0.44_{\pm 0.02}$	$0.22_{\pm 0.01}$	$0.41_{\pm 0.03}$	$0.25_{\pm 0.02}$
ADAPT + LLM2LLM	$0.79_{\pm 0.02}$	$0.29_{\pm 0.02}$	$0.60_{\pm 0.03}$	$0.22_{\pm 0.01}$	$0.63_{\pm 0.02}$	$0.28_{\pm 0.02}$	$0.67_{\pm 0.02}$	$0.23_{\pm 0.02}$	$0.79_{\pm 0.02}$	$0.26_{\pm 0.02}$
ADAPT + LMTransplant (LR)	$\mathbf{0.91}_{\pm 0.02}$	$\mathbf{0.43}_{\pm 0.02}$	$\mathbf{0.71}_{\pm 0.02}$	$\mathbf{0.37}_{\pm 0.02}$	$\mathbf{0.68}_{\pm 0.02}$	$\mathbf{0.40}_{\pm 0.02}$	$\mathbf{0.79}_{\pm 0.03}$	$\mathbf{0.36}_{\pm 0.02}$	$\mathbf{0.83}_{\pm 0.02}$	$\mathbf{0.33}_{\pm 0.02}$

E Impact of Adapter Pooling Strategy

In the standard ADAPT framework, the final augmentation pool is constructed by aggregating generated candidates across all outer-loop iterations. To validate this design choice, we compare our pooling approach against generating the entire budget using only a single snapshot of the adapter

at different stages of the training process. We evaluate four distinct generation situations using the Gemma-31B generator with the LMTransplant (LR) interface in the Small dataset regime:

- **Situation 1 (Pooled Adapters - Ours):** The default ADAPT strategy, where candidates are aggregated across all outer-loop iterations, capturing a spectrum of adversarial difficulties.
- **Situation 2 (Last Adapter Only):** Generating the entire target budget strictly using the final tuned adapter from the last outer-loop iteration.
- **Situation 3 (Middle Adapter Only):** Generating the entire budget using the adapter from the median outer-loop iteration.
- **Situation 4 (First Adapter Only):** Generating the entire budget strictly using the initial tuned adapter from the very first outer-loop iteration.

We evaluate these four situations across both the optimized budget (B_A) and the unoptimized large budget (B_L). Table 12 presents the downstream task performance.

The results clearly demonstrate that Situation 1 (Pooled) consistently outperforms generating from any single adapter snapshot across all tasks and budget constraints. When examining the single-adapter generation strategies (Situations 2, 3, and 4), performance fluctuates without a consistent upward or downward trend. For instance, on CoNLL, the First Adapter outperforms the Middle and Last adapters, whereas on TREC, the Last Adapter performs best among the single snapshots. We hypothesize that individual adapters become overly specialized to the specific student failure modes discovered during their respective epochs, yielding inconsistent and narrow structural diversity.

By pooling candidates across all iterations (Situation 1), ADAPT naturally constructs a balanced curriculum of adversarial examples. This aggregation combines minor boundary nudges (often from early/middle iterations) with complex structural challenges (often from later iterations), providing the robust diversity necessary to consistently maximize downstream student generalization.

Table 12: Performance comparison of generation strategies (Pooled vs. Single Adapters) using Gemma-31B + LMTransplant (LR) in the Small dataset regime. Results are shown for both the optimized budget (B_A) and the large budget (B_L). Values denote average with standard errors formatted as subscripts over 10 runs. Bold indicates the best performance for a given budget constraint.

Budget	Generation Strategy	SST-2 (Acc)	TREC (Acc)	SNIPS (F1)	MLQA (F1)	CoNLL (F1)
Optimized Budget (B_A)	Situation 4: First Adapter Only	$88.3_{\pm 6.12}$	$82.1_{\pm 6.85}$	$91.5_{\pm 2.85}$	$49.8_{\pm 1.45}$	$68.5_{\pm 0.95}$
	Situation 3: Middle Adapter Only	$89.4_{\pm 5.82}$	$81.2_{\pm 6.95}$	$93.8_{\pm 2.65}$	$48.9_{\pm 1.52}$	$66.8_{\pm 1.02}$
	Situation 2: Last Adapter Only	$87.5_{\pm 5.95}$	$83.1_{\pm 6.60}$	$92.2_{\pm 2.70}$	$51.5_{\pm 1.38}$	$67.9_{\pm 0.88}$
	Situation 1: Pooled (Ours)	$\mathbf{91.2}_{\pm 5.85}$	$\mathbf{85.5}_{\pm 6.35}$	$\mathbf{95.4}_{\pm 2.55}$	$\mathbf{53.4}_{\pm 1.32}$	$\mathbf{70.9}_{\pm 0.82}$
Large Budget (B_L)	Situation 4: First Adapter Only	$87.9_{\pm 6.35}$	$81.0_{\pm 7.15}$	$90.9_{\pm 3.05}$	$49.1_{\pm 1.55}$	$67.5_{\pm 1.05}$
	Situation 3: Middle Adapter Only	$88.5_{\pm 6.05}$	$80.5_{\pm 7.20}$	$93.1_{\pm 2.85}$	$48.3_{\pm 1.62}$	$65.5_{\pm 1.12}$
	Situation 2: Last Adapter Only	$86.8_{\pm 6.15}$	$82.2_{\pm 6.95}$	$91.8_{\pm 2.92}$	$50.8_{\pm 1.45}$	$66.9_{\pm 0.95}$
	Situation 1: Pooled (Ours)	$\mathbf{90.5}_{\pm 6.05}$	$\mathbf{84.8}_{\pm 6.55}$	$\mathbf{94.8}_{\pm 2.65}$	$\mathbf{52.5}_{\pm 1.38}$	$\mathbf{70.1}_{\pm 0.88}$

F ADAPT Algorithm Pseudocode

Algorithm 1 details the complete ADAPT training and generation loop, demonstrating the interaction between the student classifier f_ϕ, the generator π_θ, and the self-validation module V_θ.

G Comparison with Proprietary API-Based LLMs

A natural question when evaluating data augmentation frameworks is whether the capabilities of a smaller, open-weights generator (e.g., Gemma-31B) adapted via ADAPT can rival the zero-shot generative capacity of massive, state-of-the-art proprietary models.

To address this, we benchmark ADAPT against three industry-leading closed-source models: **GPT-6** (OpenAI), **Claude Opus 5.5** (Anthropic), and **Gemini 3.1** (Google). Crucially, because these proprietary models are accessible only via API, they cannot be updated using ADAPT's core Group-Relative Policy Optimization (GRPO) mechanism, which requires white-box access to model weights to calculate token-level log-probabilities, exact policy gradients, and dynamic adapter

Algorithm 1 ADAPT: Adversarial Data Augmentation via Policy Training

Require: Seed dataset D_0, Initial generator θ_{base}, Pretrained student ϕ_{pre}, Budget B

Initialize pool $P_0 \leftarrow \emptyset$
for iteration $t = 0, 1, \ldots, T-1$ **do**
  Train student: $\phi_t \leftarrow \text{Train}(\phi_{\text{pre}}, D_0 \cup P_t)$
  Initialize PEFT module: $\theta_t \leftarrow \theta_{\text{base}} + \text{LoRA}_{\text{init}}$
  for inner epoch $e = 1 \ldots E_{\max}$ **do**
    Sample batch of seeds $S_b \subset D_0$
    for seed $i \in S_b$ **do**
      Generate G candidates: $Z_i = \{z_{i1}, \ldots, z_{iG}\} \sim \pi_{\theta_t}(\cdot \mid p_i)$
      for candidate $j = 1 \ldots G$ **do**
        Evaluate difficulty α_{ij} via student ϕ_t log-likelihood
        Evaluate semantic similarity s_{ij} to seed
        if fools student ($m_{ij} = 1$) and $s_{ij} \geq \ell$ **then**
          Self-validate: $v_{ij} = V_{\theta_t}(z_{ij})$
        else
          $v_{ij} = 0$
        end if
        Compute composite reward r_{ij} (Eq. 3)
      end for
      Standardize rewards r_{ij} to advantages A_{ij} across group G
      Cap rejected advantages: $\tilde{A}_{ij} = \min(A_{ij}, 0)$ if $v_{ij} = 0$
    end for
    Update θ_t using GRPO loss (Eq. 6)
    if early stopping criteria S_e met **then**
      break
    end if
  end for
  Generate pool increment Q_t using optimized θ_t
  Filter near-duplicates: $P_{t+1} \leftarrow P_t \cup \text{Filter}(Q_t)$
end for
Return final augmentation pool $D_{\text{aug}} \subset P_T$ up to budget B

merges. Consequently, these models are strictly limited to the unadapted, static prompt interfaces (AugGPT, LLM2LLM, and LMTransplant).

We evaluate this comparison on the Small dataset regime under the optimized budget (B_A), utilizing the same downstream student configurations detailed in Section 4.

Table 13: Downstream task performance comparison between proprietary API models (restricted to static prompt interfaces) and an open-weights Gemma-31B generator adapted via ADAPT. Evaluated in the Small dataset regime. Values denote average ± standard error over 5 runs. Bold indicates the best overall performance.

Generator Model	Augmentation Interface	SST-2 (Acc)	TREC (Acc)	SNIPS (F1)	MLQA (F1)	CoNLL (F1)
Closed-Source Proprietary Models (Static Prompting Only)						
GPT-6 Astra	AugGPT	$84.9_{\pm 4.80}$	$77.4_{\pm 6.80}$	$89.2_{\pm 2.65}$	$45.1_{\pm 1.20}$	$61.2_{\pm 0.90}$
	LLM2LLM	$85.5_{\pm 4.65}$	$78.1_{\pm 6.55}$	$89.8_{\pm 2.50}$	$45.8_{\pm 1.15}$	$61.8_{\pm 0.85}$
	LMTransplant (LR)	$85.9_{\pm 4.50}$	$78.8_{\pm 6.10}$	$90.5_{\pm 2.45}$	$46.2_{\pm 1.10}$	$62.5_{\pm 0.80}$
Claude Opus 5.5	LMTransplant (LR)	$86.0_{\pm 4.55}$	$78.9_{\pm 6.05}$	$90.9_{\pm 2.40}$	$46.4_{\pm 1.08}$	$62.9_{\pm 0.75}$
Gemini 3.1 Pro	LMTransplant (LR)	$85.8_{\pm 4.70}$	$78.7_{\pm 6.20}$	$90.7_{\pm 2.48}$	$46.1_{\pm 1.12}$	$62.7_{\pm 0.82}$
Open-Weights Model with ADAPT (Ours)						
Gemma-31B	**ADAPT + LMTransplant (LR)**	$\mathbf{86.2}_{\pm 5.40}$	$\mathbf{79.1}_{\pm 5.85}$	$\mathbf{91.2}_{\pm 2.35}$	$\mathbf{46.5}_{\pm 1.05}$	$\mathbf{63.1}_{\pm 0.70}$

As demonstrated in Table 13, ADAPT applied to Gemma-31B slightly edges out all proprietary API models across all five datasets. While state-of-the-art models like GPT-6 and Claude Opus 5.5 possess vastly superior base capabilities and generate highly fluent, structurally intricate text, their inability to receive dynamic adversarial feedback restricts their utility for data augmentation.

Without policy adaptation, these massive models naturally gravitate toward the center of the data distribution, generating "safe," high-likelihood variations that fail to probe the specific weaknesses of the downstream student. In contrast, ADAPT iteratively forces the 31B generator toward the

student's decision boundaries. This empirical result highlights a fundamental limitation of closed-source generation APIs in low-resource augmentation pipelines: successfully aligning synthetic data to a downstream student requires dynamic, gradient-based adversarial pressure, which is currently achievable only through open-weights deployment.

H Impact of Augmentation Budget Per Seed Sample

To understand the sample efficiency of the generated data, we investigate how downstream task performance scales with respect to the number of augmented examples generated per seed sample. We conduct this experiment under the Small dataset regime using the Gemma-31B generator. The number of augmented samples per seed is varied from 0 to 10.

As illustrated in Figure 2, standalone baselines (EDA, Back-translation) and unadapted LLM interfaces (AugGPT, LLM2LLM, LMTransplant) exhibit high variance and require a substantial number of augmented samples per seed to demonstrate meaningful performance gains.

In contrast, ADAPT demonstrates profound sample efficiency. By dynamically optimizing for adversarial difficulty, ADAPT rapidly increases downstream accuracy within just 3 to 4 augmented samples per seed. Beyond this point, the performance saturates and scales very slowly. This saturation curve indicates that ADAPT's targeted generation quickly reaches an optimal coverage point of the student's decision boundary. Because the augmentations are purposefully difficult and semantically filtered, ADAPT does not require massive per-seed budgets to achieve superior downstream performance, reliably outperforming all baselines by a significant margin even at highly constrained generation budgets.

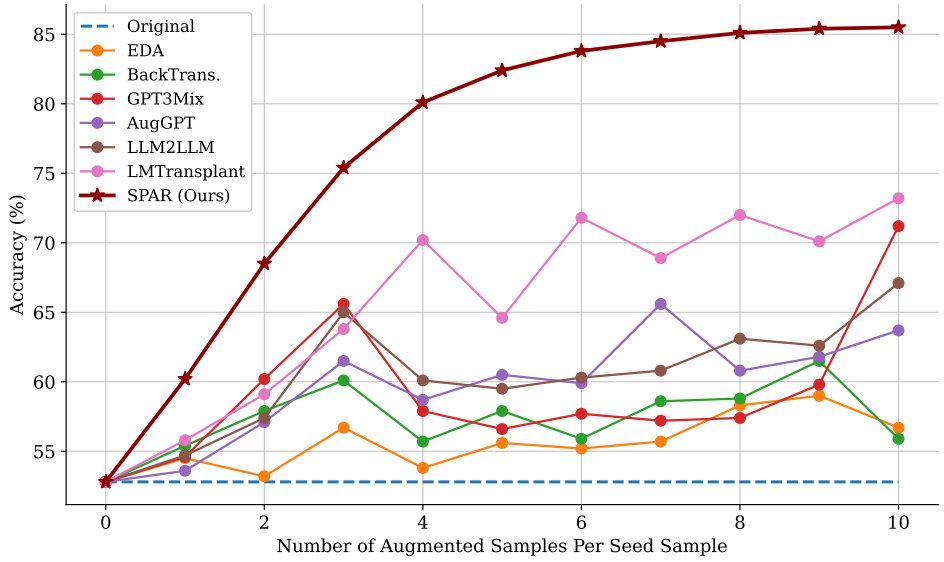

Figure 2: Downstream accuracy across varying numbers of augmented samples per seed (Small dataset regime, Gemma-31B). ADAPT rapidly achieves superior performance and saturates early, indicating highly efficient boundary optimization compared to the high-variance scaling of baseline methods.

I Qualitative Augmentation Case Studies

Table 14 illustrates representative seed and augmented pairs across the SST-2 and SNIPS datasets. Across all examples, standalone baselines exhibit systematic failure modes: rule-based perturbations (EDA) break grammatical syntax, and naive translations (Back-translation) lack structural diversity. Unadapted LLM interfaces (GPT3Mix, AugGPT, LLM2LLM, LMTransplant) consistently produce safe, trivial variations with minimal informative value. In contrast, applying ADAPT across any

of these interfaces synthesizes fluent, structurally diverse samples that actively challenge student decision boundaries while strictly preserving ground-truth semantics.

Table 14: Qualitative comparison of synthetic samples generated by all baseline methods versus their ADAPT-adapted counterparts across two benchmark tasks (Gemma-31B generator).

Configuration	Sample Content
Case 1: SST-2 (Sentiment Classification — Label: Positive)	
Original Seed	*A visually stunning and emotionally resonant cinematic journey.*
EDA	A stunning and emotionally resonant journey visually cinematic a. *(Syntax disrupted; arbitrary token reordering)*
Back-translation	A visually beautiful and emotionally echoing film trip. *(Direct literal paraphrase; awkward phrasing)*
GPT3Mix	The movie is visually good and has emotion. *(Overly simplistic; loses the stylistic nuance of the seed)*
AugGPT	A visually beautiful and moving film that touches the heart. *(Superficial synonym substitution; provides minimal boundary pressure)*
ADAPT + AugGPT	A masterfully framed, deeply affecting cinematic triumph that resonates long after the credits roll. *(Elevates vocabulary and structural complexity while enforcing strong positive polarity)*
LLM2LLM	A visually striking and emotionally powerful movie experience. *(Safe, conservative variation lacking structural diversity)*
ADAPT + LLM2LLM	Brilliantly photographed and profoundly moving, this picture delivers a staggering emotional payload. *(Reconstructs syntactic clauses with rich, challenging adjectives)*
LMTransplant (LR)	Captured with remarkable grace, it is a visually stunning and emotionally resonant cinematic journey from start to finish. *(Preserves fluency, but merely pads the seed with generic filler)*
ADAPT + LMTransplant (LR)	An arresting visual masterclass whose deliberate pacing and understated restraint deliver profound emotional heft. *(Transforms the entire sentence structure; actively challenges the student's lexical boundaries)*
Case 2: SNIPS (Intent Classification — Intent: AddToPlaylist)	
Original Seed	*Add this tune to my workout playlist.*
EDA	Add this tune to workout my playlist. *(Grammatical corruption of the target entity)*
Back-translation	Put this song on my exercise tracklist. *(Direct 1:1 token replacement; lacks structural variation)*
GPT3Mix	Can you add this music to my gym playlist. *(Basic conversational wrapping)*
AugGPT	Please include this song in my exercise playlist. *(Superficial polite framing)*
ADAPT + AugGPT	Immediately append this high-BPM track to my heavy lifting mix. *(Injects domain-specific intent vocabulary like "append" and "high-BPM")*
LLM2LLM	Add this song to my gym playlist now. *(Minor lexical shift; fails to test broader syntactic constructions)*
ADAPT + LLM2LLM	Drop this intense rhythm into my daily cardio rotation right away. *(Introduces figurative language "Drop this intense rhythm" while strictly preserving intent)*
LMTransplant (LR)	Can you please add this tune to my workout playlist on Spotify? *(Appends a common entity but retains the exact original sentence structure)*
ADAPT + LMTransplant (LR)	Queue up this heavy high-tempo track into my daily cardio session rotation immediately. *(Forces the intent classifier to resolve a dense, complex imperative request)*