IS HUMAN-READABLE TEXT NECESSARY FOR EFFECTIVE LLM FINE-TUNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Is human readability necessary for effective fine-tuning of large language models? We investigate whether model-conditioned training representations can preserve or improve adaptation utility without requiring a human-readable textual form. We propose Desired-Update-Aligned Synthetic Data (DASA), which uses activation-gradient feedback from a frozen reference model to guide the optimization of continuous synthetic input embeddings. Inspired by the role of activation gradients in local risk reduction, DASA targets useful adaptation updates rather than source-text reconstruction or linguistic fluency. The resulting embeddings are used directly for downstream fine-tuning; discrete token projections are employed only for qualitative inspection. Experiments on six models from the Llama and Qwen families, ranging from 1B to 32B parameters, cover six benchmarks spanning knowledge, mathematical reasoning, code generation, and commonsense reasoning. Under matched LoRA adaptation settings, DASA achieves performance comparable to the source natural-language data and surpasses it in multiple configurations, while outperforming GRADMM in most comparisons. Further experiments cover general-domain and task-specialized source data. Under the evaluated synthesis settings, DASA provides a 3.6–$4.9\times$ speedup over GRADMM with comparable peak GPU memory.

1 INTRODUCTION

Human readability and training utility represent two distinct properties of fine-tuning data for large language models (LLMs). Readability concerns whether the data are coherent and interpretable to humans, whereas training utility concerns whether they provide effective learning signals that improve model performance on held-out tasks. Although natural-language data can exhibit both properties simultaneously, their co-occurrence does not imply that one is necessary for the other. In particular, it remains unclear whether human-readable expressions are also required for models to benefit from fine-tuning. This raises a more fundamental question about the objective of data design: rather than treating human-readable language as a default representation, can we construct training data directly around the target model's learning requirements? This work studies the following question: Is human readability necessary for effective fine-tuning, or can training utility be preserved when linguistic form is substantially relaxed?

We focus on post-training adaptation of existing language models rather than pretraining from scratch. Concretely, given a reference model (f_{θ_0}) and a natural-language source dataset (D), we aim to construct an alternative dataset (D') that no longer relies on the conventional human-readable form of the original text, while still providing comparable adaptation gains under the same fine-tuning configuration and training budget. We further ask whether alternative representations constructed specifically for the receiving model can, in some settings, outperform the original natural-language data. Crucially, we define training utility by downstream performance after fine-tuning on independent evaluation tasks. We do not require the model to interpret, reconstruct, or translate the alternative sequences back into natural language. In this sense, the object of interest is not whether the representation remains semantically transparent to humans, but whether it serves as an effective substrate for model adaptation.

Prior work has shown that, under appropriate conditions, alignment between activation gradients and newly introduced residual directions can provide a basis for constructing local updates that

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.

reduce risk. Inspired by this perspective, we bring the idea of exploiting model-internal improvement signals into the construction of training data. Rather than requiring alternative data to preserve the human-readable form of the original text, we focus on whether it retains the effective signals needed for model adaptation. To this end, we use internal feedback produced by a reference model on the source data to guide the optimization of alternative training sequences. Our goal is to investigate whether natural-language data can be transformed into representations that are difficult for humans to read directly, while still retaining high utility for fine-tuning. This design allows us to test whether the adaptation value of data for an existing model can be preserved, or even improved, while substantially changing its surface form.

In this work, we propose Desired-Update-Aligned Synthetic Data (DASA), a method for constructing alternative training data guided by the alignment of model-internal signals. Given source data, DASA treats the internal signals induced in a reference model as optimization targets and updates alternative sequences through gradient-based optimization so that the signals they elicit align with the desired construction objective. Importantly, the procedure does not explicitly optimize for reconstructing the source text or producing fluent natural language. This allows the resulting sequences to depart substantially from the original word order, syntax, and surface form, yielding training representations that are difficult for humans to read directly. The central idea is not to treat unreadability itself as a mechanism for improving performance, but to relax constraints on linguistic form and shift the optimization objective toward internal signals useful to the model. At the same time, we do not assume that similarity in internal signals implies equivalence of the resulting training dynamics. Instead, we evaluate the validity of this construction criterion through the actual downstream performance obtained after fine-tuning on the alternative data.

We evaluate DASA on the Llama and Qwen model families, covering six models from 1B to 32B parameters across a diverse range of downstream tasks. Under a unified fine-tuning setup, DASA-generated alternative data yields adaptation gains comparable to those obtained with the original natural-language data and surpasses it on multiple model–task combinations. Compared with the gradient-matching baseline GRADMM, DASA performs better in the majority of evaluated settings while requiring less time for synthetic data generation. Additional experiments with both general-domain corpora and task-relevant data show that these advantages persist across different data sources, supporting the effectiveness and practicality of constructing training representations from model-internal signals.

Our main contributions are: ❶ We propose DASA, which constructs fine-tuning data by aligning model-internal signals without requiring text reconstruction or language fluency. ❷ We show that DASA preserves fine-tuning utility across model scales, tasks, and data budgets, and can outperform the original text in some settings. ❸ We evaluate DASA across different data sources, repeated runs, and generation costs to assess its effectiveness and practicality.

2 Related Work

Dataset distillation aims to preserve training utility with a small set of synthetic examples, and methods developed for images have since been extended to text (Wang et al., 2018; Zhao et al., 2021; Zhao & Bilen, 2021; 2023; Cazenavette et al., 2022; Nguyen et al., 2025). Text methods include continuous embedding optimization, discrete text generation, and training trajectory matching (Sucholutsky & Schonlau, 2021; Maekawa et al., 2024; Yao et al., 2025). GRADMM matches parameter gradients and, in its main procedure, projects optimized embeddings into low-perplexity text; it also reports an ablation that replaces top-k filtering with L_2 vocabulary projection (Nguyen et al., 2025).

Prior work has also explored fine-tuning with difficult-to-read instructions while retaining the original natural-language responses (Duan et al., 2025), learnable prompts (Shin et al., 2020; Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022), model-internal information for data synthesis (Yin et al., 2020; Wang et al., 2022), and theoretical analyses of local update directions (Cheng et al., 2026). Adapters and LoRA provide mechanisms for efficient model adaptation (Houlsby et al., 2019; Hu et al., 2022b). DASA constructs continuous synthetic inputs for subsequent fine-tuning by optimizing the first-order effect of their induced update direction on reference risk, without requiring source-text reconstruction or linguistic fluency. The residual expansion analysis motivates this objective but does not guarantee the outcome of finite-step fine-tuning.

3 METHODOLOGY

3.1 START POINT

The notation, basic setup, and standing assumptions used in this section are given in Appendix A; additional conditions are stated where needed. Appendix B provides the relevant definitions and derivations from this prior work.

At a fixed layer of the reference model, the activation gradient ζ measures the first-order sensitivity of the loss to a change in the hidden representation z. In their analysis of residual-block expansion, Cheng et al. (2026) consider a realizable parameter direction v whose first-order effect on population risk is

$$D_Q(v) = \frac{\mathrm{d}}{\mathrm{d}\eta}\mathbb{E}_Q\big[\ell\big(f_{\mathrm{top}}(z + h_{\eta v}(z)), y\big)\big]\Big|_{\eta=0} = \mathbb{E}_Q\big[\zeta^\top u_v(z)\big]\,. \tag{1}$$

Their Condition 1 requires the existence of such a direction with $D_Q(v) < 0$. Under the stated differentiability and realizability conditions, their Theorem 4 then establishes that, for some sufficiently small $\eta > 0$,

$$\mathbb{E}_Q\big[\ell\big(f_{\mathrm{top}}(z + h_{\eta v}(z)), y\big)\big] < \mathbb{E}_Q\big[\ell\big(f_{\mathrm{top}}(z), y\big)\big]\,. \tag{2}$$

This is an existence result for the expanded model class; it does not assert that a particular dataset or optimizer will find the improving direction. Their Theorem 5 additionally bounds the probability of nonpositive alignment between independent training and test activation-gradient means under a nonzero population mean and covariance control.

These results motivate a reverse data-construction question: given a target model and reference data, can we construct synthetic inputs whose induced update direction is locally improving on the reference set? DASA turns this directional criterion into a data-construction objective by optimizing continuous synthetic inputs Z^s such that the parameter-space update $v(Z^s)$ they induce produces a favorable first-order change in the reference loss.

For the reference set $\mathcal{D}_{\mathrm{ref}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{ref}}}$, we compute and cache

$$z_i = f_{\mathrm{bot}}(x_i), \qquad \zeta_i = \nabla_{z_i}\ell(f_{\mathrm{top}}(z_i), y_i).$$

Replacing the population expectation in $D_Q(v)$ with the empirical average gives the first-order change in reference loss along v:

$$D_{\mathcal{D}_{\mathrm{ref}}}(v) = \frac{1}{N_{\mathrm{ref}}}\sum_{i=1}^{N_{\mathrm{ref}}} \zeta_i^\top u_v(z_i). \tag{3}$$

At each synthesis step, the current synthetic examples and their retained prediction targets induce a parameter-space direction $v(Z^s)$. We then evaluate the first-order effect of this direction on the fixed reference examples and optimize

$$\boxed{\min_{Z^s} D_{\mathcal{D}_{\mathrm{ref}}}\big(v(Z^s)\big) = \min_{Z^s} \frac{1}{N_{\mathrm{ref}}}\sum_{i=1}^{N_{\mathrm{ref}}} \zeta_i^\top u_{v(Z^s)}(z_i).} \tag{4}$$

Gradients of this objective are propagated through $v(Z^s)$ to update Z^s. Thus, DASA directly optimizes the directional derivative of the reference loss induced by the candidate synthetic inputs, rather than a distance between activation-gradient means. The relation between a LoRA parameter direction and $u_v(z)$ is detailed in Appendix C. Under the stated conditions, a negative objective value indicates that the induced update direction is locally improving on the reference set; downstream performance after finite-step adaptation is evaluated empirically.

3.2 ALGORITHM DESIGN

DASA takes a frozen pretrained model f_{θ_0}, a finite reference set $\mathcal{D}_{\mathrm{ref}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{ref}}}$, a target layer l^*, and a synthetic budget of $n \times L$. Its output is a continuous input tensor for subsequent fine-tuning, rather than a set of natural-language sentences. We denote the tensor being optimized by $Z^s \in \mathbb{R}^{n\times L\times d_{\mathrm{emb}}}$, representing n synthetic sequences of L continuous input embeddings each. Let

Algorithm 1 DASA: Desired-Update-Aligned Synthetic Data

Require: Frozen model f_{θ_0}, reference dataset $\mathcal{D}_{\mathrm{ref}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{ref}}}$, target layer l^*, synthetic budget $n \times L$, steps T_{syn}, learning rate η_{syn}
Ensure: Continuous synthetic inputs S_{syn} and shifted-token targets Y^s

for $i = 1, \ldots, N_{\mathrm{ref}}$ **do**
  Compute $z_i \leftarrow f_{\mathrm{bot}}(x_i)$ and $\zeta_i \leftarrow \nabla_{z_i} \ell(f_{\mathrm{top}}(z_i), y_i)$.
end for
Define $D_{\mathcal{D}_{\mathrm{ref}}}(v) \coloneqq \frac{1}{N_{\mathrm{ref}}} \sum_{i=1}^{N_{\mathrm{ref}}} \zeta_i^\top u_v(z_i)$.
Select n length-L reference token sequences $X^{(0)}$.
Initialize Z^s from the input embeddings of $X^{(0)}$ and set $Y^s \leftarrow \mathrm{Shift}(X^{(0)})$.
for $t = 1, \ldots, T_{\mathrm{syn}}$ **do**
  Compute the tangent direction $v \leftarrow v(Z^s)$ clip its ℓ_2 norm to at most 2.0.
  Evaluate $D_{\mathcal{D}_{\mathrm{ref}}}(v)$.
  Update $Z^s \leftarrow Z^s - \eta_{\mathrm{syn}} \nabla_{Z^s} D_{\mathcal{D}_{\mathrm{ref}}}\big(v(Z^s)\big)$.
end for
$S_{\mathrm{syn}} \leftarrow \mathrm{BF16}(Z^s)$.
return (S_{syn}, Y^s)

$X^{(0)} = \{x_j^{(0)}\}_{j=1}^{n}$ denote the length-L reference token sequences used for initialization. We initialize Z_0^s from their input embeddings and set $Y^s = \mathrm{Shift}(X^{(0)})$, where $y_{j,t}^s = x_{j,t+1}^{(0)}$ for $t = 1, \ldots, L-1$. These are the standard shifted-token targets of causal language modeling, not separately annotated labels. For clarity, Algorithm 1 presents the single-layer case; in experiments, we apply the same procedure to all Transformer layers, clipping the induced direction separately at each layer.

Synthetic data generation. Algorithm 1 summarizes the synthesis procedure. We first pass the reference data through the frozen model and compute their activation gradients at the target layer. The synthetic activation gradients determine the induced direction $v(Z^s)$. To prevent the objective from being reduced merely by increasing the magnitude of the induced update, we clip its ℓ_2 norm to a maximum value of 2.0. We then evaluate the first-order effect of $\tilde{v}(Z^s)$ on the reference risk using the objective defined in Section 3.1. This objective is backpropagated to Z^s, which is updated with SGD. The model parameters and prediction targets remain fixed throughout synthesis. After optimization, we convert Z^s to BF16 to obtain the continuous synthetic data S_{syn}.

Downstream adaptation and evaluation. We pair S_{syn} with the retained prediction targets and use the continuous embeddings directly for fine-tuning, without projecting them to discrete tokens. At this stage, the synthetic data remain fixed while LoRA updates the model parameters. We evaluate the adapted model on independent tasks and compare it with fine-tuning on the original text under the same adaptation configuration and training budget.

4 Experiments

We evaluate DASA across multiple model scales, downstream tasks, calibration budgets, and data sources. Beyond overall adaptation performance, we examine robustness across random seeds, computational efficiency, sensitivity to optimization settings, and controlled baselines that isolate the effect of DASA's update-alignment objective. These experiments are designed to assess both the effectiveness and robustness of model-internal-signal-driven synthetic data construction.

4.1 Experimental Setup

Models and Benchmarks. We evaluate our method on six instruction-tuned language models spanning two model families and multiple scales: Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct (Meta, 2024), Llama-3.1-8B-Instruct (Team, 2024), and Qwen3-8B/14B/32B (Team, 2025). We consider six downstream benchmarks, including MMLU (Hendrycks et al., 2021a), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), MBPP (Austin et al., 2021), ARC-Challenge (ARC-C) (Clark et al., 2018), and CommonsenseQA (CSQA) (Talmor et al., 2019).

Baselines. We compare against three baselines. **Base** denotes the original model without downstream adaptation. **Real-text** fine-tunes the model using natural-language examples under the same data budget as our synthetic data. **GRADMM** (Nguyen et al., 2025) constructs synthetic training examples through parameter-gradient matching. In contrast, **Ours** directly optimizes synthetic sequences such that the induced activation-gradient update aligns with the desired update direction. Throughout all experiments, including the computational-efficiency comparison, both DASA and our implementation of GRADMM use all Transformer layers and pass the optimized continuous embeddings directly to downstream LoRA adaptation without vocabulary projection. Real-text uses the same sampled reference sequences $X^{(0)}$ that initialize DASA.

LoRA Adaptation and Data Budget. We freeze the pretrained backbone and optimize LoRA parameters only. All methods use identical LoRA (Hu et al., 2022a), optimization, and evaluation settings. We consider three data budgets, 50×2048, 100×2048, and 200×2048, where the first term is the number of sequences and 2048 is the maximum sequence length.

Evaluation protocol. We use accuracy for MMLU, ARC-Challenge, and CommonsenseQA; exact-match accuracy for GSM8K and MATH after standard answer extraction, and pass@1 for MBPP using execution-based functional correctness. All methods are evaluated with identical prompts, few-shot demonstrations, decoding parameters, and evaluation scripts. Generative tasks use deterministic decoding, so differences across methods arise from adaptation rather than sampling variation.

4.2 Main Results

Table 1 reports results across six models, six tasks, and three calibration budgets. DASA consistently improves over both the unadapted model and the parameter-gradient-matching baseline across model families and scales. In particular, compared with GRADMM, DASA achieves higher performance in 31 of 36 model–task configurations under the 50×2048 budget, 35 of 36 under 100×2048, and 35 of 36 under 200×2048. The advantage therefore becomes more consistent as the available synthetic-data budget increases rather than disappearing once additional training examples are introduced.

Effectiveness across model scales. The improvements are observed from 1B-scale models to Qwen3-32B. Under the 200×2048 budget, for example, DASA improves over GRADMM on six of six tasks for Llama-3.2-1B and Llama-3.1-8B, while outperforming GRADMM on all six benchmarks for both Qwen3-14B and Qwen3-32B. On Qwen3-32B, DASA improves MMLU from 84.65 to 85.26, GSM8K from 96.36 to 97.19, MATH from 87.56 to 88.28, ARC-C from 71.76 to 72.44, and CSQA from 85.83 to 86.65. The same qualitative behavior on substantially different model scales suggests that activation-level update realization is not tied to a particular parameter regime.

Effectiveness across capabilities. DASA also provides gains across qualitatively different task families. On Llama-3.2-3B with the largest budget, DASA improves MATH from 54.12 to 55.16 and MBPP from 71.2 to 72.0 over GRADMM, while improving MMLU from 61.20 to 61.96 and CSQA from 76.41 to 77.23. Similarly, on Qwen3-8B, it improves MATH from 86.44 to 87.40, MBPP from 62.2 to 63.0, and ARC-C from 66.30 to 67.24. These results indicate that the gains are not concentrated in one evaluation category, but extend to knowledge, reasoning, coding, and commonsense tasks.

Effect of the calibration budget. Increasing the calibration budget generally strengthens downstream adaptation for all synthesis methods, but DASA benefits particularly consistently from additional calibration data. Its relative number of wins over GRADMM increases from 31/36 to 35/36 as the budget grows from 50×2048 to 200×2048. This observation is important because a synthetic-data method may perform well at extremely small budgets simply by exploiting a few favorable examples. Instead, DASA continues to produce useful updates as the synthetic set grows, suggesting that activation alignment provides a stable optimization target beyond the low-data regime.

An additional feature of this experiment is that all downstream tasks use calibration samples drawn from the same cleaned C4 corpus. The model is therefore not simply trained on natural examples drawn from the target benchmark distribution. Nevertheless, the induced updates consistently improve task performance, supporting the central premise that the update signal itself, rather than direct semantic correspondence between calibration text and evaluation examples, can drive useful downstream adaptation.

Table 1: **Main results across models and tasks.** Base, Real-text, GRADMM, and DASA under three C4 calibration budgets; all adaptation methods use identical LoRA and optimization settings.

	Token Budget: 50×2048																	
	Llama-3.2-1B						**Llama-3.2-3B**						**Llama-3.1-8B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	45.79	46.32	32.70	49.6	54.61	59.13	58.62	82.03	51.40	68.4	73.72	73.87	65.41	85.90	50.70	75.4	82.08	75.92
Real-text	45.91	46.55	32.96	50.0	**54.86**	59.30	58.76	82.18	51.60	68.8	73.81	**74.20**	65.45	**86.20**	50.88	75.6	82.34	76.00
GRADMM	46.10	47.08	33.40	50.4	54.35	59.54	58.98	82.56	51.20	69.0	74.06	74.20	65.72	86.15	51.16	76.0	82.42	76.17
Ours	**46.46**	**46.63**	**34.20**	**50.8**	54.69	**59.95**	**59.22**	**82.87**	**51.66**	**69.6**	**74.32**	74.04	**65.56**	86.05	**51.56**	**76.4**	**82.68**	**76.41**
	Qwen3-8B						**Qwen3-14B**						**Qwen3-32B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	74.90	91.89	83.98	59.6	64.33	81.98	78.85	93.86	84.76	64.6	66.98	81.00	82.11	94.54	85.26	65.8	69.28	84.03
Real-text	75.05	92.04	84.12	59.8	**64.68**	81.98	78.93	**94.16**	84.88	64.8	67.15	**81.33**	**82.30**	94.69	85.34	66.2	69.37	84.28
GRADMM	75.24	92.27	84.32	60.2	64.68	82.23	79.04	93.31	85.10	65.0	67.32	81.07	82.11	94.69	85.48	66.2	69.54	84.28
Ours	**75.49**	**92.49**	**84.60**	**60.8**	64.59	**82.31**	**79.08**	94.09	**85.30**	**65.6**	**67.49**	81.24	82.24	**94.77**	**85.64**	**66.6**	**69.71**	**84.52**
	Token Budget: 100×2048																	
	Llama-3.2-1B						**Llama-3.2-3B**						**Llama-3.1-8B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	45.79	46.32	32.70	49.6	54.61	59.13	58.62	82.03	51.40	68.4	73.72	73.87	65.41	85.90	50.70	75.4	82.08	75.92
Real-text	46.05	**46.93**	33.12	50.0	**54.86**	59.38	58.83	82.34	**51.72**	68.8	73.89	74.12	**65.59**	86.20	51.00	75.6	82.25	76.09
GRADMM	46.44	46.69	33.88	50.8	54.29	59.79	59.20	82.94	51.32	69.4	74.23	74.45	65.11	86.66	51.48	76.2	82.51	76.41
Ours	**47.21**	46.78	**35.02**	**52.0**	54.69	**60.69**	**59.91**	**84.00**	51.60	**70.8**	**75.00**	**75.18**	65.50	**87.49**	**52.38**	**77.4**	**83.19**	**77.07**
	Qwen3-8B						**Qwen3-14B**						**Qwen3-32B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	74.90	91.89	83.98	59.6	64.33	81.98	78.85	93.86	84.76	64.6	66.98	81.00	82.11	94.54	85.26	65.8	69.28	84.03
Real-text	75.07	92.12	84.18	59.8	64.51	**82.15**	**79.00**	94.01	84.92	64.8	67.15	81.16	82.23	**94.69**	85.40	66.0	69.37	84.11
GRADMM	75.37	92.42	84.56	60.4	64.76	82.09	78.27	94.31	85.24	65.4	67.41	81.41	82.45	94.02	85.64	66.4	69.62	84.36
Ours	**75.98**	**93.10**	**85.24**	**61.8**	**65.44**	82.06	78.94	**94.84**	**85.84**	**66.4**	**68.00**	**81.98**	**82.91**	94.62	**86.16**	**67.4**	**70.14**	**84.85**
	Token Budget: 200×2048																	
	Llama-3.2-1B						**Llama-3.2-3B**						**Llama-3.1-8B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	45.79	46.32	32.70	49.6	54.61	59.13	58.62	82.03	51.40	68.4	73.72	73.87	65.41	85.90	50.70	75.4	82.08	75.92
Real-text	**48.01**	48.67	34.88	51.8	56.91	61.26	60.73	**84.23**	53.76	70.6	**75.77**	76.17	67.49	88.10	**52.92**	77.6	84.13	78.13
GRADMM	47.43	48.45	35.20	52.4	56.57	61.59	61.20	84.08	54.12	71.2	75.51	76.41	67.14	88.48	52.24	78.0	83.96	78.38
Ours	47.88	**48.90**	**36.12**	**53.4**	**57.17**	**62.49**	**61.96**	84.08	**55.16**	**72.0**	75.60	**77.23**	**67.63**	**89.46**	52.82	**79.0**	**84.73**	**79.28**
	Qwen3-8B						**Qwen3-14B**						**Qwen3-32B**					
Method	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA	MMLU	G8K	MATH	MBPP	ARC	CSQA
Base	74.90	91.89	83.98	59.6	64.33	81.98	78.85	93.86	84.76	64.6	66.98	81.00	82.11	94.54	85.26	65.8	69.28	84.03
Real-text	76.97	93.93	86.10	61.8	66.47	**84.11**	**80.89**	95.91	86.84	66.8	69.11	83.05	84.12	96.59	87.32	**68.0**	71.33	86.08
GRADMM	77.41	93.63	86.44	62.2	66.30	83.36	80.51	96.29	87.32	66.4	69.37	82.72	84.65	96.36	87.56	67.6	71.76	85.83
Ours	**78.09**	**94.16**	**87.40**	**63.0**	**67.24**	83.95	80.81	**97.12**	**88.08**	**67.2**	**70.14**	**83.62**	**85.26**	**97.19**	**88.28**	67.8	**72.44**	**86.65**

4.3 Robustness Across Random Seeds

The main experiment covers many model–task combinations and therefore reports single-run results. To assess run-to-run variation, we evaluate representative configurations over three independent runs on MMLU, GSM8K, and MBPP for Qwen3-8B and Qwen3-14B under the 200×2048 data budget. Table 2 reports the mean and standard deviation.

Table 2: **Robustness across random seeds.** Mean $\pm$ standard deviation over three independent runs under the 200×2048 data budget. Random uses Gaussian inputs matched to the mean and standard deviation of DASA's initialization embeddings.

	Qwen3-8B			**Qwen3-14B**		
Method	**MMLU ↑**	**GSM8K ↑**	**MBPP ↑**	**MMLU ↑**	**GSM8K ↑**	**MBPP ↑**
Random	71.8 ± 0.35	87.9 ± 0.48	55.2 ± 0.8	77.9 ± 0.31	90.1 ± 0.41	62.6 ± 0.6
Real-text	76.97 ± 0.13	$\mathbf{93.93 \pm 0.21}$	61.8 ± 0.6	80.89 ± 0.17	95.91 ± 0.22	$\mathbf{66.8 \pm 0.4}$
GRADMM	77.41 ± 0.15	93.63 ± 0.24	62.2 ± 0.6	80.51 ± 0.16	96.29 ± 0.22	66.4 ± 0.4
Ours	$\mathbf{78.09 \pm 0.16}$	93.78 ± 0.33	$\mathbf{63.0 \pm 0.6}$	$\mathbf{81.15 \pm 0.29}$	$\mathbf{97.12 \pm 0.23}$	66.6 ± 0.6

Random baseline. For each run, Random replaces the optimized synthetic inputs with a tensor whose entries are sampled independently from $\mathcal{N}(\mu_E, \sigma_E^2)$, where $\mu_E = \mathrm{mean}(Z_{\mathrm{init}})$ and $\sigma_E = \mathrm{std}(Z_{\mathrm{init}})$ are computed over all entries of the corresponding DASA initialization tensor. The random tensor

receives no synthetic-input optimization and is evaluated with the same data budget and downstream LoRA configuration.

DASA achieves a higher mean than both GRADMM and Random in all six settings, and the highest mean among all methods in four of them. On Qwen3-8B, DASA improves MMLU from 77.41 ± 0.15 with GRADMM to 78.09 ± 0.16, and MBPP from 62.2 ± 0.6 to 63.0 ± 0.6. On Qwen3-14B, its GSM8K result is 97.12 ± 0.23, compared with 96.29 ± 0.22 for GRADMM. Real-text retains the highest mean on Qwen3-8B GSM8K and Qwen3-14B MBPP.

The standard deviations of DASA are generally comparable to those of Real-text and GRADMM, although DASA has a higher standard deviation on Qwen3-14B MBPP. These results describe the consistency of the observed mean gains across the representative configurations. Three runs per setting are insufficient to establish statistical significance.

4.4 Task-Specialized Calibration Data

The main experiments deliberately use generic C4 data to separate update realization from direct access to task-specific examples. We next consider a complementary setting in which informative in-domain calibration data are available. For each benchmark, calibration examples are drawn from its corresponding training split: MMLU auxiliary train, GSM8K train, MATH train, MBPP train, ARC-Challenge train, and CommonsenseQA train. The LoRA configuration, optimization budget, and evaluation protocol remain unchanged, so the only difference from the main setting is the source of the calibration data.

Table 3 shows that DASA continues to outperform GRADMM in all 12 model–task comparisons across Qwen3-8B and Qwen3-14B. It achieves the strongest overall results in eight of the twelve settings, including MMLU, MATH, ARC-C, and CSQA, across both model scales. On Qwen3-14B, for example, DASA reaches 82.91 on MMLU, 89.18 on MATH, 71.59 on ARC-C, and 84.52 on CSQA, compared with 82.17, 88.38, 70.73, and 83.78 for GRADMM.

The improvement remains visible even though task-specialized calibration substantially strengthens the competing methods. This is an important complement to the C4 experiment: DASA is not effective only because generic calibration makes parameter-gradient matching difficult. Instead, its advantage persists when the calibration distribution is closely related to the target task and provides a substantially stronger adaptation signal.

Real-text remains slightly stronger on GSM8K and MBPP, where directly fine-tuning on task-specific natural-language examples is particularly competitive. Nevertheless, DASA consistently improves upon parameter-gradient matching across every evaluated configuration. Taken together with the generic-C4 results, these experiments indicate that activation-aligned synthesis is effective under substantially different calibration distributions rather than relying on a particular source of calibration text.

Table 3: **Results with task-specialized calibration data.** Calibration data are drawn from the corresponding training split for each benchmark: MMLU auxiliary train, GSM8K train, MATH train, MBPP train, ARC-Challenge train, and CommonsenseQA train. All adaptation methods use identical LoRA and optimization settings.

Specialized Data												
Method	**Qwen3-8B**						**Qwen3-14B**					
	MMLU	GSM8K	MATH	MBPP	ARC-C	CSQA	MMLU	GSM8K	MATH	MBPP	ARC-C	CSQA
Base	74.90	91.89	83.98	59.6	64.33	81.98	78.85	93.86	84.76	64.6	66.98	81.00
Real-text	78.02	**94.62**	87.18	**64.0**	67.49	85.01	81.82	**96.36**	88.00	**69.0**	70.39	84.03
GRADMM	78.41	94.39	87.54	62.6	67.83	84.77	82.17	96.13	88.38	68.6	70.73	83.78
Ours	**79.26**	94.54	**88.36**	62.8	**68.77**	**85.42**	**82.91**	96.29	**89.18**	68.8	**71.59**	**84.52**

4.5 Computational Efficiency

Matching parameter gradients across all Transformer layers requires repeated computation and comparison of high-dimensional gradients during synthetic-data construction. We therefore compare

DASA with our all-layer implementation of GRADMM in terms of generation time and peak GPU memory under the same three calibration budgets.

All efficiency measurements are collected on a single NVIDIA A800 GPU using FP16 precision. Both methods use all Transformer layers and optimize continuous input embeddings, which are used directly for downstream LoRA adaptation without vocabulary projection. For each calibration budget, they use the same number of devices, synthetic sequences, maximum sequence length, and synthesis steps. Peak memory is recorded over the complete generation procedure. These matched settings allow us to compare the computational costs of the two synthesis procedures.

As shown in Table 4, DASA substantially reduces generation time. At budgets of 50×2048, 100×2048, and 200×2048, GRADMM requires 28.3, 57.4, and 84.2 minutes, whereas DASA requires only 5.8, 11.6, and 23.3 minutes, respectively. This corresponds to approximately 3.6–$4.9\times$ faster synthetic-data generation. The reduction is consistent with DASA evaluating directional effects through activation gradients, while GRADMM matches parameter gradients across all Transformer layers.

Peak GPU memory is similar for the two methods, remaining between 17.3 and 17.9 GB across all settings. Thus, DASA's measured computational benefit is a substantially shorter generation time, with comparable peak memory.

Table 4: **Computational efficiency.** Generation time and peak GPU memory under three calibration budgets, measured on a single NVIDIA A800 GPU using FP16. GRADMM and DASA use matched sequence counts, sequence lengths, and synthetic-data optimization steps.

	Generation Time (min) ↓			Peak Memory (GB) ↓		
Method	$\mathbf{50 \times 2048}$	$\mathbf{100 \times 2048}$	$\mathbf{200 \times 2048}$	$\mathbf{50 \times 2048}$	$\mathbf{100 \times 2048}$	$\mathbf{200 \times 2048}$
GRADMM	28.3	57.4	84.2	17.3	17.7	17.9
DASA	**5.8**	**11.6**	**23.3**	**17.3**	**17.4**	**17.9**

4.6 Sensitivity Analysis

We examine whether DASA depends on a narrowly selected downstream adaptation configuration. We vary the LoRA rank, which controls the capacity of the parameter update, and the number of downstream LoRA optimization steps. The synthetic-data optimization schedule is held fixed. Figure 1 reports validation accuracy for adaptation on synthetic and real-text inputs under these configurations.

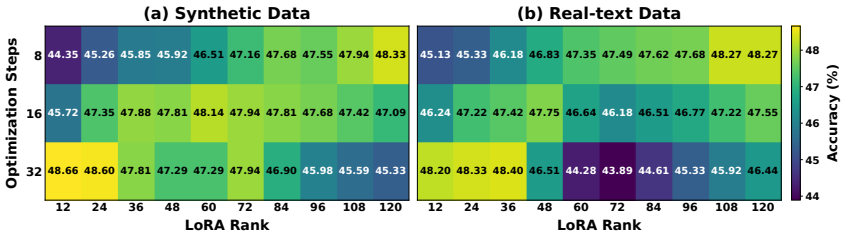


Figure 1: **Sensitivity to LoRA rank and downstream optimization steps.** Validation accuracy (%) under different LoRA ranks and numbers of downstream LoRA optimization steps for synthetic and real-text adaptation.

DASA performs competitively across a range of configurations rather than relying on a single choice of rank and training duration. At smaller LoRA ranks, additional downstream steps generally improve validation accuracy. Larger ranks reach strong performance with fewer steps, while continued adaptation can reduce accuracy. The real-text results show a similar dependence on rank and training duration, although the best configurations differ.

These results suggest that LoRA rank and downstream optimization steps are partly substitutable: a higher rank expands the space of possible updates, while more steps allow the model to use a fixed update space more fully. Several intermediate configurations achieve similar accuracy, indicating that DASA's performance does not depend on a narrowly tuned LoRA configuration.

4.7 Qualitative Analysis of Synthetic Data

Figure 2 compares a human-readable solution with a decoded visualization of an optimized DASA sample. The visualization projects the continuous synthetic embeddings to vocabulary tokens for inspection only. Downstream adaptation uses the original continuous embeddings.

The projected sample contains fragmented words, mixed scripts, and locally incoherent token combinations. This example illustrates that the synthetic inputs need not retain a fluent natural-language surface form. Because token projection loses information from the continuous embeddings, we use downstream task performance, rather than the readability of the displayed text, to assess their training utility.

The continuous synthetic inputs are optimized in the model's embedding space to induce useful update directions, and their training value is therefore determined by how they interact with the model rather than by whether they admit an interpretable token-level rendering. The visualization in Figure 2 should thus be viewed as evidence of a separation between human readability and model utility, rather than as a faithful representation of the information contained in the continuous synthetic inputs.

Human-readable solution

```
Question: Ralph watches TV for 4 hours a
day from Monday to Friday, and 6 hours a
day on Saturday and Sunday. How many hours
does Ralph spend watching TV in one week?

Answer:

From Monday to Friday, Ralph watches TV
for 4 * 5 = <<4*5=20>>20 hours.
On Saturday and Sunday, Ralph watches TV
for 2 * 6 = <<2*6=12>>12 hours.
In one week, Ralph watches TV for a total
of 20 + 12 = <<20+12=32>>32 hours.

#### 32
```

Highest-token projection

```
'\\ Frage samRal 元itablyfor theかない
hrs,_dayfrom Tuesday去١٢٨٨ expanded 澳bst
Entre-hour anday3 Saturdays酸 Tuesday
Initial Why viele minutes.Does  spending
ฉ่g  tv and the drive?'  Answer Tomas
ΔεAuraAssistantanas
.getExternal monday approvingTuesday-addons
Reducer antic tv and  migliorIRC
Path:');
<<" many[0 =16!?21_hours. On Saturdays is
soph
RAL Belg tlak the a لدي *, to8 =
Zoom ≠/ฅ๑-<\4='16 ';'24 hrs,To ONEенное;
zion SUB název hidden on musician as "15
подраз "ducted Shame "<< Falcon+.16)[-38:
pkulationHours and دیگر and なく
```

Figure 2: **Qualitative example of DASA synthetic data.** A human-readable solution is shown alongside a decoded visualization of an optimized continuous synthetic sample.

5 Conclusion

We investigated whether effective LLM fine-tuning requires human-readable training inputs. DASA constructs continuous synthetic input embeddings by optimizing the first-order effect of their induced direction on a reference loss, while keeping the pretrained model fixed during synthesis. Across six models, six benchmarks, and three data budgets, the resulting embeddings provide downstream LoRA adaptation that is often comparable to real-text fine-tuning and outperforms GRADMM in most evaluated model–task settings. Under matched synthesis workloads, DASA also reduces generation time by 3.6–$4.9\times$ with comparable peak GPU memory. These findings show that, in the evaluated adaptation settings, useful training signals can be carried by model-conditioned continuous inputs without preserving readable language. The directional objective provides a local criterion rather than a guarantee for finite-step fine-tuning; understanding its relationship to downstream updates and transfer across models remains future work.

AI USE STATEMENT

We used GPT-6 solely for language polishing. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final manuscript.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. URL https://arxiv.org/abs/2108.07732.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2203.11932.

Daning Cheng, Zeyu Liu, Jun Sun, Fen Xia, Boyang Zhang, Dongping Liu, and Yunquan Zhang. A qualitative test-risk mechanism for scaling behavior in normalized residual networks. arXiv preprint arXiv:2605.08297, 2026. URL https://arxiv.org/abs/2605.08297.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Keyu Duan, Yiran Zhao, Zhili Feng, Jinjie Ni, Tianyu Pang, Qian Liu, Tianle Cai, Longxu Dou, Kenji Kawaguchi, Anirudh Goyal, J. Zico Kolter, and Michael Qizhe Shieh. Unnatural languages are not bugs but features for LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14778–14792. PMLR, 2025. URL https://proceedings.mlr.press/v267/duan25c.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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. In *International Conference on Learning Representations*, 2022b. URL https://arxiv.org/abs/2106.09685.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Aru Maekawa, Satoshi Kosugi, Kotaro Funakoshi, and Manabu Okumura. DiLM: Distilling dataset into language model for text-level dataset distillation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3138–3153. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.199. URL https://aclanthology.org/2024.findings-naacl.199/.

Meta. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024. Accessed 2026-09-20.

Dang Nguyen, Zeman Li, MohammadHossein Bateni, Vahab Mirrokni, Meisam Razaviyayn, and Baharan Mirzasoleiman. Synthetic text generation for training large language models via gradient matching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/nguyen25c.html.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346/.

Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021. URL https://ieeexplore.ieee.org/document/9533769.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12186–12195, 2022. doi: 10.1109/CVPR52688.2022.01188. URL https://arxiv.org/abs/2203.01531.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. arXiv preprint arXiv:1811.10959, 2018. URL https://arxiv.org/abs/1811.10959.

Rong Yao, Hailin Hu, Yifei Fu, Hanting Chen, Wenyi Fang, Fanyi Du, Kai Han, and Yunhe Wang. Transferable text data distillation by trajectory matching. arXiv preprint arXiv:2504.09818, 2025. URL https://arxiv.org/abs/2504.09818.

Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://github.com/NVlabs/DeepInversion.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12674–12685. PMLR, 2021. URL https://proceedings.mlr.press/v139/zhao21a.html.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. URL https://arxiv.org/abs/2110.04181.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl.

A Basic Setup and Notation

Basic setup. Let $\mathcal{D}_{\text{ref}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{ref}}}$ be the finite reference dataset. We use $\mathbb{E}_{\mathcal{D}_{\text{ref}}}[\cdot]$ to denote averaging under its empirical distribution, i.e., $\mathbb{E}_{\mathcal{D}_{\text{ref}}}[g] = N_{\text{ref}}^{-1} \sum_{i=1}^{N_{\text{ref}}} g(x_i, y_i)$. Write $z_i = f_{\text{bot}}(x_i)$ and $\zeta_i = \nabla_{z_i} \ell(f_{\text{top}}(z_i), y_i)$.

Algorithm 1 optimizes continuous synthetic inputs Z^s. Each Z^s induces a parameter-space tangent direction $v(Z^s)$, whose first-order effect on the reference activations is $u_{v(Z^s)}(z_i)$. The synthesis objective is the first-order variation of the empirical reference loss:

$$\min_{Z^s} D_{\mathcal{D}_{\text{ref}}}(v(Z^s)), \qquad D_{\mathcal{D}_{\text{ref}}}(v(Z^s)) = \frac{1}{N_{\text{ref}}} \sum_{i=1}^{N_{\text{ref}}} \zeta_i^\top u_{v(Z^s)}(z_i).$$

A negative value indicates a locally improving direction for a sufficiently small positive step under the stated differentiability and realizability conditions; it does not guarantee the outcome of finite-step fine-tuning.

For any distribution Q, let $\mu(Q) = \mathbb{E}_Q[\zeta]$ and $\Sigma_Q = \text{Cov}_Q(\zeta)$. These statistics are used to decompose and bound $D_Q(v)$ below; they are not optimization targets for Z^s.

Assumptions. We analyze local adaptation around the fixed pretrained model f_{θ_0}. We assume differentiability of the relevant mappings and finite second moments of the activations and activation gradients, ensuring that the expectations, covariance matrices, and first-order quantities below are well defined. Additional assumptions required by individual results are stated locally.

Notation

Symbol	Meaning	Symbol	Meaning
$f_{\theta_0} = f_{\text{top}} \circ f_{\text{bot}}$	Fixed model decomposed at l^*.	l^*, N	Target layer and hidden dimension.
$z_i = f_{\text{bot}}(x_i)$	Activation of reference example i at l^*.	ζ_i	Activation gradient of reference example i.
$\mathcal{D}_{\text{ref}}$	Finite reference dataset.	N_{ref}	Number of reference examples.
$\mathbb{E}_{\mathcal{D}_{\text{ref}}}$	Average over the reference dataset.	Q	A distribution, including an empirical distribution.
Z^s	Continuous synthetic inputs optimized by DASA.	S_{syn}	Stored synthetic inputs used for fine-tuning.
$v(Z^s)$	Parameter-space tangent direction induced by Z^s.	v	A generic parameter-space tangent direction.
$u_v(z)$	First-order activation change induced by v at z.	$D_Q(v)$	First-order risk variation under Q.
$\mu(Q)$	Mean activation gradient under Q.	Σ_Q	Covariance of activation gradients under Q.
$\bar{z}_Q$	Mean activation under Q.	$\sigma_z^2(Q)$	Activation dispersion under Q.
$\bar{u}_{v,Q}$	Mean first-order activation change under Q.	$c_Q(v)$	Covariance residual in the decomposition of $D_Q(v)$.
h_θ	Residual adaptation branch.	A, B, r, α	LoRA factors, rank, and scaling factor.
$\|\cdot\|_\sigma$	Spectral norm.		

Synthetic-induced tangent direction. Let h_θ be an auxiliary residual branch at the target layer, parameterized around a zero-output initialization $h_0 = 0$. For the current synthetic inputs Z^s and their retained prediction targets y^s, let z^s denote the resulting target-layer activations and let $\zeta^s = \nabla_{z^s} \ell(f_{\text{top}}(z^s), y^s)$. Write $\mathbb{E}_{\text{syn}}$ for the empirical average using the same prediction-target mask and normalization as the synthetic loss. We construct the parameter-space tangent direction and its effect on a reference activation by

$$v(Z^s) = -\mathbb{E}_{\text{syn}}\left[\left(\frac{\partial h_\theta(z^s)}{\partial \theta}\bigg|_{\theta=0}\right)^\top \zeta^s\right], \qquad u_{v(Z^s)}(z) = \frac{\partial h_\theta(z)}{\partial \theta}\bigg|_{\theta=0} v(Z^s).$$

Thus $v(Z^s)$ is constructed from the synthetic activation gradients in the tangent space of the auxiliary residual branch. The pretrained model and the auxiliary branch remain fixed during synthesis; only Z^s is updated. LoRA fine-tuning is performed separately after the synthetic data have been generated.

B Theoretical Background

Auxiliary residual branch. Cheng et al. (2026) study the expansion of a fixed reference model $f^*_{\text{old}} = f_{\text{top}} \circ f_{\text{bot}}$. At a selected insertion point, let $z = f_{\text{bot}}(x)$ and add an auxiliary residual branch h_θ:

$$f_\theta(x) = f_{\text{top}}\big(z + h_\theta(z)\big). \tag{5}$$

The branch has a zero-output initialization, $h_0(z) = 0$, so the expanded model initially computes the same function as f^*_{old}. For a parameter-space direction v, let $u_v(z)$ denote the first-order representation change induced by the branch, and define the activation gradient

$$\zeta = \nabla_z \ell\big(f_{\text{top}}(z), y\big). \tag{6}$$

The inner product $\zeta^\top u_v(z)$ is the first-order change in the loss of an individual example. Here v is a direction in the branch's parameter space, whereas $u_v(z)$ is its effect in representation space. The representation change must be realizable by the branch.

Population risk and the population jumpboard. Let $\mathcal{Q}$ be a data distribution. The *population risk* of a model f is its expected loss under $\mathcal{Q}$:

$$R_{\mathcal{Q}}(f) = \mathbb{E}_{(x,y)\sim\mathcal{Q}}\big[\ell(f(x), y)\big]. \tag{7}$$

For the expanded model along the path ηv, write $f_{\eta v}(x) = f_{\text{top}}\big(z + h_{\eta v}(z)\big)$. Under the differentiability and expectation-interchange conditions of Cheng et al. (2026), the initial directional derivative of its population risk is

$$\begin{aligned} D_{\mathcal{Q}}(v) &= \frac{\mathrm{d}}{\mathrm{d}\eta} R_{\mathcal{Q}}(f_{\eta v})\bigg|_{\eta=0} \\ &= \mathbb{E}_{\mathcal{Q}}\big[\zeta^\top u_v(z)\big]. \end{aligned} \tag{8}$$

Condition 1 of Cheng et al. (2026) requires an admissible insertion point and a realizable direction v_{pop} such that $D_{\mathcal{Q}}(v_{\text{pop}}) < 0$. By differentiability, there is a sufficiently small positive η for which

$$R_{\mathcal{Q}}(f_{\eta v_{\text{pop}}}) < R_{\mathcal{Q}}(f^*_{\text{old}}). \tag{9}$$

Their Theorem 4 therefore establishes the existence of an expanded model $\widetilde{f}_{\text{pop}}$ with lower population risk. We refer to this comparison model as the *population jumpboard*. It witnesses an improving point in the expanded model class; the existence result does not imply that a particular optimization algorithm will find it.

Empirical risk and the empirical jumpboard. For a finite sample $S = \{(x_i, y_i)\}_{i=1}^m$, its *empirical risk* is

$$L_S(f) = \frac{1}{m}\sum_{i=1}^{m} \ell(f(x_i), y_i). \tag{10}$$

Thus $L_{S_{\text{train}}}$ and $L_{S_{\text{test}}}$ denote the losses averaged over the training and test samples, respectively; neither is identical to the population risk $R_{\mathcal{Q}}$. A direction selected using S_{train}, followed by a small positive step, produces an *empirical jumpboard* $\widetilde{f}_S$. Unlike $\widetilde{f}_{\text{pop}}$, this model depends on the training sample. It is also distinct from the final trained expanded model f_{new}.

To connect an empirical jumpboard to test loss, Cheng et al. (2026) consider a setting in which the residual branch can realize a constant representation perturbation. Let μ_M and g_K be the mean activation gradients of independent training and test samples of sizes M and K. If the training-selected direction realizes $u_v(z) = -\mu_M$, then

$$\frac{\mathrm{d}}{\mathrm{d}\eta} L_{S_{\text{train}}}(f_{\eta v})\bigg|_{\eta=0} = -\|\mu_M\|^2, \qquad \frac{\mathrm{d}}{\mathrm{d}\eta} L_{S_{\text{test}}}(f_{\eta v})\bigg|_{\eta=0} = -\mu_M^\top g_K. \tag{11}$$

Consequently, $\mu_M^\top g_K > 0$ implies a local decrease in test loss for a sufficiently small positive step. This implication requires the constant perturbation to be realizable by the branch and does not certify an arbitrary finite step.

Probability of gradient alignment. The preceding local test-loss argument depends on the sign of $\mu_M^\top g_K$. Theorem 5 of Cheng et al. (2026) bounds the probability of nonpositive alignment. Using the notation of Appendix A, let

$$\mu(\mathcal{Q}) = \mathbb{E}_{\mathcal{Q}}[\zeta], \qquad \Sigma_{\mathcal{Q}} = \mathrm{Cov}_{\mathcal{Q}}(\zeta). \tag{12}$$

Suppose that the training and test samples are independent draws from $\mathcal{Q}$, $\mu(\mathcal{Q}) \neq 0$, and $\lambda_{\max}(\Sigma_{\mathcal{Q}}) \leq C_\Sigma \tau^2$. Then

$$\begin{aligned}\Pr(\mu_M^\top g_K \leq 0) \leq \frac{4C_\Sigma\tau^2}{M\|\mu(\mathcal{Q})\|^2} + \frac{4C_\Sigma\tau^2}{K\|\mu(\mathcal{Q})\|^2} \\ + \frac{4C_\Sigma\tau^2\,\mathrm{tr}(\Sigma_{\mathcal{Q}})}{MK\|\mu(\mathcal{Q})\|^4}.\end{aligned} \tag{13}$$

The proof separates two possible failures: the training mean may deviate substantially from the population mean, or the independent test mean may fluctuate against the training direction. The covariance bound controls the probabilities of these events. Theorem 5 concerns the sign of gradient alignment; it does not by itself lower-bound the magnitude of a test-loss improvement.

From a jumpboard to the final expanded model. The empirical jumpboard is a comparison model for f_{new}. Define its realized test-loss reduction relative to the reference model as

$$\Delta_R^{\mathrm{test}} = L_{S_{\mathrm{test}}}(f^*_{\mathrm{old}}) - L_{S_{\mathrm{test}}}(\widetilde{f}_S). \tag{14}$$

Under the model-selection and uniform-generalization conditions of Cheng et al. (2026), their Theorem 3 combines this reduction with an additional training-loss improvement Δ_{ERM} and generalization terms ϵ_M and ϵ_K:

$$\begin{aligned}L_{S_{\mathrm{test}}}(f_{\mathrm{new}}) \leq L_{S_{\mathrm{test}}}(f^*_{\mathrm{old}}) - \Delta_R^{\mathrm{test}} - \Delta_{\mathrm{ERM}} \\ + 2(\epsilon_M + \epsilon_K).\end{aligned} \tag{15}$$

The comparison first relates the final model's training loss to the jumpboard's training loss, then applies the uniform train–test loss bound to both models. A strict improvement of the final model follows from this bound only when the realized gains exceed the generalization terms. The population-risk existence result, the local empirical jumpboard argument, and the final-model test-loss comparison therefore establish different parts of the prior framework.

C Covariance Control under LoRA

We specialize the first-order analysis to LoRA. Consider the residual adaptation branch

$$h_{A,B}(z) = \frac{\alpha}{r} BAz,$$

where A and B are the LoRA factors, r is the rank, and α is the scaling factor. We consider the standard initialization $B = 0$.

Let $v = (V_A, V_B)$ be a tangent direction. Along the parameter path

$$A(\eta) = A + \eta V_A, \qquad B(\eta) = \eta V_B,$$

the residual branch becomes

$$\begin{aligned}h_{A(\eta),B(\eta)}(z) &= \frac{\alpha}{r}(\eta V_B)(A + \eta V_A)z \\ &= \eta\frac{\alpha}{r} V_B Az + \eta^2 \frac{\alpha}{r} V_B V_A z.\end{aligned}$$

Taking the derivative with respect to η at $\eta = 0$ gives

$$u_v(z) = \frac{d}{d\eta} h_{A(\eta),B(\eta)}(z)\bigg|_{\eta=0} = \frac{\alpha}{r} V_B Az.$$

Thus, the perturbation V_A has no first-order contribution under the initialization $B = 0$.

We next decompose the first-order variation. By definition,

$$D_Q(v) = \mathbb{E}_Q[\zeta^\top u_v].$$

Write

$$\zeta = \mu(Q) + (\zeta - \mu(Q))$$

and

$$u_v = \bar{u}_{v,Q} + (u_v - \bar{u}_{v,Q}).$$

Substituting both decompositions gives

$$\begin{aligned} D_Q(v) &= \mathbb{E}_Q\Big[\big(\mu(Q) + \zeta - \mu(Q)\big)^\top \big(\bar{u}_{v,Q} + u_v - \bar{u}_{v,Q}\big)\Big] \\ &= \mu(Q)^\top \bar{u}_{v,Q} + \mu(Q)^\top \mathbb{E}_Q[u_v - \bar{u}_{v,Q}] \\ &\quad + \mathbb{E}_Q[\zeta - \mu(Q)]^\top \bar{u}_{v,Q} \\ &\quad + \mathbb{E}_Q\left[(\zeta - \mu(Q))^\top (u_v - \bar{u}_{v,Q})\right]. \end{aligned}$$

Since

$$\mathbb{E}_Q[\zeta - \mu(Q)] = 0$$

and

$$\mathbb{E}_Q[u_v - \bar{u}_{v,Q}] = 0,$$

the two cross terms vanish. Therefore,

$$D_Q(v) = \mu(Q)^\top \bar{u}_{v,Q} + c_Q(v),$$

where

$$c_Q(v) = \mathbb{E}_Q\left[(\zeta - \mu(Q))^\top (u_v - \bar{u}_{v,Q})\right].$$

Theorem 1 (Covariance Control)**.** *Under the assumptions in Appendix A,*

$$\boxed{|c_Q(v)| \le \sqrt{\mathrm{tr}(\Sigma_Q)}\frac{\alpha}{r}\|V_B A\|_\sigma \sigma_z(Q).}$$

Proof. By definition,

$$c_Q(v) = \mathbb{E}_Q\left[(\zeta - \mu(Q))^\top (u_v - \bar{u}_{v,Q})\right].$$

Applying Cauchy–Schwarz twice gives

$$\begin{aligned} |c_Q(v)| &\le \mathbb{E}_Q\left[\|\zeta - \mu(Q)\|_2 \|u_v - \bar{u}_{v,Q}\|_2\right] \\ &\le \sqrt{\mathbb{E}_Q\|\zeta - \mu(Q)\|_2^2}\sqrt{\mathbb{E}_Q\|u_v - \bar{u}_{v,Q}\|_2^2}. \end{aligned}$$

For the first factor, since

$$\Sigma_Q = \mathbb{E}_Q\left[(\zeta - \mu(Q))(\zeta - \mu(Q))^\top\right],$$

we have

$$\mathbb{E}_Q\|\zeta - \mu(Q)\|_2^2 = \mathrm{tr}(\Sigma_Q).$$

For the second factor, using

$$u_v(z) = \frac{\alpha}{r} V_B A z$$

and

$$\bar{u}_{v,Q} = \frac{\alpha}{r} V_B A \bar{z}_Q,$$

we obtain

$$u_v - \bar{u}_{v,Q} = \frac{\alpha}{r} V_B A (z - \bar{z}_Q).$$

Hence,

$$\begin{aligned} \mathbb{E}_Q\|u_v - \bar{u}_{v,Q}\|_2^2 &= \frac{\alpha^2}{r^2}\mathbb{E}_Q\|V_B A(z - \bar{z}_Q)\|_2^2 \\ &\le \frac{\alpha^2}{r^2}\|V_B A\|_\sigma^2 \mathbb{E}_Q\|z - \bar{z}_Q\|_2^2 \\ &= \frac{\alpha^2}{r^2}\|V_B A\|_\sigma^2 \sigma_z^2(Q). \end{aligned}$$

Substituting the two bounds above yields

$$|c_Q(v)| \le \sqrt{\mathrm{tr}(\Sigma_Q)}\frac{\alpha}{r}\|V_B A\|_\sigma \sigma_z(Q).$$

□

Implication. From the decomposition above,

$$D_Q(v) = \mu(Q)^\top \bar{u}_{v,Q} + c_Q(v).$$

Since $c_Q(v) \le |c_Q(v)|$, Theorem 1 implies

$$D_Q(v) \le \mu(Q)^\top \bar{u}_{v,Q} + \sqrt{\mathrm{tr}(\Sigma_Q)}\frac{\alpha}{r}\|V_B A\|_\sigma \sigma_z(Q).$$

Therefore, if

$$-\mu(Q)^\top \bar{u}_{v,Q} > \sqrt{\mathrm{tr}(\Sigma_Q)}\frac{\alpha}{r}\|V_B A\|_\sigma \sigma_z(Q),$$

then the right-hand side is strictly negative, and hence

$$D_Q(v) < 0.$$

For a fixed reference distribution Q, the activation statistics $\mu(Q)$, Σ_Q, and $\sigma_z(Q)$ are fixed. In contrast, the direction $v = v(Z^s)$ is induced by the synthetic inputs and can therefore be shaped during synthesis. Optimizing Z^s can increase the favorable mean component $-\mu(Q)^\top \bar{u}_{v,Q}$ while controlling the geometry of the induced LoRA direction through $\|V_B A\|_\sigma$. The bound therefore provides a mechanism by which synthetic-data optimization can favor directions whose mean improvement dominates the covariance residual, whereas a fixed real-text corpus does not directly optimize this trade-off.

D Training Details

Downstream LoRA adaptation. We freeze all pretrained backbone parameters and perform downstream adaptation using LoRA with rank $r = 32$, scaling factor $\alpha = 32$, and dropout 0.05. LoRA modules are applied to the query, key, value, output, gate, up, and down projection layers. We optimize the LoRA parameters for 32 optimizer steps using AdamW with a learning rate of 2×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, and gradient clipping at 1.0. We use a cosine learning-rate schedule with a 3% warmup ratio, BF16 precision, a maximum sequence length of 2048, a per-device batch size of 1, and gradient accumulation over 8 steps. Unless otherwise specified, all experiments use the same downstream adaptation configuration across models, tasks, and methods.

Synthetic-data optimization. DASA initializes continuous synthetic inputs from the input embeddings of sampled reference sequences and retains their next-token prediction targets. The pretrained model and prediction targets remain fixed during synthesis; only the synthetic inputs are optimized. We update these inputs for $T_{\mathrm{syn}} = 60$ steps using SGD with learning rate $\eta_{\mathrm{syn}} = 0.1$, zero momentum, zero weight decay, and gradient clipping at 1.0. The resulting synthetic tensors are stored in BF16 and used directly as input embeddings during downstream LoRA adaptation. DASA and GRADMM use identical sequence counts, sequence lengths, synthesis steps, and downstream adaptation configurations.