Mind the Gap: The Divergent Rebound Dynamics of Diffusion and Autoregressive Model

Qiwen Jin[1] **Zhengyu Zhou**[1] **Weiwei Liu**[2*] **Xiuwen Gong**[1*]
[1]School of Computer Science, Wuhan University, China
[2]Institute of Big Data, Fudan University, China
{jinqiwen12, liuweiwei863, gongxiuwen}@gmail.com

Abstract

Post-training alignment can be fragile: in autoregressive models (ARMs), subsequent fine-tuning on counter-aligned or distribution-shifted data can erode previously aligned behavior. While this rebound phenomenon has been studied mainly in ARMs, how such fine-tuning triggers a similar rebound in diffusion language models (DLMs) remains poorly understood. We conduct extensive fine-tuning experiments across diverse datasets, perturbation sizes, alignment algorithms, model scales, and inference hyperparameters, and find that DLMs consistently exhibit a slower rebound pattern than ARMs do. Furthermore, while ARMs fine-tuned on more positive data suffer a steeper degradation under reverse updates, DLMs exhibit ordering stability: models fine-tuned on larger positive alignment sets retain higher performance as negative data increases. We propose a compression perspective to account for this behavior. Unlike ARMs, whose compression naturally follows a fixed left-to-right order, DLMs generate text through iterative masked denoising. We formalize this distinction through a template-wise compression theory for DLMs. The resulting elasticity theory explains why counter-aligned updates lead to much slower alignment erosion in DLMs. Our experiment findings indicate that rebound is shaped not only by post-training data and objectives, but also by the underlying generation mechanism.

1 Introduction

Diffusion language models (DLMs) have shown increasingly competitive performance, making them a plausible alternative generation paradigm to autoregressive language models (ARMs) (Li et al., 2025; Sahoo et al., 2024; Lou et al., 2023). As both paradigms are increasingly adapted through supervised fine-tuning (SFT) (Ouyang et al., 2022), reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019; Schulman et al., 2017; Guo et al., 2025), and preference-optimization variants (Bai et al., 2022; Rafailov et al., 2023), a basic reliability question arises: whether the alignment-induced behavior of DLMs and ARMs remains stable under subsequent model updates.

In ARMs, post-training alignment has been shown to be fragile. Highly safety-aligned models may recover unsafe behavior after limited counter-aligned fine-tuning (Yang et al., 2023; Dong et al., 2024; Zou et al., 2023), and even fine-tuning on seemingly benign data can weaken safety mechanisms (Qi et al., 2024). Prior works (Ji et al., 2025; Kirkpatrick et al., 2017; Chen et al., 2025) have been largely developed in the ARM setting, where prediction and compression follow a fixed left-to-right order.

DLMs introduce a structurally different setting for studying post-alignment rebound. Instead of generating text left-to-right through next-token prediction (Brown et al., 2020; Zhang et al., 2024; Touvron et al., 2023), DLMs frame text generation as an iterative masked denoising process (Nie

*Corresponding authors

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

et al., 2025; Cheng et al., 2025). As a result, counter-aligned perturbations do not act through a single prefix-based path. Instead, a DLM may see many partially masked versions of the same sample. This leads to our central question: is post-alignment rebound a generation-agnostic consequence of further fine-tuning, or is it mediated by the masked reconstruction mechanism of DLMs? We study this question under a unified post-training protocol and find that DLMs consistently exhibit a slower rebound pattern than ARMs do. Our key contributions are summarized as follows:

- **Phenomenon.** We identify a generation-dependent difference in post-alignment rebound. Under unified counter-alignment settings, ARMs exhibit rapid degradation of aligned behavior, whereas DLMs show slower degradation. Moreover, we surprisingly observe a divergent trend: increasing positive samples accelerates performance erosion in ARMs, whereas it decelerates this degradation in DLMs. These results suggest that rebound dynamics are shaped more by the underlying generation mechanism than by post-training data and objectives alone.
- **Mechanism.** We develop a compression theory for DLMs. The theory models DLM compression through the masked reconstruction problems created by different response-mask patterns, and derives an elasticity result showing how perturbation data changes the compression of pretraining and alignment data. This provides a formal explanation for why DLMs degrade more slowly under counter-aligned updates.
- **Validation.** We substantiate the DLM–ARM rebound gap through extensive fine-tuning experiments. Across diverse datasets (spanning sentiment and safety tasks), perturbation sizes, model scales, and alignment algorithms (e.g., SFT and preference-optimization methods such as DPO and KTO), DLMs consistently exhibit a slower rebound pattern than their ARM counterparts.

2 A Compression View of DLMs

2.1 Preliminaries

Compression and prediction. Large language models (LLMs) are generative models that seek to approximate the clean data distribution p_0 by a parameterized model distribution p_θ. Let $x_0 = (x_0^1, \ldots, x_0^L) \in \mathcal{V}^L$ denote a clean sequence over vocabulary $\mathcal{V}$. Training by maximum likelihood is equivalent to minimizing the discrepancy between the two distributions:

$$\max_\theta \ \mathbb{E}_{x_0 \sim p_0}\big[\log p_\theta(x_0)\big] \qquad \Longleftrightarrow \qquad \min_\theta \ \mathrm{KL}(p_0(x_0) \,\|\, p_\theta(x_0))\,,$$

where KL denotes the Kullback-Leibler divergence. Moreover, under standard lossless coding schemes, the expected code length assigned by p_θ to data sampled from p_0 takes the form

$$\mathcal{L}(p_\theta) = \mathbb{E}_{x_0 \sim p_0}\big[-\log p_\theta(x_0)\big],$$

where the base of the logarithm determines the unit of the code length (e.g., bits for base 2). Hence, better prediction is equivalent to shorter expected code length, and minimizing negative log-likelihood can be interpreted as improving compression performance (Delétang et al., 2024; Huang et al., 2024). This connection motivates our use of compression as the basic analytical lens.

Autoregressive generation. An ARM factorizes sequence likelihood from left-to-right:

$$p_\theta^{\mathrm{AR}}(x_0) = \prod_{j=1}^{L} p_\theta(x_0^j \mid x_0^{<j}), \qquad -\log p_\theta^{\mathrm{AR}}(x_0) = \sum_{j=1}^{L} -\log p_\theta(x_0^j \mid x_0^{<j}),$$

where $x_0^{<j} = (x_0^1, \ldots, x_0^{j-1})$ denotes the prefix before position j. Prediction in ARMs follows a fixed left-to-right order, repeatedly generating the next token x_0^j conditioned on the previously generated prefix $x_0^{<j}$.

Masked diffusion generation. A masked diffusion language model (DLM) does not condition on a unique prefix state. Instead, it reconstructs masked tokens from partially observed sequences obtained via stochastic corruption. Given a masking ratio $t \in (0, 1)$, each token is independently

replaced by a special [MASK] symbol with probability t. This process induces a masked index set $\mathcal{M} \subseteq [L]$ and a corrupted sequence $\tilde{x}_0$, where $\tilde{x}_0^i = [\text{MASK}]$ for $i \in \mathcal{M}$ and $\tilde{x}_0^i = x_0^i$ otherwise.

Conditioned on the corrupted sequence $\tilde{x}_0$, a DLM predicts the original tokens at the masked positions:

$$p_\theta^{\text{DLM}}(x_0^{\mathcal{M}} \mid \tilde{x}_0) = \prod_{i \in \mathcal{M}} p_\theta(x_0^i \mid \tilde{x}_0), \qquad -\log p_\theta^{\text{DLM}}(x_0^{\mathcal{M}} \mid \tilde{x}_0) = \sum_{i \in \mathcal{M}} -\log p_\theta(x_0^i \mid \tilde{x}_0).$$

Derived from the evidence lower bound (ELBO), the corresponding masked reconstruction objective is formulated as a re-weighted cross-entropy loss:

$$\mathcal{L}_{\text{DLM}}(\theta) = \mathbb{E}_{t,\, x_0 \sim p_0,\, \mathcal{M}} \left[\frac{1}{t} \sum_{i \in \mathcal{M}} -\log p_\theta(x_0^i \mid \tilde{x}_0) \right].$$

where the expectation is taken over the data sequences $x_0 \sim p_0$, the masking ratio t, and the randomly induced mask index set $\mathcal{M}$. Consequently, the fundamental predictive mechanism in a DLM aims to model the conditional distribution $p_\theta(x_0^i \mid \tilde{x}_0)$.

2.2 Time-Free Compression Code Length

Following the compression view of language modeling (Delétang et al., 2024), we treat a learned DLM as a compressor. In ARMs, the compressor assigns code lengths through a left-to-right next-token factorization (Ji et al., 2025). In DLMs, the natural compression units are masked-token prediction instances. We therefore define the DLM code length using the time-free masked prediction form motivated by I-MDCE (Jeon et al., 2025).

Let I_1 denote the response-token index set and I_2 denote the prompt-token index set, with $I_1 \cap I_2 = \varnothing$. Let $p_\theta(a \mid \tilde{x}, i)$ denote the probability assigned by the DLM to token a at position i given the partially observed input $\tilde{x}$. We define the predictor-induced code length as:

$$\ell_\theta(x_0; I_1, I_2) = H_{|I_1|} \mathbb{E}_{J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} -\log p_\theta \left(x_0^i \mid \tilde{x}_0^{J \cup I_2}, i \right) \right],$$

where $J \subsetneq I_1$ is a randomly revealed subset of response-token indices, $i \in I_1 \setminus J$ is a masked response-token index, and $\tilde{x}_0^{J \cup I_2}$ denotes the sequence obtained from x_0 by masking all tokens whose indices are not in $J \cup I_2$. Here J is sampled from the time-free distribution $p(J) = B(|I_1| - |J|, |J| + 1)/H_{|I_1|}$, where $H_{|I_1|}$ is the $|I_1|$-th harmonic number and $B(\cdot, \cdot)$ denoting the Beta function. This code length expresses DLM compression as an average over single-token masked prediction costs. We use ℓ_θ as the predictor-induced code length in our compression-based analysis of DLM elasticity.

2.3 Template-Wise Compression

Since alignment and subsequent model updates often use data from multiple sources with distinct distributions, we study DLM compression in a joint dataset setting. In DLMs, each prompt–response example can induce many masked-token prediction instances through different response-mask patterns. We group these instances into templates to track how each dataset contributes to different types of masked-token prediction.

Definition 2.1 (DLM Masked-Input Prediction Template)**.** Using the notation from Section 2.2, a masked prediction instance is specified by:

$$(J, i, \tilde{x}_0^{J \cup I_2}).$$

A DLM masked-input prediction template U is a coarse category of such single-token prediction instances:

$$U = \Phi(J, i, \tilde{x}_0^{J \cup I_2}).$$

Here Φ is a template assignment map that assigns each masked prediction instance $(J, i, x_0^{J \cup I_2})$ to a template U. Thus, U denotes the type of masked prompt–partial-response context used to predict one masked response token, rather than an individual sequence, mask state, or diffusion time step. For brevity, we refer to such a DLM masked-input prediction template simply as a template U.

For example, with prompt `Review: The movie was` and response `surprisingly good and heartwarming`, I_2 indexes the prompt tokens and I_1 indexes the response tokens. Predicting `good` from the partial response `surprisingly [MASK] and heartwarming` may belong to $U_{\text{pos-adj}}$, while predicting `surprisingly` from `[MASK] good and heartwarming` may belong to $U_{\text{pos-int}}$. Thus, one clean sequence x_0 can induce multiple templates through different response-mask patterns.

Since training and alignment often involve multiple datasets with distinct distributions (Zou et al., 2026), we consider a joint compression view over a family of pairwise disjoint datasets $D_1, \ldots, D_N$, and write $D = \bigcup_{d=1}^{N} D_d$ for their joint dataset. Throughout this section, $x_0 \sim D_d$ denotes uniform sampling of a sequence from dataset D_d.

Definition 2.2 (Template Exposure Measure)**.** For a dataset D_d, the exposure of template U is defined as:

$$\omega_d(U) = \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{\Phi(J, i, \tilde{x}_0^{J \cup I_2}) = U\} \right].$$

Here $p(J)$ is the time-free response-mask distribution defined above. Thus, $\omega_d(U)$ measures how often samples from D_d produce masked-token prediction instances assigned to template U.

Definition 2.3 (Template-Conditional Token Distribution)**.** For a dataset D_d and a template U with $\omega_d(U) > 0$, define:

$$q_d(a \mid U) = \frac{\mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{\Phi(J, i, \tilde{x}_0^{J \cup I_2}) = U\} \mathbf{1}\{x_0^i = a\} \right]}{\omega_d(U)}, \qquad a \in \mathcal{V}.$$

For each template U, define the dataset-specific template mass as $M_d(U) = |D_d| \omega_d(U)$, and the total template mass as $M(U) = \sum_{d=1}^{N} M_d(U)$.

Definition 2.4 (Joint Template Token Distribution)**.** For any template U with $M(U) > 0$, define:

$$q_D(a \mid U) = \frac{\sum_{d=1}^{N} M_d(U) q_d(a \mid U)}{M(U)}, \qquad a \in \mathcal{V}.$$

Here $q_D(\cdot \mid U)$ is the target-token distribution obtained by pooling all datasets at template U, with each dataset weighted by its dataset-specific template mass $M_d(U)$.

For dataset D_d, we define the template-based compression rate as:

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} -\log p_\theta(x_0^i \mid U) \right],$$

This quantity averages the masked-token prediction loss over examples from D_d. It can be rewritten by grouping all masked-token predictions according to their template U.

Proposition 2.5 (Template-Wise Form)**.** *For dataset D_d, the template-based compression rate can be written as:*

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \sum_{U} \omega_d(U)\, \mathrm{CE}\big(q_d(\cdot \mid U),\, p_\theta(\cdot \mid U)\big),$$

where $\mathrm{CE}(q, p) = -\sum_{a \in \mathcal{V}} q(a) \log p(a)$.

This decomposition has a direct interpretation: each template contributes to the overall compression rate according to how often it appears in D_d and how well p_θ predicts the target tokens under that template. Thus, it gives a template-wise view of DLM compression, where dataset effects are separated into template exposure $\omega_d(U)$ and template-specific target-token distributions $q_d(\cdot \mid U)$. A derivation is provided in Appendix B.1.

3 Elasticity and Alignment in DLMs

We now use the template-wise compression framework to analyze *elasticity* in DLMs, i.e., the tendency of aligned behavior to be partially reversed by subsequent counter-aligned updates. We first formalize the two-stage post-training protocol, and then establish the theory of DLM elasticity, revealing how perturbation data alters the compression rates on pretraining and alignment data.

3.1 Problem Setup: Counter-Aligned Updates

Consider a two-stage post-training protocol. Starting from a pretrained model p_{θ_0}, we first fine-tune on a positive alignment dataset $\mathcal{D}_+$ to obtain an aligned checkpoint p_{θ^+}. The second stage continues fine-tuning p_{θ^+} on counter-aligned data $\mathcal{D}_-$, whose target behavior conflicts with $\mathcal{D}_+$. If b examples from $\mathcal{D}_-$ are used in this second stage, the resulting post-perturbation model is denoted by p_{θ_b}:

$$p_{\theta_0} \xrightarrow{\mathcal{D}_+^{(*)}} p_{\theta^+}, \qquad p_{\theta^+} \xrightarrow{\mathcal{D}_-^{(b)}} p_{\theta_b}.$$

Post-alignment rebound means that the aligned behavior becomes weaker as the amount of counter-aligned data increases. A more extreme case is *inverse alignment*, where the perturbed model moves back toward the behavior of the original pretrained model p_{θ_0}, as measured by a discrepancy metric ρ. For the compression analysis, we represent the protocol using three data sources: pretraining data D_1, alignment data D_2, and perturbation data D_3. The finite perturbation size b is modeled through the relative perturbation scale $l = |D_3|/|D_2|$. We study how increasing l changes the template-wise compression rates of D_1 and D_2.

3.2 Formal Derivation of DLM Elasticity

Theorem 3.1 (Elasticity of DLMs)**.** *Let D_1, D_2, D_3 denote pretraining, alignment, and perturbation data, respectively. Let $k = |D_1|/|D_2| \gg 1$, $l = |D_3|/|D_2| \ll 1$, and $S(U;l) = k\omega_1(U) + \omega_2(U) + l\omega_3(U)$. Under the template-wise compression rate and the joint template token distribution $q_D(\cdot \mid U; l)$, define*

$$G_{d3}(U;l) = \sum_{a \in \mathcal{V}} q_d(a \mid U) \left(\frac{q_3(a \mid U)}{q_D(a \mid U; l)} - 1 \right),$$

and

$$\mathcal{G}_{d3}(l) = \sum_U \frac{\omega_d(U)\omega_3(U)}{S(U;l)} G_{d3}(U;l), \qquad \Lambda_{\text{eff}}(l) = -\frac{\mathcal{G}_{23}(l)}{\mathcal{G}_{13}(l)}.$$

Then, for $d \in \{1, 2\}$,

$$\frac{d}{dl} r_{q_D}^{D_d/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{d3}(l).$$

Consequently, if $\mathcal{G}_{13}(l) > 0$ and $\mathcal{G}_{23}(l) < 0$, then

$$\frac{d}{dl} r_{q_D}^{D_2/D}(l) = -\Lambda_{\text{eff}}(l) \frac{d}{dl} r_{q_D}^{D_1/D}(l), \qquad \frac{d}{dl} r_{q_D}^{D_1/D}(l) < 0 < \frac{d}{dl} r_{q_D}^{D_2/D}(l).$$

Theorem 3.1 gives the derivative of the compression rate with respect to the perturbation scale l. This derivative is determined by how perturbation data D_3 overlaps with each dataset D_d across templates. For each template U, $G_{d3}(U;l)$ measures whether perturbation data agrees or conflicts with dataset D_d at that template, and $\mathcal{G}_{d3}(l)$ combines these template-wise effects over all templates. Under the rebound sign condition, $\mathcal{G}_{13}(l) > 0$ means that perturbation data is aligned with pretraining data D_1 in the aggregate, while $\mathcal{G}_{23}(l) < 0$ means that perturbation data conflicts with alignment data D_2 in the aggregate. The effective coefficient $\Lambda_{\text{eff}}(l)$ tells how the change on alignment data D_2 compares with the change on pretraining data D_1, both in direction and in size. The proof is given in Appendix B.2.

Corollary 3.2 (Asymptotic Attenuation of DLM Elasticity)**.** *Under the assumptions of Theorem 3.1, define*

$$\widetilde{G}_{23}(U;l) = -G_{23}(U;l).$$

Assume that there exist constants $C_\omega, C_G > 0$, independent of $k = |D_1|/|D_2|$, such that for every template U with $\omega_3(U) > 0$,

$$\omega_2(U) \le C_\omega\, \omega_1(U), \qquad 0 \le \widetilde{G}_{23}(U;l) \le C_G\, G_{13}(U;l).$$

Assume that

$$\sum_U \frac{\omega_1(U)\omega_3(U)}{S(U;l)} G_{13}(U;l) > 0.$$

Then the effective coefficient satisfies:

$$0 \le \Lambda_{\text{eff}}(l) \le C_\omega C_G.$$

Consequently, as $k \to \infty$,

$$\Lambda_{\text{eff}}(l) = O(1), \qquad \frac{\Lambda_{\text{eff}}(l)}{k} \to 0.$$

This corollary completes the scaling analysis of the DLM elasticity coefficient: under template-comparable conflict, $\Lambda_{\text{eff}}(l)$ remains bounded even as the global data-size ratio $k = |D_1|/|D_2|$ grows. The proof is given in Appendix B.3.

3.3 Elasticity and Rebound

Implication for Rebound. The above results connect the compression analysis to post-alignment rebound. The rate $r_{q_D}^{D_2/D}(l)$ measures how well the perturbed model compresses the alignment data D_2. When this rate increases, the model assigns lower probability to the target tokens preferred by the alignment data, which corresponds to weaker aligned behavior. Theorem 3.1 shows that this increase depends on whether counter-aligned data and alignment data overlap on the same templates. Counter-aligned data does not affect all aligned behavior uniformly. Its effect depends on which templates it overlaps with and whether, under those templates, it favors target tokens that disagree with the alignment data.

For example, recall the template $U_{\text{pos-adj}}$, where the model predicts an adjective in a partially masked review, such as `The movie was surprisingly [MASK] and heartwarming`. If the alignment data D_2 favors positive tokens such as `good` under this template, whereas the perturbation data D_3 favors negative tokens such as `bad`, then D_2 and D_3 conflict on the same template. This conflict increases the compression rate of D_2, corresponding to weaker aligned sentiment behavior. By contrast, due to the stochasticity of response masking, perturbation data may not produce enough masked prediction instances for every alignment-supporting template. If the sampled perturbation instances rarely cover a template, that template contributes less to alignment erosion. This explains why DLM rebound can be gradual: since response masking is stochastic, perturbation data may only cover part of the alignment-supporting templates, and alignment erosion mainly occurs on templates where the target-token distribution of the perturbation data conflicts with that of the alignment data. A controlled DLM experiment finds stronger gradient conflict and alignment erosion when D_3 uses opposing targets on templates shared with D_2 in Appendix D.

Comparison with Autoregressive Elasticity. This mechanism differs from autoregressive elasticity theory. In ARMs, each token is predicted from its left prefix, so perturbations naturally act through a left-to-right prediction path. Prior compression-based analysis of ARMs (Ji et al., 2025) shows that the change in alignment-data compression is k-scaled relative to the change in pretraining-data compression, where $k = |D_1|/|D_2|$ is typically large. This scaling helps explain why small counter-aligned updates can cause large changes in aligned behavior. For DLMs, our analysis gives a different scaling relation. Theorem 3.1 shows that:

$$\frac{d}{dl} r_{q_D}^{D_2/D}(l) = -\Lambda_{\text{eff}}(l) \frac{d}{dl} r_{q_D}^{D_1/D}(l), \qquad \frac{d}{dl} r_{q_D}^{D_1/D}(l) < 0 < \frac{d}{dl} r_{q_D}^{D_2/D}(l).$$

where $\Lambda_{\text{eff}}(l)$ depends on template overlap and template-level disagreement between perturbation, pretraining, and alignment data. Under template-comparable overlap, Corollary 3.2 gives:

$$\Lambda_{\text{eff}}(l) = O(1), \qquad \frac{\Lambda_{\text{eff}}(l)}{k} \to 0.$$

Thus, the effective coefficient is lower order than the k-scaled coefficient in autoregressive elasticity. Under the same perturbation scale, this yields a lower alignment-data compression rate $r_{q_D}^{D_2/D}(l)$ relative to the k-scaled autoregressive case. Since a lower compression rate means assigning higher probability to the target tokens preferred by the alignment data (Delétang et al., 2024), this provides a theoretical explanation for the weaker rebound observed in DLMs.

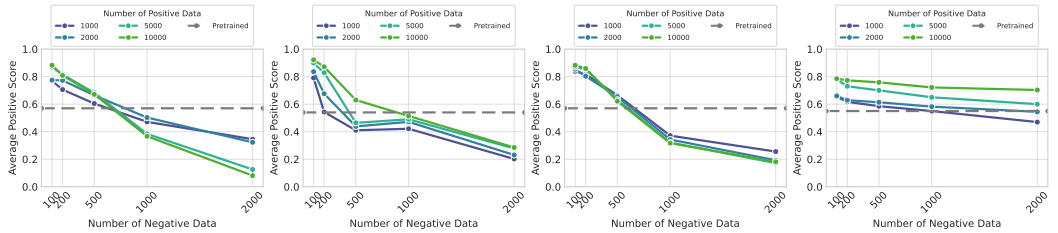

Figure 1: Experimental results demonstrating the divergent rebound dynamics on IMDb (left to right: Gemma-2B, Sdar-1.7B, Llama2-7B, and Llada-8B). After negative fine-tuning, ARMs (Gemma-2B, Llama2-7B) exhibit significant performance degradation. In contrast, DLMs (Sdar-1.7B, Llada-8B) show slower degradation and retain more alignment gains under the same negative data.

4 Experiments

We compare the post-alignment rebound of DLMs and ARMs under a unified two-stage protocol. Across diverse model families, alignment algorithms, model scale, and hyperparameters, we systematically measure how rapidly aligned model behavior deteriorates after reverse updates.

4.1 Experimental Setup

Experiment Design. The pretrained model is first fine-tuned on varying volumes of positive (e.g., safety) alignment data to yield a sequence of aligned checkpoints. Subsequently, each aligned checkpoint is subjected to counter-aligned perturbations via further fine-tuning on varying quantities of negative (e.g., unsafe) data.

Model Families. We consider Gemma (Team et al., 2024), Llama (Touvron et al., 2023), and Qwen (Bai et al., 2023) as autoregressive model families, and Mdm (Nie et al., 2024), Sdar (Cheng et al., 2025), Llada (Nie et al., 2025), and Wedlm (Liu et al., 2025) as diffusion model families. All models are evaluated under the same post-alignment protocol, data splits, and perturbation sizes.

Tasks and Datasets. We select two tasks: sentiment generation using IMDb (Maas et al., 2011) and single-turn safety alignment using BeaverTails (Ji et al., 2023). Following (Rafailov et al., 2023), IMDb prompts are constructed from the first 2–8 tokens of each text to condition generation. To systematically evaluate post-alignment fragility, we perform alignment fine-tuning using positive/safe sample sizes of $\{1000, 2000, 5000, 10000\}$. Subsequently, we apply counter-aligned updates using negative/unsafe perturbation sample sizes scaling across $\{100, 200, 500, 1000, 2000\}$ samples.

Evaluation and Metrics. We assess model generations on held-out test prompts using standardized automated evaluators. For sentiment generation, following (Rafailov et al., 2023), we utilize Sentiment RoBERTa (Hartmann et al., 2023), reporting the ratio of positive responses as the primary metric. For safe dialogue, we compute the average safety score across generated responses using the BeaverTails cost model (Dai et al., 2023). For details on hyperparameters please refer to Section E in Appendix.

4.2 Rebound Dynamics Differ between ARMs and DLMs

We investigate rebound dynamics across ARMs (Gemma-2B, Llama2-7B) and DLMs (Mdm-1028M, Sdar-1.7B, Llada-8B). As shown in Figures 1 and 2, the two paradigms exhibit starkly divergent behaviors. For ARMs fine-tuned with a larger amount of positive sample data, their performance drops quicker under only a small amount of negative sample fine-tuning. Conversely, DLMs yield substantially flatter degradation trajectories, robustly preserving their initial alignment gains under equivalent negative data volume.

Furthermore, DLMs demonstrate remarkable *ordering stability*, i.e., as the amount of negative data increases, the performance ordering induced by the amount of positive alignment data remains stable: models fine-tuned with more positive samples tend to achieve higher performance. In contrast, ARMs fail to preserve this ordering: models fine-tuned with more positive samples can degrade as quickly

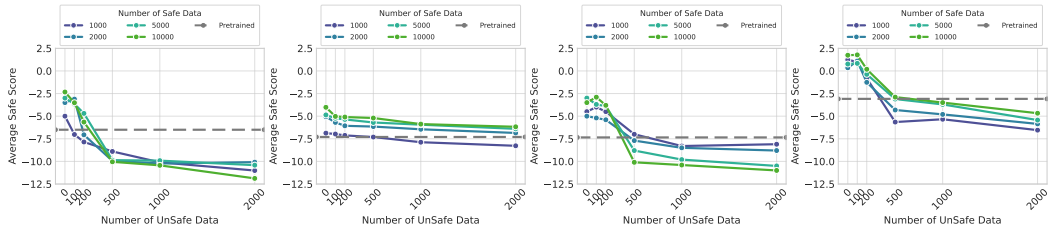

Figure 2: Experimental results demonstrating the divergent rebound dynamic on BeaverTails (left to right: Gemma-2B, Mdm-1028M, Llama2-7B, and Llada-8B). Following unsafe fine-tuning, ARMs (Gemma-2B, Llama2-7B) rapidly lose their safety advantage. DLMs (MDM-1028M, LLaDA-8B) degrade more slowly and preserve more of their safety gains under the same unsafe data.

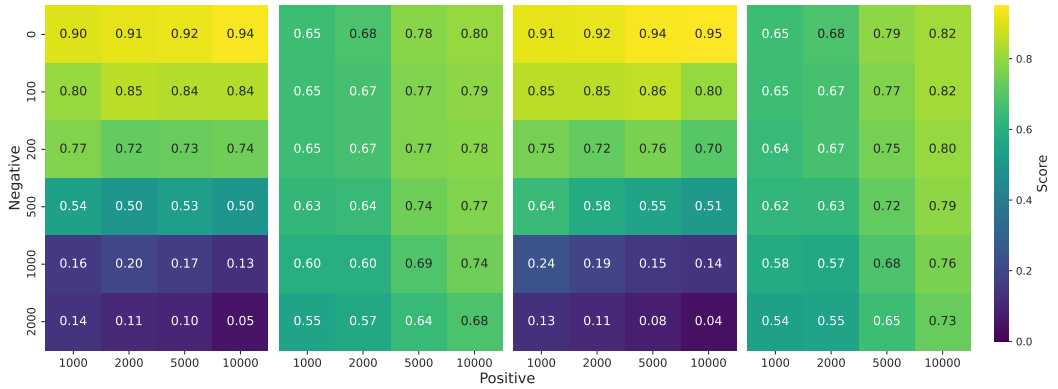

Figure 3: Experimental results validating rebound dynamics across different alignment algorithms. The heatmaps display, from left to right, the performance of Llama-2-7B DPO, Llada-8B DPO, Llama-2-7B KTO, and Llada-8B KTO algorithms after counter-aligned fine-tuning with varying amounts of positive and negative data. Lighter (yellower) colors indicate better model performance, while darker (purpler) colors indicate worse performance. Notably, DLM (Llada-8B) structurally retain alignment gains and ordering stability far better than ARM (Llama-2-7B), confirming this advantage is agnostic to the specific preference alignment algorithm.

as, or even faster than, those models fine-tuned with fewer positive samples. Further experimental details and results are provided in Appendix C.

4.3 Rebound Across Alignment Algorithms

To examine whether the ARM–DLM rebound gap persists across different alignment algorithms, we extend the analysis in Section 4.2 from supervised fine-tuning to preference-based post-training. Specifically, we conduct rebound experiments using Direct Preference Optimization (DPO) (Rafailov et al., 2023) and Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024). Since these objectives are not applied directly to the pre-trained model in our setup, each experiment consists of two stages: (a) forward alignment via SFT on positive data of varying scales, and (b) backward alignment via the corresponding preference-based objective on negative data of varying scales, where negative samples are treated as chosen responses and positive samples as rejected responses.

As shown in Figure 3, the qualitative rebound pattern remains consistent. Under both DPO and KTO, the autoregressive Llama2-7B suffers severe degradation, with larger positive alignment data sizes often leading to sharper post-perturbation collapses. Conversely, the diffusion model Llada-8B consistently exhibits weaker rebound and strictly preserves the relative advantage of larger positive-data budgets. Taken together, these results indicate that the observed ARM–DLM difference in rebound behavior is not specific to supervised fine-tuning, but persists across preference-based alignment algorithms.

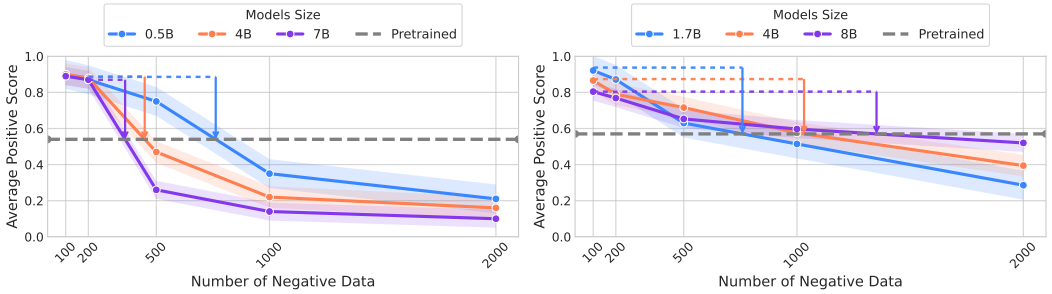

Figure 4: The impact of model scale on rebound dynamics for Qwen (left) and Sdar (right) on IMDb. Each curve traces the performance of a specific model scale as the amount of negative data increases, while the positive alignment data size is fixed. Shaded regions indicate the standard deviation. Larger ARMs (Qwen) suffer more severe degradation under counter-aligned fine-tuning, whereas larger DLMs (Sdar) demonstrate attenuated rebound, retaining more alignment-induced gains.

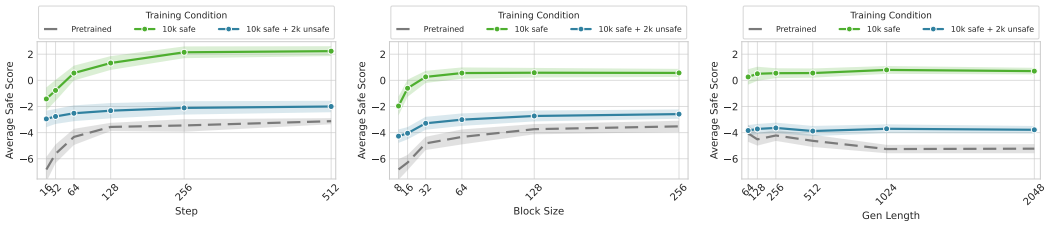

Figure 5: Inference-time modulation of rebound-related safety behavior in LLaDA-8B. From left to right, we independently vary `step`, `block_size`, and `gen_length`. Shaded regions indicate the standard deviation. While increasing `step` and `block_size` enhances the absolute safety scores up to a saturation point, both the aligned checkpoint (10,000 safe samples) and the post-perturbation checkpoint (further trained with 2,000 unsafe samples) consistently preserve a robust safety margin over the pretrained model across all configurations.

4.4 Factors Affecting Rebound

We conduct an in-depth analysis of the factors that influence the ARM-DLM rebound gap. Specifically, we investigate how model scale and inference-time hyperparameters affect rebound behavior.

Model Scale. As shown in Figure 4, within the Qwen ARM family, larger models exhibit a markedly steeper degradation trajectory under negative fine-tuning, rapidly erasing their alignment gains and regressing to the pretrained baseline with much less counter-aligned data (scale-amplified rebound). In contrast, the Sdar DLM family shows a weaker and less monotonic scale effect; larger models like Sdar-8B inherently retain more positive gain even with a large number of negative examples. Thus, while model scaling strengthens rebound in ARMs, this effect is structurally attenuated in DLMs.

Inference-Time Hyperparameters. We further examine how decoding hyperparameters modulate post-alignment behavior in DLMs. Specifically, using Llada-8B as a representative model, we independently vary `step`, `block_size`, and `gen_length` to isolate their effects. As shown in Figure 5, inference settings substantially affect the expression of safety behavior: increasing `step` yields the strongest improvement, `block_size` shows a threshold-and-saturation pattern, and `gen_length` has a weaker, non-monotonic effect. Crucially, the relative ordering of checkpoints remains stable across all settings: the aligned model performs best, and while the post-perturbation model exhibits a rebound after unsafe fine-tuning, it consistently outperforms the pretrained baseline. Therefore, although decoding hyperparameters modulate absolute safety scores, they do not disrupt the slower-rebound pattern observed in DLMs.

5 Conclusion

We studied post-alignment rebound in DLMs under unified post-training protocols. Across different tasks, perturbation sizes, model scales, alignment algorithms, and inference-time settings, DLMs consistently exhibit weaker rebound than ARMs: aligned behavior degrades more slowly after counter-aligned fine-tuning, and larger positive alignment sets induce more stable performance ordering as negative data increases. We explained this pattern through a template-wise compression theory of DLM elasticity, which shows that the effect of counter-aligned data depends on how it overlaps with the masked reconstruction structure of DLMs. Our experiment suggest that rebound is shaped not only by post-training data and objectives, but also by the underlying generation mechanism.

Acknowledgement

This work was supported by the Key R&D Program of Hubei Province under Grant 2024BAB038 and the National Key R&D Program of China under Grant 2023YFC3604702.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 34:17981–17993, 2021.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

Boyuan Chen, Donghai Hong, Jiaming Ji, Jiacheng Zheng, Bowen Dong, Jiayi Zhou, Kaile Wang, Juntao Dai, Xuyao Wang, Wenqi Chen, et al. Intermt: Multi-turn interleaved preference alignment with human feedback. *arXiv preprint arXiv:2505.23950*, 2025.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *ICLR*, 2024.

Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. Attacks, defenses and evaluations for llm conversation safety: A survey. In *ACL*, pages 6734–6747, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *NeurIPS*, 36:16693–16715, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *ACL*, pages 11575–11596, 2023.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87, 2023.

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. *arXiv preprint arXiv:2404.09937*, 2024.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Moongyu Jeon, Sangwoo Shin, Dongjae Jeon, and Albert No. Information-theoretic discrete diffusion. *arXiv preprint arXiv:2510.24088*, 2025.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *NeurIPS*, 36:24678–24704, 2023.

Jiaming Ji, Kaile Wang, Tianyi Alex Qiu, Boyuan Chen, Jiayi Zhou, Changye Li, Hantao Lou, Josef Dai, Yunhuai Liu, and Yaodong Yang. Language models resist alignment: Evidence from data compression. In *ACL*, pages 23411–23432, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *NeurIPS*, 35:4328–4343, 2022.

Aiwei Liu, Minghua He, Shaoxun Zeng, Sijun Zhang, Linhao Zhang, Chuhan Wu, Wei Jia, Yuan Liu, Xiao Zhou, and Jie Zhou. Wedlm: Reconciling diffusion language models with standard causal attention for fast inference. *arXiv preprint arXiv:2512.22737*, 2025.

Xiaoran Liu, Yuerong Song, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. Longllada: Unlocking long context capabilities in diffusion llms. In *AAAI*, volume 40, pages 32186–32194, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *NeurIPS*, 36:56998–57025, 2023.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 35:27730–27744, 2022.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 36:53728–53741, 2023.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *NeurIPS*, 37:130136–130184, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, et al. Self-conditioned embedding diffusion for text generation. *arXiv preprint arXiv:2211.04236*, 2022.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *NeurIPS*, 36:80079–80110, 2023.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Zhengyu Zhou and Weiwei Liu. An error analysis of flow matching for deep generative modeling. In *ICML*, 2025.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Xin Zou, Xiuwen Gong, and Weiwei Liu. Generalization bounds for out-of-distribution generalization. In *Forty-third International Conference on Machine Learning*, 2026.

A Related Work

Alignment Fragility and Post-Alignment Rebound. Training-based alignment methods, including supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF), and preference-optimization variants (Ouyang et al., 2022; Ziegler et al., 2019; Bai et al., 2022; Rafailov et al., 2023; Schulman et al., 2017; Shao et al., 2024; Guo et al., 2025), have become the dominant paradigm for steering large language models toward desired and safe behaviors. At the same time, a growing body of work has shown that aligned behavior can remain fragile after post-training: substantial degradation may arise not only under explicitly counter-aligned updates, but also unintentionally under seemingly benign fine-tuning (Qi et al., 2024; Yang et al., 2023; Hubinger et al., 2024). This phenomenon has been discussed through concepts such as inverse alignment, elasticity, and shallow safety alignment, and has been studied primarily in ARMs (Ji et al., 2025; Wei et al., 2023). These results suggest that strong immediate alignment does not necessarily imply robust retention under later fine-tuning (Hubinger et al., 2024; Qi et al., 2024). However, existing analyses remain largely centered on autoregressive architectures, leaving it unclear whether such rebound behavior is specific to single-path next-token generation or reflects a broader property of aligned language models.

Diffusion-Based Language Models. DLMs provide a generation paradigm distinct from autoregressive next-token prediction (Li et al., 2022; Austin et al., 2021). Existing approaches can be broadly grouped into continuous and discrete variants, depending on whether diffusion is performed in continuous representation space or directly over token-level states. In language modeling, continuous diffusion (Gulrajani and Hashimoto, 2023; Han et al., 2023; Li et al., 2022; Strudel et al., 2022; Gong et al., 2022; Lovelace et al., 2023; Zhou and Liu, 2025) has generally been more difficult to scale effectively (Lou et al., 2023), whereas recent progress in discrete diffusion (Austin et al., 2021; Ye et al., 2025; Liu et al., 2026), especially masked diffusion, has led to increasingly competitive results. Representative work such as Llada (Nie et al., 2025) shows that masked diffusion can be scaled to the multi-billion-parameter regime while approaching strong autoregressive baselines. More recent studies (Prabhudesai et al., 2025) have further examined diffusion models in data-constrained regimes, but the emphasis has remained on capability, scaling, and data efficiency rather than on post-alignment fragility. Consequently, whether rebound and inverse alignment in DLMs follow the same pattern as in ARMs remains largely unexplored.

B Assumptions and Proofs

B.1 Proof of Proposition 2.5

Proposition B.1 (Template-Wise Form)**.** *For dataset D_d, the template-based compression rate can be written as*

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U)\, \mathrm{CE}\big(q_d(\cdot \mid U), p_\theta(\cdot \mid U)\big),$$

where

$$\mathrm{CE}(q,p) = -\sum_{a\in\mathcal{V}} q(a)\log p(a).$$

Proof. For a masked-token prediction instance $(J, i, \tilde{x}_0^{J\cup I_2})$, define

$$U_{x_0,J,i} = \Phi(J, i, \tilde{x}_0^{J\cup I_2})$$

as the template assigned to this instance. By the definition of the template-based compression rate,

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \mathbb{E}_{x_0\sim D_d,\ J\sim p(J)} \left[\sum_{i\in I_1\setminus J} -\log p_\theta(x_0^i \mid U_{x_0,J,i}) \right]. \tag{1}$$

Since each masked-token prediction instance is assigned to exactly one template, the templates form a partition of the instance space. Hence, for every (x_0, J, i),

$$1 = \sum_U \mathbf{1}\{U_{x_0,J,i} = U\}. \tag{2}$$

Using Eq. (2), the inner loss term can be rewritten as

$$-\log p_\theta(x_0^i \mid U_{x_0,J,i}) = \sum_U \mathbf{1}\{U_{x_0,J,i} = U\} \left[-\log p_\theta(x_0^i \mid U)\right].$$

Substituting this identity into Eq. (1) gives

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \sum_U \mathbf{1}\{U_{x_0,J,i} = U\} \left[-\log p_\theta(x_0^i \mid U)\right] \right]. \tag{3}$$

Because the sum over templates is finite, or absolutely summable, we can exchange the order of summation and expectation:

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \sum_U \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \left[-\log p_\theta(x_0^i \mid U)\right] \right]. \tag{4}$$

Now fix a template U. We show that the expectation inside Eq. (4) equals a template exposure term times a cross-entropy term. By the definition of $q_d(a \mid U)$, for $\omega_d(U) > 0$,

$$q_d(a \mid U) = \frac{\mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \mathbf{1}\{x_0^i = a\} \right]}{\omega_d(U)}.$$

Equivalently,

$$\mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \mathbf{1}\{x_0^i = a\} \right] = \omega_d(U) q_d(a \mid U). \tag{5}$$

Using the token indicator expansion

$$-\log p_\theta(x_0^i \mid U) = \sum_{a \in \mathcal{V}} \mathbf{1}\{x_0^i = a\} \left[-\log p_\theta(a \mid U)\right],$$

the inner expectation in Eq. (4) becomes

$$\begin{aligned}
&\mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \left[-\log p_\theta(x_0^i \mid U)\right] \right] \\
&= \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \sum_{a \in \mathcal{V}} \mathbf{1}\{x_0^i = a\} \left[-\log p_\theta(a \mid U)\right] \right] \\
&= \sum_{a \in \mathcal{V}} \left[-\log p_\theta(a \mid U)\right] \mathbb{E}_{x_0 \sim D_d,\ J \sim p(J)} \left[\sum_{i \in I_1 \setminus J} \mathbf{1}\{U_{x_0,J,i} = U\} \mathbf{1}\{x_0^i = a\} \right] \\
&= \sum_{a \in \mathcal{V}} \left[-\log p_\theta(a \mid U)\right] \omega_d(U) q_d(a \mid U) \\
&= \omega_d(U) \sum_{a \in \mathcal{V}} q_d(a \mid U) \left[-\log p_\theta(a \mid U)\right] \\
&= \omega_d(U)\, \mathrm{CE}\big(q_d(\cdot \mid U), p_\theta(\cdot \mid U)\big).
\end{aligned} \tag{6}$$

Substituting Eq. (6) into Eq. (4), we obtain

$$r_{p_\theta}^{D_d/D} = \frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U)\, \mathrm{CE}\big(q_d(\cdot \mid U), p_\theta(\cdot \mid U)\big).$$

This is the desired template-wise form. □

B.2 Proof of Theorem 3.1

Assumption B.2 (Template mixture)**.** *For each template U, the joint template token distribution is*

$$q_D(a \mid U; l) = \frac{k\,\omega_1(U) q_1(a \mid U) + \omega_2(U) q_2(a \mid U) + l\,\omega_3(U) q_3(a \mid U)}{S(U; l)}, \tag{A.1}$$

where

$$S(U; l) = k\omega_1(U) + \omega_2(U) + l\omega_3(U), \qquad k = \frac{|D_1|}{|D_2|}, \qquad l = \frac{|D_3|}{|D_2|}.$$

The quantities $\omega_d(U)$ and $q_d(\cdot \mid U)$ are independent of l. Moreover, $q_D(a \mid U; l) > 0$ whenever $q_d(a \mid U) > 0$, and the template sums are finite, or absolutely summable, so that differentiation and summation can be exchanged.

Theorem B.3 (Elasticity of DLMs)**.** *Let D_1, D_2, D_3 denote pretraining, alignment, and perturbation data, respectively. Let $k = |D_1|/|D_2| \gg 1$, $l = |D_3|/|D_2| \ll 1$, and $S(U; l) = k\omega_1(U) + \omega_2(U) + l\omega_3(U)$. Under the template-wise compression rate and the joint template token distribution $q_D(\cdot \mid U; l)$, define*

$$G_{d3}(U; l) = \sum_{a \in \mathcal{V}} q_d(a \mid U) \left(\frac{q_3(a \mid U)}{q_D(a \mid U; l)} - 1 \right),$$

and

$$\mathcal{G}_{d3}(l) = \sum_U \frac{\omega_d(U)\omega_3(U)}{S(U; l)} G_{d3}(U; l), \qquad \Lambda_{\text{eff}}(l) = -\frac{\mathcal{G}_{23}(l)}{\mathcal{G}_{13}(l)}.$$

Then, for $d \in \{1, 2\}$,

$$\frac{d}{dl} r_{q_D}^{D_d/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{d3}(l).$$

Consequently, if $\mathcal{G}_{13}(l) > 0$ and $\mathcal{G}_{23}(l) < 0$, then

$$\frac{d}{dl} r_{q_D}^{D_2/D}(l) = -\Lambda_{\text{eff}}(l) \frac{d}{dl} r_{q_D}^{D_1/D}(l), \qquad \frac{d}{dl} r_{q_D}^{D_1/D}(l) < 0 < \frac{d}{dl} r_{q_D}^{D_2/D}(l).$$

Proof. By Assumption B.2, the joint template token distribution can be written as

$$q_D(a \mid U; l) = \frac{N(a, U; l)}{S(U; l)},$$

where

$$N(a, U; l) = k\omega_1(U) q_1(a \mid U) + \omega_2(U) q_2(a \mid U) + l\omega_3(U) q_3(a \mid U).$$

Since $\omega_d(U)$ and $q_d(\cdot \mid U)$ are independent of l, only the D_3 contribution depends on l. Hence

$$\frac{\partial}{\partial l} N(a, U; l) = \omega_3(U) q_3(a \mid U), \qquad \frac{\partial}{\partial l} S(U; l) = \omega_3(U).$$

Applying the quotient rule gives

$$\begin{aligned}
\frac{\partial}{\partial l} q_D(a \mid U; l) &= \frac{\left(\frac{\partial}{\partial l} N(a, U; l)\right) S(U; l) - N(a, U; l) \left(\frac{\partial}{\partial l} S(U; l)\right)}{S(U; l)^2} \\
&= \frac{\omega_3(U) q_3(a \mid U) S(U; l) - \omega_3(U) N(a, U; l)}{S(U; l)^2} \\
&= \frac{\omega_3(U)}{S(U; l)} \left(q_3(a \mid U) - \frac{N(a, U; l)}{S(U; l)} \right) \\
&= \frac{\omega_3(U)}{S(U; l)} \left(q_3(a \mid U) - q_D(a \mid U; l) \right).
\end{aligned} \tag{7}$$

Next, consider the template-wise compression rate

$$r_{q_D}^{D_d/D}(l) = \frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U) \mathrm{CE}(q_d(\cdot \mid U), q_D(\cdot \mid U; l)).$$

By the definition of cross-entropy,

$$\mathrm{CE}(q_d(\cdot \mid U), q_D(\cdot \mid U; l)) = -\sum_{a \in \mathcal{V}} q_d(a \mid U) \log q_D(a \mid U; l).$$

Since $q_d(\cdot \mid U)$ and $\omega_d(U)$ do not depend on l, differentiating with respect to l gives

$$\begin{aligned}
\frac{d}{dl} r_{q_D}^{D_d/D}(l) &= \frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U) \frac{\partial}{\partial l} \left[-\sum_{a \in \mathcal{V}} q_d(a \mid U) \log q_D(a \mid U; l) \right] \\
&= -\frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U) \sum_{a \in \mathcal{V}} q_d(a \mid U) \frac{\partial}{\partial l} \log q_D(a \mid U; l) \\
&= -\frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U) \sum_{a \in \mathcal{V}} q_d(a \mid U) \frac{1}{q_D(a \mid U; l)} \frac{\partial}{\partial l} q_D(a \mid U; l). \qquad (8)
\end{aligned}$$

Substituting Eq. (7) into Eq. (8) yields

$$\begin{aligned}
\frac{d}{dl} r_{q_D}^{D_d/D}(l) &= -\frac{H_{|I_1|}}{|I_1|} \sum_U \omega_d(U) \sum_{a \in \mathcal{V}} q_d(a \mid U) \frac{1}{q_D(a \mid U; l)} \frac{\omega_3(U)}{S(U; l)} \left(q_3(a \mid U) - q_D(a \mid U; l) \right) \\
&= -\frac{H_{|I_1|}}{|I_1|} \sum_U \frac{\omega_d(U)\omega_3(U)}{S(U; l)} \sum_{a \in \mathcal{V}} q_d(a \mid U) \left(\frac{q_3(a \mid U)}{q_D(a \mid U; l)} - 1 \right). \qquad (9)
\end{aligned}$$

Define

$$G_{d3}(U; l) = \sum_{a \in \mathcal{V}} q_d(a \mid U) \left(\frac{q_3(a \mid U)}{q_D(a \mid U; l)} - 1 \right).$$

Then Eq. (9) becomes

$$\frac{d}{dl} r_{q_D}^{D_d/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \sum_U \frac{\omega_d(U)\omega_3(U)}{S(U; l)} G_{d3}(U; l). \qquad (10)$$

Using

$$\mathcal{G}_{d3}(l) = \sum_U \frac{\omega_d(U)\omega_3(U)}{S(U; l)} G_{d3}(U; l),$$

we obtain

$$\frac{d}{dl} r_{q_D}^{D_d/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{d3}(l). \qquad (11)$$

This proves the derivative identity for any dataset D_d, and in particular for $d \in \{1, 2\}$.

Taking $d = 2$ in Eq. (11) gives

$$\frac{d}{dl} r_{q_D}^{D_2/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{23}(l), \qquad (12)$$

and taking $d = 1$ gives

$$\frac{d}{dl} r_{q_D}^{D_1/D}(l) = -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{13}(l). \qquad (13)$$

If $\mathcal{G}_{13}(l) \neq 0$, the effective coefficient is well-defined as

$$\Lambda_{\text{eff}}(l) = -\frac{\mathcal{G}_{23}(l)}{\mathcal{G}_{13}(l)}.$$

Multiplying Eq. (13) by $-\Lambda_{\text{eff}}(l)$, we get

$$\begin{aligned}
-\Lambda_{\text{eff}}(l) \frac{d}{dl} r_{q_D}^{D_1/D}(l) &= -\left(-\frac{\mathcal{G}_{23}(l)}{\mathcal{G}_{13}(l)} \right) \left(-\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{13}(l) \right) \\
&= -\frac{H_{|I_1|}}{|I_1|} \mathcal{G}_{23}(l) \\
&= \frac{d}{dl} r_{q_D}^{D_2/D}(l), \qquad (14)
\end{aligned}$$

where the last equality follows from Eq. (12). Therefore,

$$\frac{d}{dl} r_{q_D}^{D_2/D}(l) = -\Lambda_{\text{eff}}(l) \frac{d}{dl} r_{q_D}^{D_1/D}(l).$$

Finally, since

$$\frac{H_{|I_1|}}{|I_1|} > 0,$$

Eq. (13) implies that

$$\mathcal{G}_{13}(l) > 0 \quad \Longrightarrow \quad \frac{d}{dl} r_{q_D}^{D_1/D}(l) < 0.$$

Similarly, Eq. (12) implies that

$$\mathcal{G}_{23}(l) < 0 \quad \Longrightarrow \quad \frac{d}{dl} r_{q_D}^{D_2/D}(l) > 0.$$

Thus, under the rebound sign condition

$$\mathcal{G}_{13}(l) > 0, \qquad \mathcal{G}_{23}(l) < 0,$$

we have

$$\frac{d}{dl} r_{q_D}^{D_1/D}(l) < 0 < \frac{d}{dl} r_{q_D}^{D_2/D}(l).$$

This completes the proof. □

B.3 Proof of Corollary 3.2

Corollary B.4 (Template-Comparable Coupling)**.** *Under the assumptions of Theorem 3.1, suppose that the aggregate pretraining-attraction term is positive:*

$$B_{13}(l) := \sum_U \frac{\omega_1(U)\omega_3(U)}{S(U;l)} G_{13}(U;l) > 0.$$

Assume further that, for every template U with nonzero perturbation exposure $\omega_3(U) > 0$, the following template-comparability conditions hold:

$$0 \le \omega_2(U) \le C_\omega \, \omega_1(U),$$

and

$$0 \le \widetilde{G}_{23}(U;l) \le C_G \, G_{13}(U;l), \qquad \widetilde{G}_{23}(U;l) := -G_{23}(U;l),$$

where C_ω and C_G are constants independent of k. Then

$$0 \le \Lambda_{\text{eff}}(l) \le C_\omega C_G.$$

Consequently,

$$\Lambda_{\text{eff}}(l) = O(1), \qquad \frac{\Lambda_{\text{eff}}(l)}{k} \to 0 \quad \textit{as } k \to \infty.$$

Thus, under template-comparable conflict, the DLM coupling is asymptotically smaller than the k-scaled coupling in autoregressive elasticity.

Proof. Because $G_{23}(U;l) \le 0$ under alignment-conflicting templates, define the positive conflict magnitude

$$\widetilde{G}_{23}(U;l) = -G_{23}(U;l) \ge 0.$$

Then the effective coefficient can be written as

$$\Lambda_{\text{eff}}(l) = \frac{\sum_U \frac{\omega_2(U)\omega_3(U)}{S(U;l)} \widetilde{G}_{23}(U;l)}{\sum_U \frac{\omega_1(U)\omega_3(U)}{S(U;l)} G_{13}(U;l)}.$$

The denominator is $B_{13}(l) > 0$ by assumption. Using the template-comparability conditions,

$$\omega_2(U) \le C_\omega \omega_1(U), \qquad \widetilde{G}_{23}(U;l) \le C_G G_{13}(U;l),$$

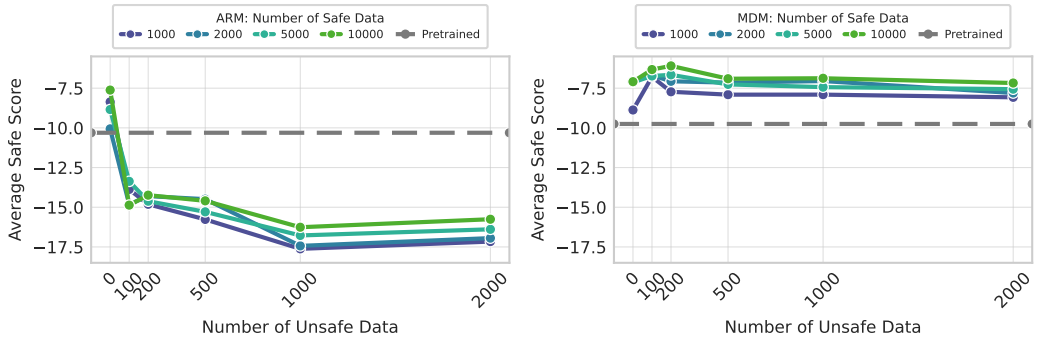

Figure 6: Backbone- and pretraining-matched comparison on BeaverTails (left: ARM-1028M-100e18; right: MDM-1028M-100e18). Each curve traces performance across increasing amounts of negative/unsafe data, given a fixed amount of positive/safe alignment data. Under the same data and evaluation protocol, the ARM rapidly falls below its pretrained baseline, whereas the MDM remains above its baseline across all evaluated combinations of safe and unsafe data sizes.

we obtain

$$\sum_U \frac{\omega_2(U)\omega_3(U)}{S(U;l)} \widetilde{G}_{23}(U;l) \le C_\omega C_G \sum_U \frac{\omega_1(U)\omega_3(U)}{S(U;l)} G_{13}(U;l)$$
$$= C_\omega C_G B_{13}(l).$$

Therefore,

$$0 \le \Lambda_{\text{eff}}(l) \le C_\omega C_G.$$

Since C_ω and C_G do not depend on k, this implies

$$\Lambda_{\text{eff}}(l) = O(1).$$

Hence, as $k \to \infty$,

$$\frac{\Lambda_{\text{eff}}(l)}{k} \le \frac{C_\omega C_G}{k} \to 0.$$

This proves the claim. □

C Additional Experiment Results

Backbone- and Pretraining-Matched ARM–MDM Comparison. To further reduce model-family confounds in the comparison between ARMs and DLMs, we conduct a paired experiment using `ar-1028M-100e18` and `mdm-1028M-100e18` from the same SMDM model suite (Nie et al., 2024). The two checkpoints have the same Transformer backbone dimensions, tokenizer, 1.028B non-embedding parameters, SlimPajama pretraining corpus, approximately 16.21B pretraining tokens, and nominal pretraining compute of 100×10^{18} FLOPs. In contrast, the MDM-1028M result in Figure 2 uses the more extensively pretrained `mdm-1028M-1600e18` checkpoint, whereas we use `mdm-1028M-100e18` here to match the ARM's pretraining compute. The primary difference between the two checkpoints in the present comparison is the modeling paradigm: the ARM uses causal attention and next-token prediction, whereas the MDM uses bidirectional context and masked denoising. We evaluate both models using the same two-stage fine-tuning protocol and evaluation setup described in Section 4.1.

Figure 6 shows a consistent ARM–MDM rebound gap across all four safe-alignment data sizes. After only 100 unsafe examples, all ARM trajectories drop sharply below the pretrained baseline, whereas the MDM trajectories exhibit substantially slower rebound and remain above the baseline throughout the evaluated range. The MDM also largely preserves the performance ordering induced by the amount of safe alignment data, consistent with the ordering stability observed in the main experiments.

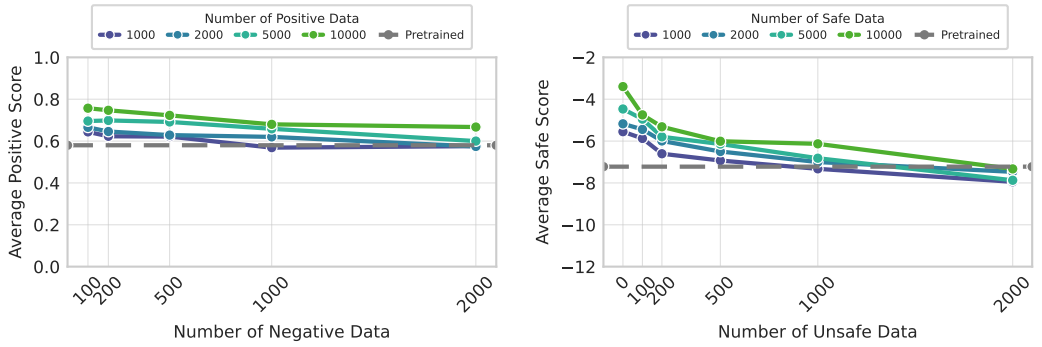

Figure 7: Experimental results for Wedlm-7B under counter-aligned fine-tuning (left: IMDb, right: BeaverTails). Each curve traces performance across increasing amounts of negative/unsafe data, given a fixed amount of positive/safe alignment data. Across both tasks, Wedlm-7B shows weaker degradation and maintains ordering stability under counter-aligned fine-tuning. This pattern is consistent with the results observed on other DLM backbones.

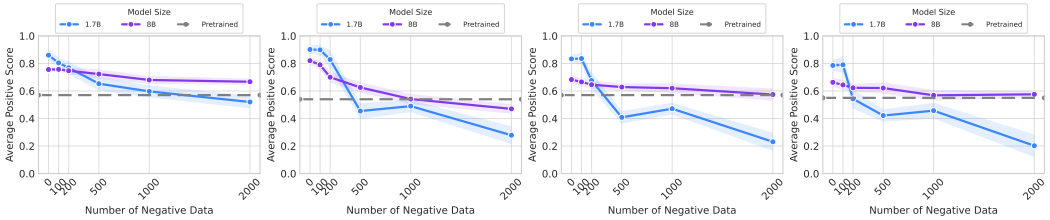

Figure 8: Rebound under counter-aligned perturbations for the Sdar family at different positive-alignment data sizes. From left to right, the four panels fix the positive alignment data size at 10000, 5000, 2000, and 1000 samples, respectively. As the amount of negative perturbation data increases, Sdar-8B shows weaker rebound than Sdar-1.7B while maintaining ordering stability.

Ablations on the Different Model This experiment tests whether the weaker-rebound pattern observed in the main text extends to an additional DLM. Figure 7 shows that WeDLM-7B exhibits the same qualitative behavior on both IMDb and BeaverTails. Models fine-tuned on larger positive/safe alignment sets consistently achieve higher post-perturbation performance as the amount of counter-aligned data increases. These results provide further evidence that weaker rebound is not limited to the main DLM families studied in the paper, and are consistent with our interpretation that DLMs respond more stably to counter-aligned updates.

Additional Results on Rebound under Model Scaling in the DLM Family To examine whether rebound varies systematically with model scale within the same diffusion model family, we compare Sdar-1.7B and Sdar-8B under different positive-alignment data sizes. Figure 8 shows that, as the amount of counter-aligned data increases, the larger Sdar model degrades more slowly across positive-alignment data sizes. This provides additional evidence that, within a fixed diffusion architecture, larger model scale is associated with weaker rebound and more stable post-perturbation performance.

D Controlled Test of Template Overlap and Alignment Erosion

We use a controlled experiment to test whether template overlap changes the effect of perturbation target direction on alignment erosion. We first vary whether perturbation data D_3 uses templates U also present in alignment data D_2. A template in D_3 is *Shared* if it also occurs in D_2; otherwise, it is *Disjoint*. Recall $U_{\text{pos-adj}}$, illustrated by `The movie was surprisingly [MASK] and heartwarming`. A D_3 example may use different words yet be assigned to this same U; its template is then *Shared*. By contrast, `It was [MASK] to experience the acting as a critic` illustrates a template used in D_3 but absent from D_2; this template is *Disjoint*.

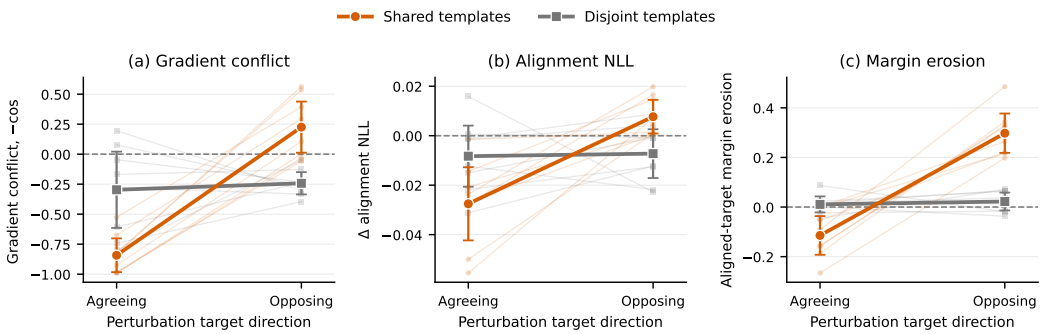

Figure 9: Controlled test of whether template reuse amplifies gradient conflict and alignment erosion. Shared means that perturbation data D_3 uses templates U present in alignment data D_2; Disjoint means that it uses templates absent from D_2. Agreeing and Opposing targets follow and reverse, respectively, each template's reference direction. Under Shared, Opposing targets yield stronger (a) gradient conflict, (b) increases in alignment evaluation NLL, and (c) aligned-target margin erosion than Agreeing targets. These differences are small under Disjoint. Gradient conflict is measured by the negative cosine between alignment and perturbation gradients. All conditions use one update with $\|\Delta\theta\|_2 = 0.002$. Light lines show eight paired seeds; bold lines show their means, and error bars denote 95% confidence intervals. All three interactions have the predicted direction in every seed (two-sided exact sign-flip tests, $p = 0.0078$ each).

We also vary the perturbation target direction for each template. An *Agreeing* perturbation favors targets in the template's reference direction, whereas an *Opposing* perturbation favors targets in the opposite direction. For a Shared template, D_2 determines the reference direction during alignment. For example, if D_2 favors positive targets such as `good` under $U_{\text{pos-adj}}$, a D_3 example on the same U is Shared–Agreeing when it favors positive targets and Shared–Opposing when it favors negative targets such as `bad`. For the Disjoint template above, suppose the reference direction is positive. Favoring positive targets such as `good` then gives Disjoint–Agreeing, whereas favoring negative targets such as `bad` gives Disjoint–Opposing. The latter does not imply conflict with D_2 at this U, since the template is absent from D_2.

We instantiate four Shared and four Disjoint templates, with no exact masked input repeated across alignment, perturbation, and evaluation. Starting from the aligned MDM-1028M checkpoint, each condition receives one update with $\|\Delta\theta\|_2 = 0.002$. Figure 9 shows that Opposing targets produce stronger gradient conflict, a larger increase in alignment evaluation NLL, and greater aligned-target margin erosion than Agreeing targets when templates are Shared. For Disjoint templates, switching from Agreeing to Opposing changes these three measures only slightly. This pattern supports the predicted dependence of alignment erosion on template overlap and perturbation target direction in this controlled task.

E Implementation Details

Unless otherwise specified, all rebound experiments follow a two-stage full-parameter SFT protocol. In the first stage, each pretrained model is fine-tuned for one epoch on different amounts of positive/safe data, producing aligned checkpoints with different positive-alignment data sizes. In the second stage, each aligned checkpoint is further fine-tuned for one epoch on negative/unsafe data of varying sizes. The same data sizes and splits are strictly matched across ARMs and DLMs.

Comprehensive training hyperparameters are summarized in Table 1. While both generation paradigms share an identical optimization setup (e.g., learning rate, batch size, and precision settings) to ensure a fair comparison, they fundamentally differ in their training objectives: ARMs optimize standard next-token likelihood, whereas DLMs optimize masked-token reconstruction over partially observed sequences.

Table 1: SFT hyperparameters for ARMs and DLMs.

Hyperparameter	ARMs	DLMs
Forward alignment stage	1 epoch	1 epoch
Counter-alignment stage	1 epoch	1 epoch
Optimizer	AdamW	AdamW
Learning rate	1×10^{-5}	1×10^{-5}
Weight decay	0.0	0.0
Scheduler	cosine, no warmup	cosine, no warmup
Mixed precision	`bf16`	`bf16`
TF32	enabled	enabled
Per-device batch size	2	2
Gradient accumulation	4	4
Max sequence length	512	512
Training objective	next-token likelihood	masked-token reconstruction
Masking strategy	–	random masking
Mask ratio	–	$[0.1, 0.9]$

F Limitations

This work focuses on post-alignment rebound, because it can be directly measured under controlled positive-to-negative fine-tuning protocols. We do not provide a complete empirical study of resistance in DLMs, which would require controlled forward and reverse alignment experiments across comparable checkpoints. A second limitation is the lack of controlled pretraining-scale experiments for DLMs. Prior work on ARMs suggests that rebound may depend on pretraining data volume, but testing the analogous effect in DLMs would require model families with the same architecture and training setup but different pretraining data sizes. To our knowledge, such public DLM checkpoints are not currently available. Therefore, our results should be read as a first DLM-specific analysis of rebound, rather than a complete account of every form of elasticity in DLMs. The experiments and theory together show a consistent weaker-rebound pattern across the models, tasks, and post-training settings studied here.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We clearly delineate this work's contributions in Introduction, as well as in the abstract.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All assumptions and results are clearly stated, and all proofs are provided in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide a detailed algorithm and description of the exact setup of our experiments.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [No]

Justification: All datasets and pretrained models used in this study are publicly available.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Experimental settings and hyperparameters are well described.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All of our experimental results come from the mean of five trials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the details of the compute resources in Appendix E.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our work adheres to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: There are no potential societal consequences of our work.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets are properly mentioned and cited in Section 4.1.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.