# STABILIZING LANGUAGE MODELS UNDER CONTINUAL LEARNING VIA CONDITION-ANCHORED DISTILLATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Continual adaptation of language models can change their output distribution on prompts learned earlier, while retaining every old prompt–answer pair may be undesirable or impossible. We study *condition-anchored generative distillation* (CAGD): retain a small set of old prompts, use a frozen previous model to reconstruct completions and generation states, and match its predictive distributions while learning the next task. The formulation separates three roles that ordinary replay conflates: conditions select the behavior to protect, teacher generations locate relevant states, and soft targets specify how predictions may change. For autoregressive language generation, teacher-rollout distillation admits an exact chain-rule decomposition of sequence divergence. For masked-diffusion language modeling, our implementation directly controls local denoising drift on teacher-generated completions. In continual adaptation of a 219M masked diffusion language model, CAGD reduces four-task final held-out loss from 2.927 to 1.114 in one task order and from 2.168 to 0.891 in exact reverse. The same soft targets, in matched Qwen3-0.6B and 1.7B interventions, improve mean final GSM8K exact match by 3.36 and 1.82 percentage points and conditional retention by 11.38 and 4.97 points over hard replay. The direction persists on fresh facts and natural instructions across SMDM and Qwen3. In a longer two-boundary stream, Qwen answer-format retention improves. Together with an eight-task TRACE comparison, these results support condition-anchored functional preservation across the tested language-generation objectives. Code is available.

## 1 INTRODUCTION

Continual learning for language-generation models arises after deployment: a model receives a stream of domains, instruction families, or factual updates while users still expect earlier conditions to remain useful. We study this as a sequence of conditional language distributions $p_{\theta_t}(y \mid c)$. Sequential optimization can move an old conditional distribution and produce catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999; Parisi et al., 2019). The retention target is richer than a class label: one condition may admit many valid answers, styles, and generation paths, so preserving a behavior cannot be reduced to replaying one old label or one fixed string.

Model-side continual learning commonly uses regularization, replay, or architectural isolation (Guo et al., 2025; Zheng et al., 2024). Parameter regularization protects a previous solution through importance estimates, while functional regularization constrains its predictions (Kirkpatrick et al., 2017; Li & Hoiem, 2016). Replay instead revisits stored examples or teacher-generated substitutes (Shin et al., 2017; Sun et al., 2020a;b). These approaches offer different ways to balance retention and acquisition, but they also make different choices about what represents an old task. For a conditional generator, protecting selected weights, fitting a finite set of strings, and preserving an output distribution are related objectives rather than interchangeable ones. This distinction matters when old prompt–answer pairs cannot be retained in full and the model must continue learning through shared parameters.

The resulting design question is both *where to compare* the old and new models and *what to match* there. A generated answer provides a concrete training target, but fitting its tokens does not explicitly constrain the probabilities of alternative continuations. Conversely, matching teacher predictions requires a set of prompts and intermediate states on which those predictions are relevant. Generation



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.

and distillation thus play complementary roles: the former locates support, while the latter specifies the behavior to preserve on that support. An end-to-end replay gain can reflect either role. Separating them motivates a comparison that varies target information on fixed generated support.

We take the condition itself as the persistent index of an old behavior. A small, task-balanced cache stores old conditions but not their outputs. At each task boundary, the frozen previous model reconstructs completions for those anchors. Teacher and student are then evaluated at shared teacher-induced generation states, and the student matches the teacher's soft conditional distributions while learning current data. We call this *condition-anchored generative distillation* (CAGD). The condition identifies a behavior worth revisiting without making its original answer the permanent memory. Completions can be reconstructed at each stage, while soft targets constrain the student's predictions beyond the generated token alone.

The same construction applies at different generation states. In an autoregressive (AR) language model, the states are prompt–completion prefixes and the transitions are next-token distributions. In a masked-diffusion language model, a teacher completion is randomly corrupted, and teacher and student distributions are matched at masked answer positions (Sahoo et al., 2024; Nie et al., 2025). This state-based view also clarifies the connection to retained behavior: matching distributions along sampled AR teacher paths decomposes sequence divergence, whereas the implemented masked-diffusion objective directly controls denoising drift. The analysis identifies which generation-process assumptions connect these local constraints to output stability.

Our evaluation moves from controlled attribution to broader task behavior. SMDM factual streams, exact order reversals, and fresh facts test retention under changes in exposure and content. Matched SMDM and Qwen interventions isolate soft targets from hard generated replay at fixed generated support; the Qwen study measures whether that distinction changes GSM8K answer retention. Dolly instructions extend the comparison to question answering, summarization, and creative writing. Qwen3-0.6B and 1.7B test the AR specialization. An eight-task TRACE comparison adds heterogeneous continual adaptation on Qwen3-4B. We assess retention and new-task acquisition throughout.

**Contributions:**

- We formulate continual language-model adaptation around condition anchors and a common teacher-induced state objective, with explicit AR and masked-diffusion language specializations.
- We connect transition matching to output-behavior stability, giving an exact AR path decomposition and the local guarantee supported by the masked-diffusion implementation, while exposing finite-anchor and teacher errors.
- Matched-support interventions isolate the effect of soft targets: they improve SMDM loss and mean Qwen GSM8K exact-match and conditional retention over hard replay at both tested AR scales. Broader task-order, fresh-content, natural-instruction, and TRACE studies report retention together with new-task acquisition and unedited generations.

## 2 Related Work

### 2.1 Functional Regularization for Continual Learning

Continual-learning methods often protect a previous solution either in parameter space or function space. Elastic weight consolidation (EWC) penalizes changes to parameters deemed important by a Fisher approximation (Kirkpatrick et al., 2017); related methods use structured curvature or other importance signals (Ritter et al., 2018; Zenke et al., 2017; Aljundi et al., 2018). Learning without Forgetting instead asks the updated model to reproduce the previous model's outputs (Li & Hoiem, 2016). Functional constraints are especially natural for conditional generators because the protected object is the distribution $p(y \mid c)$, not a unique parameter vector. Our method uses that perspective.

### 2.2 Generative Replay for Continual Language Models

Deep generative replay replaces unavailable old observations with samples from a learned generator (Shin et al., 2017; van de Ven & Tolias, 2018). LAMOL lets a language model generate pseudo-

examples from earlier tasks (Sun et al., 2020a); Distill and Replay combines replay with teacher–student transfer (Sun et al., 2020b). Prompt-conditioned generative replay has also been developed for lifelong task-oriented dialogue (Zhao et al., 2022). These precedents mean that retaining prompts, producing old-task completions, and applying distillation are established but usually studied as separate design choices. CAGD supplies a common accounting of their roles and compares soft transition matching against one-hot learning on the *same* generated support. This matched contrast is necessary to distinguish distribution preservation from ordinary pseudo-example replay.

Self-Distillation Fine-Tuning (SDFT) instead samples on-policy student trajectories and matches a demonstration-conditioned EMA teacher, learning a current task from expert demonstrations while reducing forgetting (Shenfeld et al., 2026). CAGD uses a different teacher and support: a frozen previous checkpoint generates states for retained old-task prompts rather than supervising current-task student rollouts from paired demonstrations.

Continual instruction-tuning benchmarks expose broad task heterogeneity (Wang et al., 2023; Zhang et al., 2023). Our reversal streams serve a different purpose: disjoint facts, balanced anchors, prompt disjoint evaluation, and exact order reversal isolate retention and plasticity. We use them as mechanism tests, then assess transfer with Dolly instructions and the eight-task TRACE stream.

### 2.3 Continual Learning for Diffusion Models

Diffusion-based replay has been used to synthesize observations for continual classifiers and dense predictors (Gao et al., 2023; Chen et al., 2023; Kim et al., 2024). Other work studies continual customization of text-to-image diffusion models (Smith et al., 2023). Most closely related, generative distillation matches a previous diffusion model across its reverse process (Masip et al., 2025). In language, diffusion objectives range from continuous representations to absorbing-mask processes (Austin et al., 2021; Li et al., 2022; Gong et al., 2023; Sahoo et al., 2024; Lou et al., 2024).

We connect these lines through condition-indexed state matching and test retention on AR prefixes and masked-diffusion corruptions under matched anchor and generated-support controls.

## 3 Condition-Anchored Generative Distillation

### 3.1 Continual Conditional Generation

Let tasks arrive as datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$ over conditions and outputs $(c, y)$. After task $t$ the model defines $P_{\theta_t}(y \mid c)$. When task $t+1$ arrives, its complete training data are available but previous outputs are not. The learner may retain a compact set of conditions

$$\mathcal{A}_{\le t} = \bigcup_{j=1}^{t} \mathcal{A}_j, \qquad \mathcal{A}_j = \{c_{j,1}, \ldots, c_{j,m_j}\}, \tag{1}$$

which we call *condition anchors*. In language generation these are prompts. They index which old behaviors should be revisited without fixing a single answer as the behavior itself.

For an old task $i$, let $R_i(\theta) = \mathbb{E}_{(c,y)\sim\mathcal{D}_i}[\ell(\theta; c, y)]$ be an evaluation risk. We report the final average risk and past-task forgetting

$$R_{\text{final}} = \frac{1}{T}\sum_{i=1}^{T} R_i(\theta_T), \qquad \mathcal{F}_{\text{past}} = \frac{1}{T-1}\sum_{i<T} \left[R_i(\theta_T) - R_i(\theta_i)\right]. \tag{2}$$

Because a model can reduce forgetting merely by refusing to learn, every retention result is paired with the final task's risk $R_T(\theta_T)$ as a plasticity endpoint.

At the beginning of stage $t > 1$, we freeze $\theta^- = \theta_{t-1}$ as a temporary teacher. This teacher represents the behavior that can still be recovered at the stage boundary; CAGD cannot repair knowledge that $\theta^-$ has already lost. Anchor sets persist across stages, whereas the teacher snapshot and its generated outputs are needed only while training the current stage.

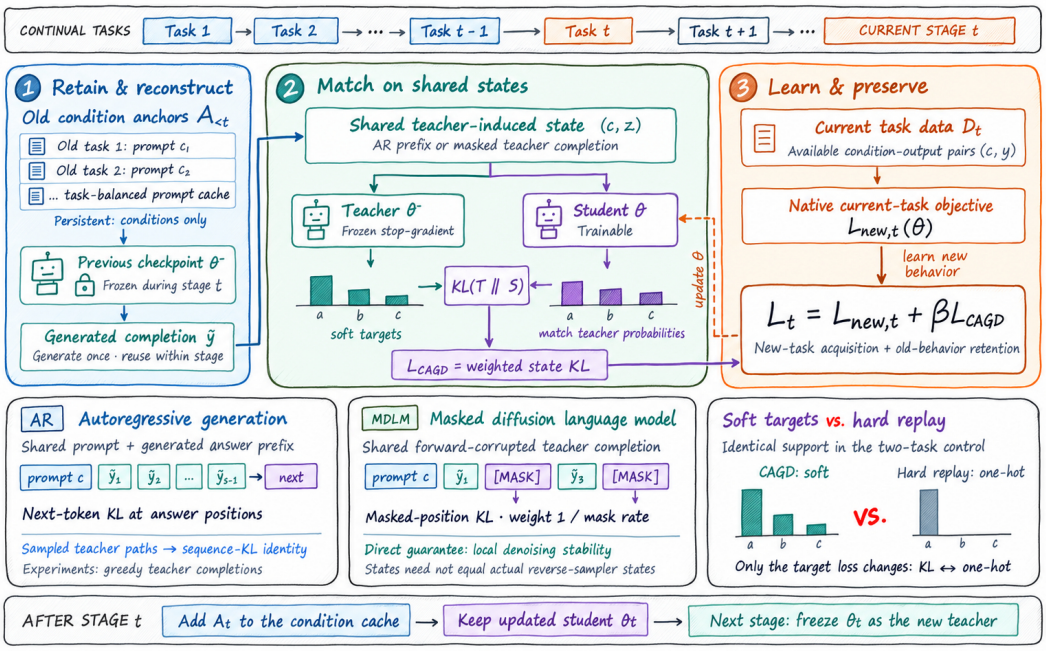


Figure 1: CAGD retains conditions rather than old outputs. A frozen teacher reconstructs completions and relevant states; the student matches its predictive distributions while learning current data. AR prefixes instantiate next-token KL. The masked-diffusion language-model branch uses denoiser KL on corrupted teacher completions.

### 3.2 Distillation on Teacher-Generated Trajectories

For each $c \in \mathcal{A}_{<t}$, the teacher generates a completion $\widetilde{y}$ and a collection of relevant generation states $\tau = (z_s)_{s\in\mathcal{S}}$. Let $K_{\theta,s}(\cdot \mid z_s, c)$ denote the model's predictive distribution at state $z_s$, and let $w_s \geq 0$ be the native objective weight. The common distillation term is

$$\mathcal{L}_{\mathrm{CAGD}}(\theta;\theta^-) = \mathbb{E}_{\substack{c\sim\widehat{\mathcal{A}}_{<t} \\ \tau\sim Q_{\theta^-}(\cdot|c)}} \left[ \sum_{s\in\mathcal{S}} w_s \, D_{\mathrm{KL}}\big(K_{\theta^-,s}(\cdot \mid z_s, c) \,\|\, K_{\theta,s}(\cdot \mid z_s, c)\big) \right]. \tag{3}$$

Here $\widehat{\mathcal{A}}_{<t}$ is the task-balanced empirical anchor distribution and $Q_{\theta^-}$ is the teacher-induced state distribution. The stage objective is

$$\mathcal{L}_t(\theta) = \mathcal{L}_{\mathrm{new},t}(\theta) + \beta \mathcal{L}_{\mathrm{CAGD}}(\theta;\theta^-). \tag{4}$$

No parameter-space penalty is part of the method.

**Autoregressive specialization.** For an AR model, the teacher produces $\widetilde{y} = (\widetilde{y}_1, \ldots, \widetilde{y}_L)$, $z_s = \widetilde{y}_{<s}$, and $K_{\theta,s} = p_\theta(y_s \mid c, \widetilde{y}_{<s})$. Equation (3) becomes token-level teacher–student KL over answer positions. Sampling complete trajectories from the teacher gives the population objective in the pathwise decomposition below. Our Qwen experiments use greedy teacher completions to hold generated support fixed across paired methods; they therefore match one deterministic high-probability trajectory per anchor rather than Monte Carlo sampling the full teacher distribution.

**Conditional diffusion specialization.** For a general conditional diffusion generator, $z_s$ can be a reverse-process state and $K_{\theta,s}$ its reverse transition kernel. Matching those actual kernels yields a pathwise guarantee. The masked diffusion language model used in our experiments implements a related but narrower state objective. Write $x = (c, \widetilde{y})$, let $A$ index completion tokens, draw $t \sim \mathcal{U}(t_{\min}, 1)$, and independently replace answer tokens with `[MASK]` with probability $t$. If $M_t \subseteq A$ is the masked set, then

$$\mathcal{L}^{\mathrm{MDLM}}_{\mathrm{CAGD}} = \mathbb{E}\left[ \frac{1}{|A|} \sum_{i\in M_t} \frac{1}{t} D_{\mathrm{KL}}(p_{\theta^-}(x_i \mid x_t) \,\|\, p_\theta(x_i \mid x_t)) \right]. \tag{5}$$

Prompts remain visible, only completion positions are corrupted, and the $1/t$ importance weight matches the native masked-diffusion objective (Sahoo et al., 2024; Nie et al., 2025). Empty masks contribute zero. This construction evaluates denoisers on forward corruptions of a teacher-generated completion; these states are not assumed to be the realized states of every reverse sampler.

**Stage procedure.** Anchors are balanced across old tasks and, in the factual experiments, across facts. One teacher completion per anchor is generated once per stage and reused; teacher and student are evaluated on shared states. Hard replay replaces soft KL targets with one-hot generated tokens, while real replay uses stored prompt–answer pairs. Section 4.2 specifies the matched controls; Appendix B gives the algorithm and resource accounting.

### 3.3 Why Condition-Anchored Distillation Preserves Behavior

The common objective is useful because it measures function-space drift on states selected by old conditions. Its strongest interpretation applies when those states and kernels constitute the model's actual generation path. For a fixed condition $c$, suppose teacher and student use the same initial distribution $\pi(z_S \mid c)$ and reverse factorizations

$$P_\theta^{\mathrm{path}}(z_{0:S} \mid c) = \pi(z_S \mid c) \prod_{s=1}^{S} K_{\theta,s}(z_{s-1} \mid z_s, c). \tag{6}$$

Whenever the divergences are finite,

$$D_{\mathrm{KL}}\left(P_{\theta^-}^{\mathrm{path}} \| P_\theta^{\mathrm{path}}\right) = \sum_{s=1}^{S} \mathbb{E}_{z_s \sim P_{\theta^-}^{\mathrm{path}}} D_{\mathrm{KL}}\left(K_{\theta^-,s}(\cdot \mid z_s, c) \| K_{\theta,s}(\cdot \mid z_s, c)\right), \tag{7}$$

$$D_{\mathrm{TV}}(P_{\theta^-}(z_0 \mid c), P_\theta(z_0 \mid c)) \le \sqrt{\tfrac{1}{2} D_{\mathrm{KL}}\left(P_{\theta^-}^{\mathrm{path}} \| P_\theta^{\mathrm{path}}\right)}. \tag{8}$$

The first identity is the KL chain rule; the second follows from data processing and Pinsker's inequality. For AR generation, the same argument applies in the forward direction with prefixes as states and EOS included in the sequence. Consequently, matching next-token distributions under sampled teacher rollouts exactly decomposes sequence-level teacher–student KL. Moreover, for any bounded behavior score $h(y) \in [0, 1]$, the change in $\mathbb{E}[h(y) \mid c]$ is at most the terminal total variation in (8). This connects state matching to observable behavior rather than parameter distance.

**Finite anchors expose the coverage term.** Let $\Delta_\theta(c) = D_{\mathrm{TV}}(P_{\theta^-}(\cdot \mid c), P_\theta(\cdot \mid c))$. Suppose conditions have metric $d$, this drift is $L_\Delta$-Lipschitz, and every old condition is assigned to an anchor $a_{\nu(c)}$. With cell masses $\rho_j = \Pr[\nu(c) = j]$,

$$\mathbb{E}_{c \sim \mu_{\mathrm{old}}} \Delta_\theta(c) \le \sum_j \rho_j \Delta_\theta(a_j) + L_\Delta \, \mathbb{E}_{c \sim \mu_{\mathrm{old}}} d(c, a_{\nu(c)}). \tag{9}$$

The two terms are anchor drift and coverage error. If training weights $\widehat{\rho}_j$ differ from $\rho_j$, an additional $D_{\mathrm{TV}}(\rho, \widehat{\rho})$ term suffices because $0 \le \Delta_\theta \le 1$. Both anchor fit and coverage therefore matter.

**Generated support and teacher error remain.** For a fixed student, if the per-trajectory distillation cost is bounded by $B$, $m$ independent teacher trajectories estimate its expectation within $B\sqrt{\log(2/\delta)/(2m)}$ with probability at least $1 - \delta$. Deterministic greedy generations trade this sampling error for support bias. In addition, CAGD preserves the previous teacher rather than ground truth: for a bounded old-task score, student-to-ground-truth drift is no larger than the teacher's existing error plus teacher–student drift.

**What the masked-diffusion loss guarantees.** For a masked state and position drawn from the normalized state weighting in (5), Pinsker and Jensen give

$$\mathbb{E}\, D_{\mathrm{TV}}(p_{\theta^-}(x_i \mid x_t), p_\theta(x_i \mid x_t)) \le \sqrt{\tfrac{1}{2}\, \mathbb{E}\, D_{\mathrm{KL}}(p_{\theta^-}(x_i \mid x_t) \| p_\theta(x_i \mid x_t))}. \tag{10}$$

Thus, the implemented loss directly controls local denoising drift on teacher-generated support. Equations (7) and (8) additionally control terminal outputs only for a sampler whose distilled

conditional distributions are its transition kernels and whose state weighting matches its path visitation. We do not assume that correspondence for the present SMDM implementation. Appendix A gives the proofs and the exact specialization conditions.

## 4 EXPERIMENTS

Our analysis links teacher-state matching to predictive stability (Section 3.3), motivating tests of retention and generated behavior. We organize the experiments around four questions: (1) Can CAGD mitigate forgetting under changes in task order? (2) Do soft targets provide benefits beyond hard replay? (3) Do the gains extend to autoregressive models? (4) Do loss-based gains translate into improved generation behavior? **All current adaptation experiments use full-parameter fine-tuning of the student model.** Evaluation details are given in Appendix C.4. Figure 2 summarizes paired contrasts for Sections 4.1–4.3.

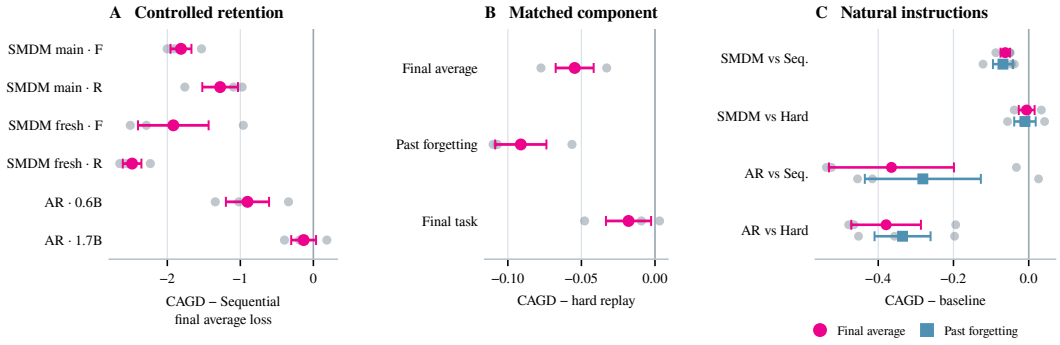


Figure 2: Paired evidence map. (A) CAGD minus Sequential on controlled SMDM streams and two AR checkpoints. (B) CAGD minus hard replay with identical generated support. (C) Within-backend natural-instruction contrasts. Markers show means ± SEM and gray points show seeds; negative loss differences favor CAGD.

### 4.1 RETENTION ON CONTROLLED MASKED-DIFFUSION STREAMS

To evaluate forgetting mitigation and sensitivity to task order, we adapt SMDM-219M (Nie et al., 2025) on four disjoint factual tasks, each with four facts, 120 training prompts, and 40 prompt-disjoint test prompts, derived from the reversal-association benchmark (Berglund et al., 2023). Forward and reverse streams contain the same tasks in opposite orders. Sequential receives no old-task signal; CAGD retains 64 balanced prompt-only anchors per old task. Fixed-$T_4$ loss evaluates the same task $\mathcal{D}$ after $\mathcal{A} \to \mathcal{B} \to \mathcal{C} \to \mathcal{D}$ and $\mathcal{D} \to \mathcal{C} \to \mathcal{B} \to \mathcal{A}$, comparing retention when that task is learned last versus first.

Table 1: Four-task masked-diffusion endpoints. Joint training is an order-free offline reference, all tasks are visible. Bold marks the better continual result within each order; the offline reference is excluded from this comparison. Fixed-task loss evaluates the same canonical $T_4$ after either order.

| Order | Method | Final avg. loss ↓ | Past forgetting ↓ | Fixed-$T_4$ loss ↓ | Final-task loss ↓ |
|---|---|---|---|---|---|
| Forward | Sequential | $2.927_{\pm 0.231}$ | $2.591_{\pm 0.429}$ | $0.439_{\pm 0.039}$ | $0.439_{\pm 0.039}$ |
| | **Ours** | $\mathbf{1.114}_{\pm 0.374}$ | $\mathbf{0.674}_{\pm 0.387}$ | $\mathbf{0.318}_{\pm 0.014}$ | $\mathbf{0.318}_{\pm 0.014}$ |
| Reverse | Sequential | $2.168_{\pm 0.390}$ | $1.997_{\pm 0.395}$ | $2.117_{\pm 0.419}$ | $1.163_{\pm 0.380}$ |
| | **Ours** | $\mathbf{0.891}_{\pm 0.215}$ | $\mathbf{0.505}_{\pm 0.106}$ | $\mathbf{0.409}_{\pm 0.086}$ | $\mathbf{0.738}_{\pm 0.314}$ |
| Order-free | Joint | $0.805_{\pm 0.193}$ | – | $0.332_{\pm 0.021}$ | $0.332_{\pm 0.021}$ |

CAGD lowers final average loss **from 2.927 to 1.114** forward and **from 2.168 to 0.891** in reverse, with lower mean final loss and forgetting (Table 1; Figure 2A). Fixed-$T_4$ loss remains 0.318 forward and 0.409 reverse, versus 0.439 and 2.117 with Sequential. Both orders show retention gains without increasing mean final-task loss. See Appendix D.2 for anchor-budget sensitivity.

### 4.2 Soft Targets versus Hard Generated Replay

To isolate target information from generated support, we run matched AR behavior interventions on Qwen3-0.6B and 1.7B. Each paired branch loads the same full-parameter GSM8K checkpoint, replays the same 64 teacher-generated trajectories while learning Dolly summarization, and differs only in one-hot versus teacher-distribution targets (Shin et al., 2017; Sun et al., 2020a;b).

Table 2: Matched AR target interventions across Qwen scales. Within each pair, both branches load the same GSM8K checkpoint and replay the same 64 teacher-generated trajectories; only one-hot versus soft teacher targets change. Values are mean ± SEM over three paired seeds. Boldface marks the better method mean within each model and endpoint; Δ is ours minus hard replay. Retention is the fraction of questions answered correctly at the shared checkpoint that remain correct after adaptation.

| Model | Method | Final EM (%) ↑ | Retention (%) ↑ | New-task loss ↓ | New-task acc. (%) ↑ |
|---|---|---|---|---|---|
| Qwen3-0.6B | Hard replay | $10.61_{\pm 1.25}$ | $26.57_{\pm 1.89}$ | $3.020_{\pm 0.059}$ | $67.61_{\pm 0.25}$ |
| | **Ours** | $\mathbf{13.98}_{\pm 0.93}$ | $\mathbf{37.95}_{\pm 2.98}$ | $\mathbf{2.936}_{\pm 0.053}$ | $\mathbf{68.67}_{\pm 0.21}$ |
| | Paired Δ | $+3.36_{\pm 0.47}$ | $+11.38_{\pm 1.24}$ | $-0.084_{\pm 0.105}$ | $+1.06_{\pm 0.46}$ |
| Qwen3-1.7B | Hard replay | $28.15_{\pm 1.95}$ | $47.43_{\pm 3.56}$ | $2.941_{\pm 0.117}$ | $\mathbf{71.68}_{\pm 0.12}$ |
| | **Ours** | $\mathbf{29.97}_{\pm 1.59}$ | $\mathbf{52.40}_{\pm 2.77}$ | $\mathbf{2.903}_{\pm 0.156}$ | $71.51_{\pm 0.04}$ |
| | Paired Δ | $+1.82_{\pm 3.34}$ | $+4.97_{\pm 6.21}$ | $-0.039_{\pm 0.120}$ | $-0.18_{\pm 0.09}$ |

At 0.6B, soft targets improve final GSM8K exact match by $3.36_{\pm 0.47}$ percentage points and conditional retention by $11.38_{\pm 1.24}$ points; at 1.7B, the corresponding mean differences are $+1.82_{\pm 3.34}$ and $+4.97_{\pm 6.21}$ points (Table 2). Mean new-task loss does not increase at either scale, while answer-token accuracy changes by $+1.06_{\pm 0.46}$ and $-0.18_{\pm 0.09}$ points, respectively. The analogous SMDM-219M and SMDM-1.14B checkpoints reach only $1.49\%$ and $1.62\%$ mean exact match immediately after GSM8K training, so their behavioral comparison remains at the task floor; complete three-seed four-model aggregates are in Appendix D.5.

A separate matched SMDM two-task study provides likelihood-level evidence: relative to hard replay, CAGD lowers final average loss by $\mathbf{0.055}_{\pm 0.013}$ and forgetting by $\mathbf{0.091}_{\pm 0.017}$ on identical generated completions (Appendix Table 9; Figure 2B). Longer streams use evolving teachers and supports and therefore also include optimization history (Appendix Table 10).

### 4.3 Task and Objective Scope

We vary factual content, natural-instruction tasks, and the modeling objective to evaluate transfer beyond the controlled diffusion stream.

**Fresh factual content.** We replace every main-stream association with three unseen two-fact tasks and run both orders. CAGD lowers final average loss **from $\mathbf{2.445}$ to $\mathbf{0.527}$** forward and **from $\mathbf{2.920}$ to $\mathbf{0.438}$** reverse, with lower forgetting in both orders (Figure 2A).

**Natural instructions.** The prompt-disjoint Dolly stream contains closed QA, summarization, and creative writing (Databricks, 2023). SMDM-219M and Qwen3-0.6B receive the same source rows and task order, with Sequential and hard generated replay as baselines.

Table 3: **Natural-instruction endpoints.** Values are comparable within Model.

| Model | Method | Final avg. loss ↓ | Past forgetting ↓ | Final-task loss ↓ | Accuracy ↑ |
|---|---|---|---|---|---|
| SMDM-219M | Sequential | $2.343_{\pm 0.024}$ | $0.006_{\pm 0.025}$ | $\mathbf{3.824}_{\pm 0.014}$ | $0.093_{\pm 0.001}$ |
| | Hard replay | $\underline{2.286}_{\pm 0.019}$ | $\underline{-0.052}_{\pm 0.034}$ | $\mathbf{3.824}_{\pm 0.043}$ | $\mathbf{0.104}_{\pm 0.002}$ |
| | **Ours** | $\mathbf{2.281}_{\pm 0.012}$ | $\mathbf{-0.062}_{\pm 0.006}$ | $3.842_{\pm 0.031}$ | $\underline{0.100}_{\pm 0.001}$ |
| Qwen3-0.6B | Sequential | $\underline{3.959}_{\pm 0.113}$ | $\underline{0.211}_{\pm 0.155}$ | $5.910_{\pm 0.094}$ | $0.542_{\pm 0.006}$ |
| | Hard replay | $3.973_{\pm 0.053}$ | $0.265_{\pm 0.069}$ | $\underline{5.803}_{\pm 0.038}$ | $\underline{0.562}_{\pm 0.001}$ |
| | **Ours** | $\mathbf{3.594}_{\pm 0.074}$ | $\mathbf{-0.070}_{\pm 0.056}$ | $\mathbf{5.586}_{\pm 0.072}$ | $\mathbf{0.564}_{\pm 0.002}$ |

CAGD lowers mean final average loss and forgetting over Sequential in both backends (Table 3; Figure 2C). On SMDM, loss falls from 2.343 to 2.281, and forgetting from 0.006 to −0.062. On Qwen, mean loss falls from 3.959 to 3.594, and forgetting from 0.211 to −0.070; soft targets also outperform hard replay on both mean metrics within each backend. Here, forgetting tracks closed QA and summarization after creative-writing adaptation, so the improvements concern retention of earlier natural-instruction tasks rather than only synthetic factual associations. These results extend the observed mean retention gains to natural-language tasks under both masked-diffusion and autoregressive training objectives.

**Autoregressive transfer.** We adapt Qwen3-0.6B and 1.7B (Yang et al., 2025), using teacher-prefix token KL as the AR specialization of (3). Both AR checkpoints use the same four-task factual stream as SMDM and retain 64 prompt-only anchors per old task. Paired final-loss reductions are $\mathbf{0.902}_{\pm 0.296}$ and $0.133_{\pm 0.170}$, respectively (Fig. 3). Mean final-task loss changes by $+0.017$ and $+0.008$. Together, these comparisons show average retention gains across new factual content, natural-instruction tasks, and both tested AR checkpoints.

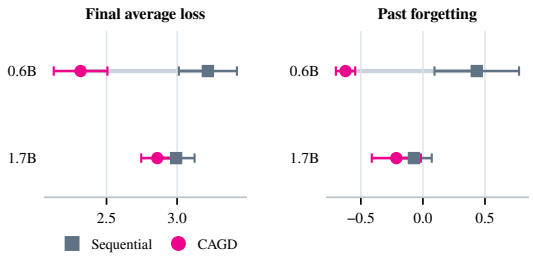


Figure 3: Autoregressive adaptation across Qwen3 checkpoints. Points show means ± SEM.

### 4.4 Behavioral Retention on GSM8K and TRACE

We test earlier-task retention on GSM8K (Cobbe et al., 2021) and heterogeneous continual adaptation on TRACE (Wang et al., 2023) to evaluate whether loss-based retention translates into generated behavior.

**GSM8K behavioral retention.** We place GSM8K before Dolly summarization and creative writing, using 1,000 updates per task and 64 prompt-only anchors per old task. Greedy final-answer exact match (EM) on all 1,319 test questions before and after later adaptation measures answer retention; format compliance separately measures preservation of the required answer delimiter.

Table 4: GSM8K endpoints for three current checkpoints, with the archived 4B control shown separately. Values are comparable only within a checkpoint and objective. Format compliance counts the required #### answer delimiter.

| Model | Method | After GSM8K EM (%) ↑ | Final EM (%) ↑ | Retention Δ (pp) ↑ | Format compliance (%) ↑ | Final loss ↓ |
|---|---|---|---|---|---|---|
| SMDM-219M | Sequential | $1.36_{\pm 0.16}$ | $1.49_{\pm 0.13}$ | $0.13_{\pm 0.07}$ | $0.00_{\pm 0.00}$ | $3.629_{\pm 0.022}$ |
| | **Ours** | $1.36_{\pm 0.16}$ | $\mathbf{1.62}_{\pm 0.22}$ | $\mathbf{0.25}_{\pm 0.20}$ | $\mathbf{0.30}_{\pm 0.19}$ | $\mathbf{3.625}_{\pm 0.054}$ |
| SMDM-1.14B | Sequential | $1.36_{\pm 0.27}$ | $\mathbf{1.90}_{\pm 0.12}$ | $\mathbf{0.53}_{\pm 0.38}$ | $0.00_{\pm 0.00}$ | $3.542_{\pm 0.042}$ |
| | **Ours** | $1.36_{\pm 0.27}$ | $1.42_{\pm 0.33}$ | $0.05_{\pm 0.40}$ | $\mathbf{0.56}_{\pm 0.48}$ | $\mathbf{3.447}_{\pm 0.051}$ |
| Qwen3-0.6B | Sequential | $26.33_{\pm 1.14}$ | $10.06_{\pm 0.70}$ | $-16.27_{\pm 1.26}$ | $0.40_{\pm 0.33}$ | $5.861_{\pm 0.061}$ |
| | **Ours** | $26.33_{\pm 1.14}$ | $\mathbf{10.84}_{\pm 0.27}$ | $\mathbf{-15.49}_{\pm 0.92}$ | $\mathbf{88.70}_{\pm 6.80}$ | $\mathbf{5.770}_{\pm 0.134}$ |
| Qwen3-4B | Sequential | $75.84_{\pm 0.94}$ | $8.39_{\pm 2.48}$ | $-67.45_{\pm 3.28}$ | $0.00_{\pm 0.00}$ | $4.609_{\pm 0.037}$ |
| | **Ours** | $75.84_{\pm 0.94}$ | $\mathbf{74.32}_{\pm 1.06}$ | $\mathbf{-1.52}_{\pm 1.31}$ | $\mathbf{98.43}_{\pm 0.09}$ | $\mathbf{4.595}_{\pm 0.007}$ |

CAGD raises delimiter compliance **from** 0.40% **to** 88.70% at 0.6B (Table 4). The paired final-EM difference is $+0.78_{\pm 0.84}$ percentage points. SMDM post-GSM8K EM is only 1.36% at either scale. **These results show answer-format retention on Qwen.** Three-seed aggregate endpoints and output examples are in Appendices D.5 and E. Unlike the fixed-support intervention in Section 4.2, this longer stream compares against Sequential after two later tasks and allows the intermediate teacher and generated support to evolve.

**Continual adaptation on TRACE.** TRACE tests eight heterogeneous tasks on Qwen3-4B-Instruct-2507. ACC averages final official task scores; BWT measures learned-to-final score changes on

earlier tasks. We compare CAGD with Sequential, stored Replay, Self-Distillation Fine-Tuning (SDFT) (Shenfeld et al., 2026), and two On-Policy Replay variants (Chen et al., 2026). In our TRACE adapter, SDFT samples completions from the student and minimizes token-level forward KL to an EMA teacher given the same query and its paired expert demonstration. For the On-Policy Replay variants, OPR-RU ranks generated answers on historical prompts using reference-based scores, whereas OPR-SC ranks them by mean token log-probability. All methods use the same task order, backbone, and optimizer schedule; replay-based methods retain 50 historical records in total. CAGD instead distills on fixed prompt anchors. Differences in historical-data access, support selection, and compute make this an **end-to-end comparison**; implementation details are in Appendix C.6.

Table 5: **TRACE comparison** on Qwen3-4B-Instruct-2507 in the canonical eight-task order. Task columns show the final official metric after Stage 8, averaged over training seeds 3407, 3408, and 3409; each seed averages eight decoding repetitions. ACC averages all eight final task scores, and BWT averages final-minus-learned scores over the first seven tasks. ACC and BWT show mean and sample standard deviation across seeds. Higher is better; bold and underline mark the best and second-best mean in each column.

| Model | Method | ACC ↑ | BWT ↑ | C-STANCE ↑ | FOMC ↑ | MeetingBank ↑ | Py150 ↑ |
|---|---|---|---|---|---|---|---|
| Qwen3-4B | Sequential | $59.71_{\pm 0.97}$ | $-7.16_{\pm 1.24}$ | 44.54 | 67.99 | 58.10 | 63.71 |
| | Replay | $62.61_{\pm 2.46}$ | $-4.14_{\pm 2.68}$ | <u>49.88</u> | <u>70.01</u> | 61.10 | 65.09 |
| | SDFT | $61.23_{\pm 0.85}$ | $\mathbf{-1.00}_{\pm 1.54}$ | 49.45 | 69.43 | 57.78 | 63.30 |
| | OPR-RU | <u>62.73</u>$_{\pm 0.54}$ | $-4.12_{\pm 0.32}$ | 46.15 | 69.24 | <u>62.71</u> | <u>65.80</u> |
| | OPR-SC | $60.67_{\pm 1.91}$ | $-6.35_{\pm 2.04}$ | 43.05 | 64.76 | 61.62 | 65.57 |
| | **Ours** | $\mathbf{64.47}_{\pm 1.75}$ | <u>$-1.79$</u>$_{\pm 1.32}$ | **50.15** | **72.91** | **64.45** | **66.26** |
| Qwen3-4B | **Method** | **ACC ↑** | **BWT ↑** | **ScienceQA ↑** | **NG-cm ↑** | **NG-ds ↑** | **20Minuten ↑** |
| | Sequential | $59.71_{\pm 0.97}$ | $-7.16_{\pm 1.24}$ | 71.11 | 56.64 | **74.64** | 40.93 |
| | Replay | $62.61_{\pm 2.46}$ | $-4.14_{\pm 2.68}$ | 79.91 | 61.47 | 72.67 | 40.78 |
| | SDFT | $61.23_{\pm 0.85}$ | $\mathbf{-1.00}_{\pm 1.54}$ | <u>83.80</u> | 59.38 | 72.19 | 40.35 |
| | OPR-RU | <u>62.73</u>$_{\pm 0.54}$ | $-4.12_{\pm 0.32}$ | 80.75 | <u>62.76</u> | <u>73.41</u> | **40.98** |
| | OPR-SC | $60.67_{\pm 1.91}$ | $-6.35_{\pm 2.04}$ | 77.83 | 59.88 | 71.87 | 40.81 |
| | **Ours** | $\mathbf{64.47}_{\pm 1.75}$ | <u>$-1.79$</u>$_{\pm 1.32}$ | **85.00** | **63.01** | 73.04 | <u>40.96</u> |

Across three training seeds, CAGD raises ACC from 62.73 to **64.47** and BWT from $-4.12$ to $\mathbf{-1.79}$ versus OPR-RU, leading on six of eight task scores (Table 5). Thus, generation-level benefits appear in the matched GSM8K target intervention and TRACE performance.

## 5 Limitations

For AR models, teacher-path token matching decomposes sequence divergence; for masked diffusion, our loss controls only local denoiser drift and does not by itself guarantee terminal samples. CAGD trades stored outputs for prompt anchors and boundary-time teacher generation, so memory still grows with the number of tasks and teacher–student inference remains an added cost. Evidence is limited to the tested model sizes and finite task streams; bounded-memory anchors, longer horizons, and explicit compute–retention tradeoffs remain open.

## 6 Conclusion

We introduced CAGD, which stores old conditions and uses a frozen previous model to reconstruct teacher states and soft targets at task boundaries. Across SMDM and Qwen, matched-support comparisons favor soft over hard replay; four-task SMDM final loss falls from 2.927 to 1.114 forward and from 2.168 to 0.891 in reverse. On identical generated support, Qwen3-0.6B gains $3.36_{\pm 0.47}$ exact-match points and $11.38_{\pm 1.24}$ conditional-retention points over hard replay; the corresponding 1.7B mean effects are positive but uncertain. On TRACE, CAGD obtains the best ACC ($64.47_{\pm 1.75}$), the second-best BWT ($-1.79_{\pm 1.32}$), and the leading mean on six of eight tasks. These results support condition-anchored functional preservation across the tested AR and masked-diffusion settings.

## AI USE STATEMENT

Generative AI tools assisted with reviewing proofs and code, orchestrating experiments, interpreting checked outputs, generating plots and tables, and drafting and editing the manuscript. The authors defined the research question, inspected raw outputs, verified all mathematical and empirical claims, and take responsibility for the content. Numerical claims were regenerated from checked summaries of hashed experiment artifacts.

## REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European conference on computer vision*, pp. 144–161. Springer, 2018.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2107.03006.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". *arXiv preprint arXiv:2309.12288*, 2023.

Jingfan Chen, Yuxi Wang, Pengfei Wang, Xiao Chen, Zhaoxiang Zhang, Zhen Lei, and Qing Li. DiffusePast: Diffusion-based generative replay for class incremental semantic segmentation. *arXiv preprint arXiv:2308.01127*, 2023.

Yan Chen, Taojie Zhu, Meng Zhang, Xin Chen, Jiaqi Huang, Dongyang Xu, and Yizhi Wang. On-policy replay for continual supervised fine-tuning. *arXiv preprint arXiv:2605.29495*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Databricks. Databricks dolly 15k. https://huggingface.co/datasets/databricks/databricks-dolly-15k, 2023. Dataset card and data repository.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3 (4):128–135, 1999.

Ruikang Gao et al. DDGR: Continual learning with deep diffusion-based generative replay. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

Annette Rios Gonzales, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. A new dataset and efficient baselines for document-level text simplification in german. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pp. 152–161, 2021.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Jiayi Wang, Xukai Wang, Jingang Zhou, Hongbo Zhao, Wenzhuo Liu, Shijie Ma, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Continual learning for generative ai: From llms to mllms and beyond. *arXiv preprint arXiv:2506.13045*, 2025. URL https://arxiv.org/abs/2506.13045.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16409–16423, 2023.

Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Y. Tiruneh, and Seungryul Baek. SDDGR: Stable diffusion-based deep generative replay for class incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Kim_SDDGR_Stable_Diffusion-based_Deep_Generative_Replay_for_Class_Incremental_Object_CVPR_2024_paper.pdf.

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.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629. Springer, 2016.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32819–32848. PMLR, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Sergi Masip, Pau Rodriguez, Tinne Tuytelaars, and Gido M. van de Ven. Continual learning of diffusion models with generative distillation. In *Proceedings of Machine Learning Research*, volume 274, 2025.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3505–3523, 2022.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WNvvwK0tut.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Veselin Raychev, Pavol Bielik, and Martin Vechev. Probabilistic model for code with decision trees. *ACM SIGPLAN Notices*, 51(10):731–747, 2016.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.07524.

Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6664–6679, 2023.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017.

James Smith et al. Continual customization of text-to-image diffusion with C-LoRA. *arXiv preprint arXiv:2304.06027*, 2023. URL https://openreview.net/forum?id=TZdEgwZ6f3.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: Language modeling for lifelong language learning. In *International Conference on Learning Representations*, 2020a.

Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. Distill and replay for continual language learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3569–3579, 2020b.

Gido M van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018. URL https://arxiv.org/abs/1809.10635.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. Citb: A benchmark for continual instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. C-STANCE: A large dataset for chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13369–13385, 2023.

Yingxiu Zhao, Yinhe Zheng, Zhiliang Tian, Chang Gao, Jian Sun, and Nevin L. Zhang. Prompt conditioned VAE: Enhancing generative replay for lifelong learning in task-oriented dialogue. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11153–11169, 2022.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey, 2024.

# A PROOFS

## A.1 PATHWISE DIVERGENCE DECOMPOSITION

For a fixed condition $c$, abbreviate the teacher and student path distributions by $P^-$ and $P$. Their common initial distribution cancels, so

$$\log \frac{P^-(z_{0:S} \mid c)}{P(z_{0:S} \mid c)} = \sum_{s=1}^{S} \log \frac{K_{\theta^-,s}(z_{s-1} \mid z_s, c)}{K_{\theta,s}(z_{s-1} \mid z_s, c)}.$$

Taking expectation under $P^-$ and conditioning each summand on $z_s$ gives (7). The same calculation holds for densities with respect to a common dominating measure. For an autoregressive sequence, reverse the indexing and let $z_s = (c, y_{<s})$; including EOS makes variable-length generation an ordinary product of next-token conditionals.

## A.2 OUTPUT-BEHAVIOR STABILITY BOUND

Marginalization from a complete path $z_{0:S}$ to its terminal output $z_0$ is a measurable map. The data-processing inequality therefore gives

$$D_{\mathrm{KL}}(P_{\theta^-}(z_0 \mid c) \| P_\theta(z_0 \mid c)) \le D_{\mathrm{KL}}(P^{\mathrm{path}}_{\theta^-} \| P^{\mathrm{path}}_{\theta}).$$

Pinsker's inequality yields (8). With total variation defined as $\sup_E |P^-(E) - P(E)|$, its variational characterization implies, for any $h$ taking values in $[0, 1]$,

$$|\mathbb{E}_{P^-} h - \mathbb{E}_P h| \le D_{\mathrm{TV}}(P^-, P).$$

For a bounded ground-truth loss, the triangle inequality additionally separates the teacher's pre-existing task error from teacher–student drift. Unbounded negative log likelihood requires a separate tail or minimum probability condition; we do not claim a distribution-free NLL bound.

## A.3 FINITE-ANCHOR COVERAGE BOUND

Let $\nu(c)$ map each old condition to an anchor and assume $|\Delta_\theta(c) - \Delta_\theta(c')| \le L_\Delta d(c, c')$. Pointwise,

$$\Delta_\theta(c) \le \Delta_\theta(a_{\nu(c)}) + L_\Delta d(c, a_{\nu(c)}).$$

Integrating over $\mu_{\mathrm{old}}$ proves (9). If the training distribution assigns anchor weights $\widehat{\rho}$ instead of cell masses $\rho$, then

$$\left| \sum_j (\rho_j - \widehat{\rho}_j) \Delta_\theta(a_j) \right| \le D_{\mathrm{TV}}(\rho, \widehat{\rho}),$$

because $\Delta_\theta(a_j) \in [0, 1]$. Finally, for a fixed $\theta$ and independent trajectory costs $U_k \in [0, B]$, Hoeffding's inequality gives

$$\Pr\left[ \left| \frac{1}{m} \sum_{k=1}^{m} U_k - \mathbb{E}U \right| > B\sqrt{\frac{\log(2/\delta)}{2m}} \right] \le \delta.$$

This last statement is pointwise in the student; a uniform post-training claim would require controlling the capacity or stability of the learned model.

## A.4 DIFFUSION-PROCESS EXTENSION

If a conditional diffusion sampler has common terminal noise law $\pi(z_S \mid c)$ and model-dependent reverse kernels $K_{\theta,s}(z_{s-1} \mid z_s, c)$, distilling every such kernel under teacher path visitation is exactly the right-hand side of (7). Equations (7)–(8) then apply without changing the argument for a continuous or discrete state space.

For the masked-language implementation, define a normalized distribution over tuples $(c, \widetilde{y}, t, M_t, i)$ using the nonnegative native weights in (5), conditional on $i \in M_t$. Pinsker holds at every tuple. Taking expectations and using concavity of the square root gives (10). This is a statement about the categorical clean-token predictor at sampled corrupted states. It becomes a terminal-output statement only if those predictors induce the sampler kernels and the normalized weighting equals their teacher path visitation; neither property is assumed by the current experiment.

# B Algorithm and Implementation

## B.1 Training Algorithm

Given an initial model, task stream, per-task anchor budget $m$, and distillation weight $\beta$, CAGD performs the following operations. TRACE uses a total replay budget instead (Appendix C.6).

**Algorithm 1** Condition-Anchored Generative Distillation for continual language-model adaptation.

**Require:** Initial parameters $\theta_0$; task stream $\mathcal{D}_{1:T}$; anchor budget $m$; distillation weight $\beta$; backend $b$
**Ensure:** Adapted parameters $\theta_T$ and prompt-only memory $\mathcal{A}_{\le T}$

$\mathcal{A}_{<1} \leftarrow \emptyset, \quad \theta \leftarrow \theta_0$ ▷ Initialize the model and empty prompt memory
**for** $t = 1, \dots, T$ **do**
  **if** $t > 1$ **then**
    $\theta^- \leftarrow \mathrm{StopGrad}(\theta)$ ▷ Freeze the previous-stage model as teacher
    $\widetilde{\mathcal{R}}_{<t} \leftarrow \{(c, \mathrm{Decode}_{\theta^-}(c)) : c \in \mathcal{A}_{<t}\}$ ▷ Generate one completion per old anchor and stage
  **end if**
  **for** $k = 1, \dots, N_t$ **do**
    $B_{\mathrm{new}} \sim \mathcal{D}_t$ ▷ Sample a current-task minibatch
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{new},t}(\theta; B_{\mathrm{new}})$ ▷ Optimize acquisition of the incoming task
    **if** $t > 1$ **then**
      $B_{\mathrm{old}} \sim_{\mathrm{balanced}} \widetilde{\mathcal{R}}_{<t}$ ▷ Sample generated replay uniformly across old tasks
      $\mathcal{L} \leftarrow \mathcal{L} + \beta \widehat{\mathcal{L}}^{b}_{\mathrm{CAGD}}(\theta; \theta^-, B_{\mathrm{old}})$ ▷ Apply backend-specific soft distillation
    **end if**
    $\theta \leftarrow \mathrm{OptimizerStep}(\theta, \nabla_\theta \mathcal{L})$ ▷ Update the student model
  **end for**
  $\mathcal{A}_t \leftarrow \mathrm{SelectConditions}(\mathcal{D}_t, m)$ ▷ Select current-task prompts as new anchors
  $\mathcal{A}_{<t+1} \leftarrow \mathcal{A}_{<t} \cup \mathcal{A}_t$ ▷ Persist conditions rather than old outputs
  Discard temporary $\theta^-$ and $\widetilde{\mathcal{R}}_{<t}$ when $t > 1$ ▷ Release all stage-local replay resources
**end for**
**return** $\theta_T, \mathcal{A}_{\le T}$

The implementation uses separate deterministic random streams for current examples and replay examples, so the matched two-task soft-versus-hard comparison changes only the target loss.

## B.2 Autoregressive Distillation

Prompts and teacher-generated answers are concatenated and padded in a batch. Teacher and student receive the same sequence; the KL is evaluated only at answer-token prediction positions. Teacher probabilities are detached and temperature-scaled, with the usual temperature-squared correction. The factual and natural Qwen experiments use temperature 1, greedy generation, a maximum of 32 new tokens on the factual stream or 96 on the natural stream, and a replay minibatch per update.

## B.3 Masked-Diffusion Language Distillation

SMDM prompts stay visible. For every replay example, the implementation draws one $t \sim \mathcal{U}(10^{-3}, 1)$ and independently masks each answer token with probability $t$. Teacher and student consume exactly the same corrupted tensor. Token KL is divided by $t$, summed over masked positions, and divided by answer length. Hard replay uses the same reduction with one-hot generated tokens; real replay uses the same reduction with stored true tokens.

## B.4 Memory and Compute Accounting

In the factual experiments $m = 64$, so stage 4 uses 192 old anchors and temporary completions. Both generated methods pay one decoding pass per anchor per stage. CAGD additionally keeps a frozen model during updates and performs its forward pass for every replay minibatch. Hard replay and CAGD share update counts but are not FLOP matched because their replay losses require different forward computation. Exact record counts, generated-record identities, and wall times are stored in each run artifact. TRACE uses a fixed 50-record budget and caches teacher logits before student updates, as described in Appendix C.6.

Table 6: Resources beyond current-task training for the per-task-budget implementation. At stage $t$, $m(t-1)$ denotes the balanced number of old records. Anchors persist; completions are temporary.

| Method | Permanent old records | Teacher during updates | Replay target |
|---|---|---|---|
| Sequential | 0 | ✗ | none |
| Hard replay | $m(t-1)$ conditions | ✗ | one-hot completion |
| Real replay | $m(t-1)$ condition–output pairs | ✗ | one-hot output |
| Joint | all old pairs | ✗ | one-hot output |
| **Ours** | $m(t-1)$ conditions | ✓ | teacher distribution |

# C Details

## C.1 Datasets and Task Construction

The controlled benchmark uses the official reversal-association files (Berglund et al., 2023). A task fixes one direction and a disjoint group of entities. Training and test templates come from separate JSONL files; exact prompts and complete encoded rows are recorded per task. The main stream uses four groups of four facts (indices 8–23), and the fresh stream uses three groups of two facts (indices 24–29). Exact reverse order contains the same canonical tasks in the opposite sequence, so evaluation masks can be keyed by task identity rather than encounter stage.

The natural-language extension is constructed from the CC BY-SA 3.0 Databricks Dolly 15K dataset (Databricks, 2023) at a fixed revision. It forms the stream closed-question answering $\rightarrow$ summarization $\rightarrow$ creative writing, with 120 training and 40 held-out examples per task. Prompts and source rows are disjoint across all splits and tasks. Selection is deterministic and independent of model outcomes. A row is retained only if the complete prompt–answer sequence fits within 256 tokens for both the SMDM and Qwen tokenizers and its prompt fits within 160 tokens. Both backends therefore receive the same 480 source rows and task order; no natural example is mixed into the factual experiments or hyperparameter selection.

## C.2 Training Settings

Table 7: Training and model settings.

| Setting | SMDM factual | Qwen factual | SMDM natural | Qwen natural |
|---|---|---|---|---|
| Checkpoint | Public SMDM-170M | Qwen3 0.6B / 1.7B | SMDM-219M | Qwen3-0.6B |
| Parameter count | 219M | 596M / 1.721B | 219M | 596M |
| Updates per task | 1,000 | 1,000 | 1,000 | 1,000 |
| Batch size | 4 | 4 | 2 | 2 |
| Evaluation batch size | 4 | 4 | 2 | 2 |
| Optimization seeds | 3407, 3408, 3409 | 3407, 3408, 3409 | 3407, 3408, 3409 | 3407, 3408, 3409 |
| Optimizer | AdamW | AdamW | AdamW | AdamW |
| Learning rate | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ |
| Gradient clipping | 1 | 1 | 1 | 1 |
| Weight decay | 0 | 0 | 0 | 0 |
| Anchors per old task | 64 | 64 | 64 | 64 |
| Replay weight | $\beta = 1$ | $\beta = 1$ | $\beta = 1$ | $\beta = 1$ |
| Distillation temperature | 1 | 1 | 1 | 1 |
| Maximum length | 128 | 128 | 256 | 256 |
| Generation limit | Replay length 52 | 32 new tokens | Replay length 256 | 96 new teacher tokens |
| Generation | 32 reverse steps; CFG 0.8; temperature 0 | Greedy | 32 reverse steps; CFG 0.8; temperature 0 | Greedy |
| Evaluation | 32 mask draws; $t \sim \mathcal{U}(10^{-3}, 1)$ | Causal next-token objective | 16 mask draws; $t \sim \mathcal{U}(10^{-3}, 1)$ | Causal next-token objective |
| Cross-backend reading | – | | Direction only; native losses and decoding objectives differ | |

## C.3 Hyperparameter Selection

The CAGD coefficient is fixed at $\beta = 1$ throughout. Component and natural-task settings were specified before execution; their protocols and validation checks are collected in Appendix F.

### C.4 Evaluation Protocols

Unless stated otherwise, results are means ± SEM over three paired training seeds (3407–3409), with paired differences computed within seed before aggregation. TRACE uses the same three training seeds, with each learned and final task score averaged over eight decoding repetitions. The anchor-budget sweep and qualitative runs use seed 3407. **Loss magnitudes are comparable only within the same checkpoint, objective, and task stream.**

Final average held-out loss measures stream-wide performance, past-task forgetting follows (2), and final-task loss measures new-task acquisition. SMDM factual losses average 32 corruption draws per test prompt; natural-task losses use 16. For Qwen, answer loss and teacher-forced answer-token accuracy use the causal next-token objective. Generated target containment is secondary; exact match is zero for the principal SMDM four-task methods.

Natural-stream evaluation uses prompt-disjoint held-out rows and reports final average answer loss, forgetting over the first two tasks, final-task loss, and secondary answer-token accuracy. Primary contrasts are CAGD minus Sequential and minus hard generated replay.

The matched masked-diffusion intervention fixes the Task-1 checkpoint, 64 anchors, teacher completions, corruption draws, replay minibatches, current-task stream, and loss weight; only teacher–student KL versus training on the sampled completion changes. The matched AR behavior intervention similarly gives both branches the same full-parameter GSM8K checkpoint, 64 prompts, saved teacher completions, Dolly summarization stream, minibatch RNGs, and update budget; only soft KL versus one-hot replay changes.

Within each model and setting, bold scores mark the best method; underlining marks the second-best only when at least three methods are compared.

### C.5 GSM8K Behavioral-Retention Protocol

**Protocol card.**

| Protocol item | Specification |
|---|---|
| Task stream | GSM8K → Dolly summarization → Dolly creative writing. |
| GSM8K data | 5,250 training examples and all 1,319 official test questions, with no exact train–test question overlap. |
| GSM8K input template | `Question: <question>`<br>`Answer:` |
| GSM8K target template | `<original rationale>`<br>`#### <answer>` |
| Dolly input template | `Instruction: <instruction>`<br>`[Context: <context>]`<br>`Response:`, where the context line is included only when nonempty. |
| Dolly target template | `<reference response>`. |
| Models | SMDM-219M and 1.14B; Qwen3-0.6B; archived Qwen3-4B control. |
| Sequential | No old-task signal. |
| **Ours** | 64 prompt-only anchors per previous task; completions from the frozen preceding checkpoint; teacher-distribution matching with $\beta = 1$. |
| Optimization | 1,000 AdamW updates per task; batch size 2; learning rate $5 \times 10^{-5}$; no weight decay; gradient clipping at 1. |
| Paired seeds | 3407, 3408, and 3409. |
| Teacher generation | At most 128 new tokens. SMDM uses 32 denoising steps with classifier-free guidance 0.8. |
| Evaluation decoding | At most 256 new tokens. SMDM uses 256 denoising steps with guidance 0.1, equal-prompt-length batches, and EOS truncation. |
| Answer extraction | Numeric value after `####`; otherwise the last numeric value in the generated completion. |
| Primary endpoint | Final GSM8K exact match after the complete task stream. |
| Additional endpoints | Exact match immediately after GSM8K, retention change, `####` format rate, and final creative-writing loss. |
| Statistics | 10,000-draw seed-by-example bootstrap for Qwen3-0.6B, with fixed seed 20260908. |

For SMDM, final creative-writing loss averages 16 independent corruption draws.

**Matched soft-target intervention.** The controlled interventions in Tables 2 and 11 use the shorter stream GSM8K → Dolly summarization on Qwen3-0.6B and 1.7B and on SMDM-219M and 1.14B. For each model and seed, one shared stage-0 process performs 1,000 full-parameter GSM8K updates, saves the checkpoint, evaluates all 1,319 test questions, and generates one completion from each of the first 64 training prompts. Qwen uses greedy decoding; SMDM uses 32 denoising steps and

guidance 0.8. Hard replay and CAGD then load byte-identical checkpoint and completion artifacts and run 1,000 updates on the same 120 summarization examples. Conditional retention is the fraction of questions correct at stage 0 that remain correct after adaptation. Final GSM8K exact match, conditional retention, and held-out summarization loss are primary endpoints; all contrasts are paired by seed.

**Qualitative selection.** The GSM8K examples use seed 3407 and the first three official test indices that are correct immediately after GSM8K, wrong after final Sequential adaptation, and correct after final CAGD adaptation; the reverse count is also reported. For the additional Qwen output tables, three summarization and three creative-writing IDs were fixed from their inputs before generation, then decoded greedily for at most 96 new tokens after the complete stream. These examples are descriptive and do not define a quantitative endpoint.

### C.6 TRACE Protocol

We use the TRACE 5,000-example-per-task release (Wang et al., 2023) in the order C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, and 20Minuten (Zhao et al., 2023; Shah et al., 2023; Hu et al., 2023; Raychev et al., 2016; Lu et al., 2022; Mishra et al., 2022; Gonzales et al., 2021). The released OPR scorers use first-character accuracy for C-STANCE, FOMC, and ScienceQA; ROUGE-L F1 (Lin, 2004) for MeetingBank; fuzzy edit similarity for Py150; numeric exact match for the two NumGLUE tasks; and SARI (Xu et al., 2016) for 20Minuten. Scores are on a 0–100 scale and higher is better. ACC averages final task scores with equal weights; it is not classification accuracy. BWT averages final-minus-learned scores across the seven earlier tasks.

Both methods adapt Qwen3-4B-Instruct-2507 with AdamW, learning rate $10^{-5}$, cosine decay, global batch size 128, and task epochs $(5, 3, 7, 5, 3, 5, 5, 7)$. They use the non-thinking chat template and assistant-only loss; examples exceeding 2,048 tokens are excluded. Evaluation uses the eligible official test examples and temperature 0.1. For each of training seeds 3407–3409, every learned and final task score averages eight decoding repetitions.

The replay budget is 50 records *in total*, allocated as evenly as possible across old tasks, rather than 64 per task as in the other studies. OPR-RU generates candidates on all eligible historical training prompts, scores them against references, and trains on the selected prompt–response pairs. CAGD uses a fixed per-task prompt ordering, reduced to each stage's allocation, and greedy teacher completions of at most 512 tokens. Frozen teacher logits are cached once and reused for answer-token KL with $\beta = 1$ and temperature 1. The cache and completions are stage-local.

## D Additional Results

### D.1 Complete Four-Task Results

The largest sequential drift occurs on the earliest task in the forward order and on the second encountered task in reverse (Fig. 4). CAGD lowers all positive mean old-task drifts in both orders.

### D.2 Condition-Anchor Budget Sensitivity

The sweep changes support coverage without increasing the replay-update budget. Consequently, a larger support also reduces how often each anchor is revisited. The main setting gives the lowest final average loss in both orders: $0.894$ forward and $0.463$ reverse. The smallest setting slightly reduces forgetting relative to $m = 64$ ($0.228$ versus $0.257$ forward and $0.291$ versus $0.293$ reverse), but its reverse final-task loss rises from $0.163$ to $0.276$. Using all 120 prompts is not uniformly beneficial: final average loss and forgetting reach $1.432/1.096$ forward and $0.759/0.647$ reverse. **Broader coverage does not guarantee better retention under fixed compute.** This trade-off qualitatively echoes coverage and fitting error in Eq. (9); the sweep does not estimate either bound term.

### D.3 Fresh-Fact Results

CAGD has the lowest mean loss and forgetting in both fresh-fact orders (Table 8).

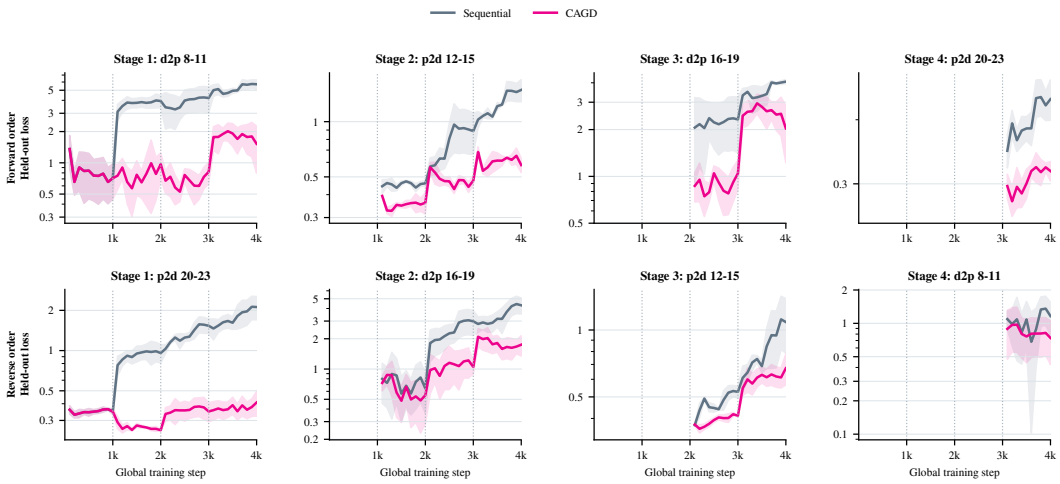


Figure 4: Four-task SMDM-219M held-out loss throughout continual adaptation (mean ± SEM over three paired seeds; lower is better). Every 100 optimizer steps, each learned task is evaluated on the same 40 prompt-disjoint test prompts and 32 fixed Monte Carlo mask draws used for the endpoint results. Dotted vertical lines mark task boundaries; panel vertical scales differ and are logarithmic.

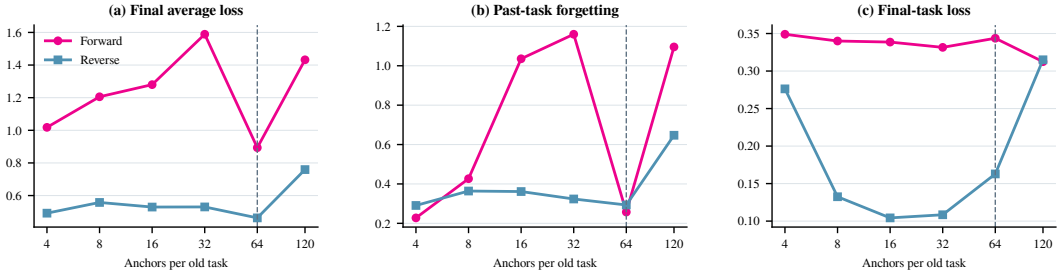


Figure 5: Anchor-budget sensitivity on the four-task SMDM-219M stream (seed 3407; lower is better). Each old task contributes $m \in \{4, 8, 16, 32, 64, 120\}$ prompts, balanced over its four facts; the supports are nested, and $m = 120$ contains all training prompts. Both orders keep 1,000 updates and four replay draws per update fixed. The dashed line marks the main-paper setting, $m = 64$.

### D.4 Component Isolation

Table 9 reports the aggregate matched-study endpoints; Table 10 gives the four-task end-to-end comparison. **The retention–acquisition trade-off varies by replay method:** stored real replay has the lowest four-task forgetting in both orders, while CAGD has the lowest final average loss.

### D.5 GSM8K Behavioral Retention

The complete three-seed aggregates are reported in Table 11. Both Qwen scales have positive mean exact-match and conditional-retention differences. The SMDM stage-0 exact matches remain near $1.5\%$, making their conditional estimates low-count and not establishing the same behavioral advantage. These fixed-support comparisons isolate target information at one task boundary; the longer Sequential-versus-CAGD stream in which teachers and generated support can evolve is summarized by the three-seed aggregates in Table 4.

The 10,000-draw paired seed-by-example bootstrap (seed 20260908) gives a 95% interval of $[-1.01, 2.50]$ percentage points for the 0.6B final-EM contrast; it does not establish a consistent correctness gain. On the final creative-writing task, Sequential and CAGD obtain answer-token accuracies $0.337_{\pm 0.003}$ and $0.342_{\pm 0.001}$, respectively. The archived 4B three-seed mean contrast remains $+65.93_{\pm 2.19}$ points; it is not evidence that this gain persists under full-parameter adaptation.

Table 8: Fresh-fact endpoints and end-to-end replay comparisons (mean ± SEM over three paired seeds). The stream has three tasks of two facts and is not magnitude matched to the main four-task stream. Generated-replay methods use their own evolving teachers and completions after divergence. Fixed-task loss evaluates the same canonical $T_3$ after either order.

| Order | Method | Final avg. loss ↓ | Past forgetting ↓ | Fixed-$T_3$ loss ↓ | Final-task loss ↓ |
|---|---|---|---|---|---|
| Forward | Sequential | $2.445_{\pm 0.524}$ | $2.678_{\pm 0.783}$ | $\underline{0.985}_{\pm 0.105}$ | $\underline{0.985}_{\pm 0.105}$ |
| | Hard replay | $\underline{0.719}_{\pm 0.192}$ | $\underline{0.075}_{\pm 0.101}$ | $1.136_{\pm 0.316}$ | $1.136_{\pm 0.316}$ |
| | Stored real replay | $1.090_{\pm 0.510}$ | $0.297_{\pm 0.315}$ | $1.627_{\pm 0.999}$ | $1.627_{\pm 0.999}$ |
| | **Ours** | $\mathbf{0.527}_{\pm 0.132}$ | $\mathbf{0.004}_{\pm 0.073}$ | $\mathbf{0.703}_{\pm 0.268}$ | $\mathbf{0.703}_{\pm 0.268}$ |
| Reverse | Sequential | $2.920_{\pm 0.220}$ | $3.413_{\pm 0.302}$ | $7.031_{\pm 0.444}$ | $1.025_{\pm 0.445}$ |
| | Hard replay | $\underline{0.710}_{\pm 0.147}$ | $\underline{0.294}_{\pm 0.113}$ | $\underline{1.040}_{\pm 0.272}$ | $\underline{0.747}_{\pm 0.236}$ |
| | Stored real replay | $0.913_{\pm 0.411}$ | $0.312_{\pm 0.244}$ | $1.146_{\pm 0.786}$ | $1.180_{\pm 0.410}$ |
| | **Ours** | $\mathbf{0.438}_{\pm 0.093}$ | $\mathbf{0.116}_{\pm 0.271}$ | $\mathbf{0.603}_{\pm 0.199}$ | $\mathbf{0.317}_{\pm 0.226}$ |

Table 9: Matched two-task component isolation. **Ours** and hard replay use identical teacher-generated completion records within each seed.

| Method | Final avg. loss ↓ | Past forgetting ↓ | Final-task loss ↓ |
|---|---|---|---|
| Hard replay | $\underline{0.291}_{\pm 0.025}$ | $0.085_{\pm 0.035}$ | $\underline{0.332}_{\pm 0.015}$ |
| Stored real replay | $0.336_{\pm 0.066}$ | $\underline{0.078}_{\pm 0.064}$ | $0.429_{\pm 0.010}$ |
| **Ours** | $\mathbf{0.237}_{\pm 0.031}$ | $\mathbf{-0.006}_{\pm 0.053}$ | $\mathbf{0.314}_{\pm 0.014}$ |

Table 10: End-to-end four-task replay comparison (mean ± SEM over three seeds). Unlike the two-task isolation, methods use their own evolving teacher and generated completions after they diverge. Fixed-task loss evaluates the same canonical $T_4$ after either order.

| Order | Method | Final avg. loss ↓ | Past forgetting ↓ | Fixed-$T_4$ loss ↓ | Final-task loss ↓ |
|---|---|---|---|---|---|
| Forward | Hard replay | $1.321_{\pm 0.298}$ | $0.887_{\pm 0.264}$ | $\mathbf{0.304}_{\pm 0.015}$ | $\mathbf{0.304}_{\pm 0.015}$ |
| | Stored real replay | $\underline{1.138}_{\pm 0.139}$ | $\mathbf{0.379}_{\pm 0.023}$ | $0.405_{\pm 0.032}$ | $0.405_{\pm 0.032}$ |
| | **Ours** | $\mathbf{1.114}_{\pm 0.374}$ | $\underline{0.674}_{\pm 0.387}$ | $\underline{0.318}_{\pm 0.014}$ | $\underline{0.318}_{\pm 0.014}$ |
| Reverse | Hard replay | $\underline{0.906}_{\pm 0.344}$ | $\underline{0.255}_{\pm 0.172}$ | $0.443_{\pm 0.056}$ | $\underline{1.395}_{\pm 0.751}$ |
| | Stored real replay | $1.102_{\pm 0.264}$ | $\mathbf{0.070}_{\pm 0.045}$ | $\mathbf{0.372}_{\pm 0.030}$ | $2.125_{\pm 0.516}$ |
| | **Ours** | $\mathbf{0.891}_{\pm 0.215}$ | $0.505_{\pm 0.106}$ | $\underline{0.409}_{\pm 0.086}$ | $\mathbf{0.738}_{\pm 0.314}$ |

Table 11: Complete aggregate matched-target results. "Learned EM" is measured at the shared post-GSM8K checkpoint, before Dolly adaptation; retention is the fraction of those correctly answered questions that remain correct. Values are mean ± SEM over three paired seeds. Boldface marks the better method mean within each model and post-adaptation endpoint; learned EM is identical by construction.

| Model | Method | Learned EM (%) | Final EM (%) ↑ | Retention (%) ↑ | New loss ↓ | New acc. (%) ↑ |
|---|---|---|---|---|---|---|
| SMDM-219M | Hard replay | $1.49_{\pm 0.21}$ | $\mathbf{1.90}_{\pm 0.12}$ | $28.03_{\pm 4.65}$ | $1.594_{\pm 0.014}$ | $\mathbf{12.82}_{\pm 1.34}$ |
| | **Ours** | $1.49_{\pm 0.21}$ | $1.39_{\pm 0.18}$ | $\mathbf{35.09}_{\pm 3.73}$ | $\mathbf{1.554}_{\pm 0.021}$ | $12.79_{\pm 0.88}$ |
| SMDM-1.14B | Hard replay | $1.62_{\pm 0.05}$ | $\mathbf{1.92}_{\pm 0.11}$ | $\mathbf{20.45}_{\pm 8.19}$ | $1.410_{\pm 0.029}$ | $12.45_{\pm 0.74}$ |
| | **Ours** | $1.62_{\pm 0.05}$ | $1.57_{\pm 0.39}$ | $12.27_{\pm 9.88}$ | $\mathbf{1.366}_{\pm 0.014}$ | $\mathbf{13.53}_{\pm 1.04}$ |
| Qwen3-0.6B | Hard replay | $26.33_{\pm 1.14}$ | $10.61_{\pm 1.25}$ | $26.57_{\pm 1.89}$ | $3.020_{\pm 0.059}$ | $67.61_{\pm 0.25}$ |
| | **Ours** | $26.33_{\pm 1.14}$ | $\mathbf{13.98}_{\pm 0.93}$ | $\mathbf{37.95}_{\pm 2.98}$ | $\mathbf{2.936}_{\pm 0.053}$ | $\mathbf{68.67}_{\pm 0.21}$ |
| Qwen3-1.7B | Hard replay | $46.20_{\pm 0.59}$ | $28.15_{\pm 1.95}$ | $47.43_{\pm 3.56}$ | $2.941_{\pm 0.117}$ | $\mathbf{71.68}_{\pm 0.12}$ |
| | **Ours** | $46.20_{\pm 0.59}$ | $\mathbf{29.97}_{\pm 1.59}$ | $\mathbf{52.40}_{\pm 2.77}$ | $\mathbf{2.903}_{\pm 0.156}$ | $71.51_{\pm 0.04}$ |

# E QUALITATIVE OUTPUT COMPARISONS

**Annotation guide.** Outputs are reproduced verbatim from the current runs. Green assessment marks denote correct or useful behavior, red marks a substantive error, and amber marks a qualification. Answer correctness, reasoning quality, and format compliance are assessed separately. GSM8K examples are selected by the fixed asymmetric rule, not a representative sample of overall gains; the corresponding reverse counts are reported in Appendix D.5.

**Table 12:** GSM8K output comparisons. Model: Qwen3-0.6B; all parameters adapted; seed 3407. The first three official test indices satisfying the predeclared selection rule are shown. The original target rationale is elided.

**Example 1: GSM8K test index 3**

**Question:** James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?
**Answer:**

**Target template:** `<original rationale>`
`#### 540`

**Sequential final output**

Each sprint is 60 meters so he runs 3 times a week so that's 180 meters He also runs 3 times a week so that's another 180 meters He runs 60 meters each sprint and he does 3 sprints so that's 180 meters total

✗ **Assessment:** returns 180 rather than 540 and omits the answer delimiter.

**Ours final output**

He runs 3*60=180 meters a week So he runs 180*3=540 meters a week #### 540

✓ **Assessment:** returns the correct 540 with the delimiter. ! The intermediate 180 meters are per session, not per week.

**Example 2: GSM8K test index 16**

**Question:** Two trains leave San Rafael at the same time. They begin traveling westward, both traveling for 80 miles. The next day, they travel northwards, covering 150 miles. What's the distance covered by each train in the two days?
**Answer:**

**Target template:** `<original rationale>`
`#### 230`

**Sequential final output**

First find the total distance the two trains travel westward: 80 miles * 2 = 160 miles Then find the total distance the two trains travel northwards: 150 miles * 2 = 300 miles Then add the distance each train travels west and the distance each train travels north to find the total distance each train travels: 160 miles + 300 miles = 460 miles

✗ **Assessment:** counts both trains rather than each train, returning 460 instead of 230.

**Ours final output**

The first day, the two trains travel 80 miles. The second day, they travel 150 miles. The total distance covered by the two trains is 80+150=230 miles. #### 230

✓ **Assessment:** returns the correct 230 with the delimiter. ! The explanation calls this the distance of both trains rather than of each train.

**Example 3: GSM8K test index 32**

**Question:** John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
**Answer:**

**Target template:** `<original rationale>`
`#### 35`

**Sequential final output**

He has 1000000 business hours in a week. He will spend 500000 hours a week taking care of the dogs.

✗ **Assessment:** invents business-hour counts and does not compute the required 35 hours.

**Ours final output**

He has 10*0.5=5 hours a week So he spends 5*7=35 hours a week #### 35

✓ **Assessment:** returns the correct 35 with the delimiter. ! The intermediate five hours are per day, not per week.

**Table 13:** GSM8K output comparisons. Model: SMDM-219M; all parameters adapted; seed 3407. The first three official test indices satisfying the predeclared selection rule are shown. The original target rationale is elided.

**Example 1: GSM8K test index 378**

**Question:** Lindsay is doing the laundry, and thinks she has missed some socks. There are 50 socks that need washing. If she washes 10 pairs of socks and 15 loose socks, how many socks has Lindsay missed?
**Answer:**

**Target template:** `<original rationale>`
`#### 15`

**Sequential final output**

Lindsay has been washing her socks, and has a 50 pair of loose socks. Lindsay has been washing her socks, and she washes 10 pairs of socks. While Lindsay was washing her socks, Lindsay was washing a 15 pair of loose socks. Lindsay says she was missing 2 loose socks, and she was missing a 50 pair of loose socks. Lindsay says she was missing a loose sock, and that she was missing 2 loose socks. Lindsay says she was missing a loose sock, and that it was a loose sock. Lindsay says she was missing a loose sock, and that she was missing a loose sock. Lindsay says she was missing a loose sock, and that she was missing a loose sock. Lindsay says she was missing a loose sock, and that she was missing a loose sock. Lindsay says she was missing a loose sock, and that she was missing a loose sock. Lindsay says that she was missing a loose sock, and that they were loose socks. Lindsay

✗ **Assessment:** Parsed answer: 2; incorrect. Repetition does not establish a solution.

**Ours final output**

Lindsay has been doing the laundry, and has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been washing 10 pairs of socks and 15 loose socks. Lindsay has been

! **Assessment:** Parsed answer: 15; correct under the last-number fallback. The repetitive response does not establish valid reasoning or compliant answer format.

**Example 2: GSM8K test index 569**

**Question:** Liam and Mitchell own competing lemonade stands across the street from one another. When Liam bragged that he had made $63 one weekend, Mitchell laughed and told Liam he had sold 21 lemonades at $4 apiece the same weekend. How many more dollars did Mitchell make selling lemonade that weekend than Liam?
**Answer:**

**Target template:** `<original rationale>`
`#### 21`

**Sequential final output**

Liam made 21 lemonades that weekend. Mitchell made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam also made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made 4 lemonades that weekend. Liam made

✗ **Assessment:** Parsed answer: 4; incorrect. Repetition does not establish a solution.

**Ours final output**

Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades at $4 apiece that weekend. Liam made 21 lemonades

! **Assessment:** Parsed answer: 21; correct under the last-number fallback. The repetitive response does not establish valid reasoning or compliant answer format.

**Example 3: GSM8K test index 588**

**Question:** Jared is trying to increase his typing speed. He starts with 47 words per minute (WPM). After some lessons the next time he tests his typing speed it has increased to 52 WPM. If he continues to increase his typing speed once more by 5 words, what will be the average of the three measurements?
**Answer:**

**Target template:** `<original rationale>`
`#### 52`

**Sequential final output**

The first way he will continue to increase his typing speed. The second one will be the first one. The first one will be the second one. The second one will be the third one. The third one will be the fourth one. The fourth one will be the fifth one. The fifth one will be the sixth one. The sixth one will be the sixth one. The sixth one will be the sixth one. The six

**Ours final output**

Dhollywood is the Gujarati cinema, also known as Dhollywood. It is one of the major regional and vernacular film industries of the cinema of India, having produced more than one thousand films since its inception.

✓ **Assessment:** identifies Gujarati cinema and remains grounded in the context, although the summary is verbose.

**Example 2: summarization:test:2**

**Instruction:** Please give me the list of airports that are owned by Australia Pacific Airports Corporation.
**Context:** Australia Pacific Airports Corporation Limited (APAC) is an unlisted company and owner of two Australian airports: Melbourne Airport and Launceston Airport. Each airport is operated by an airport lessee company, in which APAC has a controlling interest: Australia Pacific Airports (Melbourne) Pty Ltd and Australia Pacific Airports (Launceston) Pty Ltd respectively.
**Response:**

**Target template:** Australia Pacific Airports Corporation Limited owns two airports in Australia called Launceston Airport and Melbourne Airport.

**Sequential final output**

Australia Pacific Airports Corporation Limited (APAC) is an unlisted company and owner of two Australian airports: Melbourne Airport and Launceston Airport. Each airport is operated by an airport lessee company, in which APAC has a controlling interest: Australia Pacific Airports (Melbourne) Pty Ltd and Australia Pacific Airports (Launceston) Pty Ltd respectively.

✓ **Assessment:** names both requested airports; the remaining context is copied rather than summarized.

**Ours final output**

Australia Pacific Airports Corporation Limited is an unlisted company and owner of two Australian airports: Melbourne Airport and Launceston Airport. Each airport is operated by an airport lessee company, in which APAC has a controlling interest: Australia Pacific Airports (Melbourne) Pty Ltd and Australia Pacific Airports (Launceston) Pty Ltd respectively.

✓ **Assessment:** names both requested airports; like Sequential, it largely repeats the context.

**Example 3: summarization:test:33**

**Instruction:** How many people attended the Summer Breeze Open Air?
**Context:** Summer Breeze Open Air is an annual German heavy metal music festival. It was first held in 1997. The festival had been held in Abtsgmünd until 2006 when it was moved to its new location of Dinkelsbühl, Bavaria. The festival draws around 40,000 attendees annually.
**Response:**

**Target template:** The German heavy metal music festival Summer Breeze Open Air drew 40,000 attendees annually.

**Sequential final output**

The Summer Breeze Open Air has a capacity of 40,000 attendees annually. It is hosted by the Bavaria Ensemble, a collective of Bavarian heavy metal star players. The festival includes various concertos, street performances by the Bavarian pop starletes, and is known for its architecture.

! **Assessment:** includes 40,000 but fabricates organizers and activities absent from the context.

**Ours final output**

The Summer Breeze Open Air is an annual German heavy metal music festival. It was first held in 1997. The festival had been held in Abtsgmünd until 2006 when it was moved to its new location of Dinkelsbühl, Bavaria. The festival draws around 40,000 attendees annually.

✓ **Assessment:** preserves the 40,000 attendance figure without the unsupported embellishments.

**Table 15:** Unedited creative-writing outputs from a separately paired Qwen3-0.6B qualitative run (seed 3407), after closed QA → summarization → creative writing. Input-selected IDs are unchanged.

**Example 1: creative_writing:test:0**

**Instruction:** Write a dialogue between two hikers, John and Jane, who are lost in the woods and can't decide what to do. End the argument with Jane winning and them deciding to turn around and retrace their steps.
**Response:**

**Target template:** John: I think we're lost, this trail doesn't look familiar
Jane: I think you're right. We must have taken a wrong turn at the fork a mile back. Let's turn around and trace our steps back
John: No, I think we should just keep going forward. We are still on a path, so we should just go and see if we can find our way back to a path we know
Jane: I'm worried its going to get dark, and we'll just be going further into the woods. We should turn around and repeat our steps
John: Okay, you're right. Let's go back to where we first got lost and take the other side of the fork.

**Sequential final output**

John: I am lost, but here's the trail ahead of us. We've got to get as far as we can before we run out of supplies. After we figure out what to do next, we'll come back to talk it over. Before we do that, we've got to make sure we don't run out of supplies. After we do that, we'll decide to stop here and rest for the day. We'll look for any clues we can find before

✗ **Assessment:** only John speaks; the requested argument, Jane winning, and decision to retrace their steps are absent.

**Ours final output**

Hi, I am lost in this forest. Can you please share your views with me?

✗ **Assessment:** produces a generic request rather than the specified two-person dialogue and ending.

**Example 2: creative_writing:test:6**

**Instruction:** Write a poem about George Washington.
**Response:**

**Target template:** George Washington was a British man
The British had laws he couldn't understand
Washington and his friends put together a plan
That the people would resist and America could stand

**Sequential final output**

George Washington, Washington, George Washington, Washington, I love this, I think, I will be so born, and be so kind, and help me to do so, and make me happy, I will surely do so, and be so so, and I will be so so, and I will be so so, and I will be so so, and I will be so so, and I will be so so, and I will be so so, and I will

✗ **Assessment:** degenerates into repetition rather than a coherent poem.

**Ours final output**

George Washington, an American and a born in 1799, was the first person to achieve the feat of flying. He passed away in 1848, but his legacy still lives on many lands around the world.

He inspired the next generation of Americans with his wisdom and example.
He challenged the world with his courage and honesty.
He made history with his actions, not just as an American but as an international figure.
After George Washington's death

✗ **Assessment:** invents dates and credits Washington with flying; the text is neither historically grounded nor a coherent poem.

**Example 3: creative_writing:test:13**

**Instruction:** Write me a thank you note for my high-school teacher
**Response:**

**Target template:** Thank you for being a supportive teacher and encouraging me to keep learning. I will always remember your encouragement as I move forward in my studies.

**Sequential final output**

Thank you for being such a huge part of my success. You made all the difference in my high school days and still have a loyal fanbase across the world. I would love to reconnect with my high-school teacher and share more testimonials.

! **Assessment:** expresses thanks, but adds unsupported claims about a worldwide fanbase and testimonials.

**Ours final output**

Thank you for the physics textbook you provided. It's a huge burden for you to put all the math and physics questions in order. I hope you find it easier to manage than I do. Let me know if you think so.

! **Assessment:** thanks the teacher, but invents a physics-textbook context and drifts toward an unrelated request.

# F Reproducibility and Audit

Table 16 lists the principal entry points and artifacts needed to reproduce the reported results. The current snapshot is the completed run `runs/reproduction/fullparam_smdm_qwen06_qwen17_20260912_v2`: 11 experiment groups and 182 result cells, plus six shared SMDM GSM8K initial-stage records. The matched Qwen behavior intervention adds six branches and three shared checkpoints under `runs/qwen_gsm8k_soft_targets/formal_v1`; the scale extension adds 18 branches and nine shared checkpoints under `runs/gsm8k_soft_targets_scale`. Seeds are 3407–3409 except the seed-3407 anchor sweep and qualitative runs. The preliminary SMDM two-task confirmation, Qwen3-4B controls, and TRACE comparison retain their archived records and are not represented as newly rerun results.

**Table 16:** Principal files in the reproducibility release.

```
./ ..... ▷ Release root.
├── reproduction/ ..... ▷ Self-contained training and evaluation entry points.
│   ├── run_all_paper_experiments.sh ..... ▷ Run all non-TRACE experiments with configurable
│   │   checkpoints, seeds, and trainable scope.
│   ├── run_trace_experiment.sh ..... ▷ Train and evaluate the canonical eight-task suite; Table 5.
│   ├── table_cagd_main.py ..... ▷ Train/test factual SMDM; Table 1 and Fig. 2.
│   ├── table_cagd_components.py ..... ▷ Matched support; Table 9.
│   ├── table_cagd_deployed_components.py ..... ▷ End-to-end replay; Table 10.
│   ├── table_cagd_fresh.py ..... ▷ Fresh facts in both orders; Table 8.
│   ├── table_cagd_natural.py ..... ▷ Train/test natural-task SMDM and Qwen; Table 3 and Fig. 2.
│   ├── figure_qwen_cagd.py ..... ▷ AR factual checkpoint comparisons; Fig. 3.
│   ├── figure_loss_dynamics.py ..... ▷ Loss trajectories; Fig. 4.
│   ├── figure_anchor_budget.py ..... ▷ Coverage-budget sensitivity; Fig. 5.
│   ├── table_gsm8k_behavior.py ..... ▷ Train/test both GSM8K backends; Table 4.
│   ├── table_gsm8k_soft_targets.py ..... ▷ Matched behavior interventions; Tables 2 and 11.
│   └── tables_qualitative.py ..... ▷ Reproduce Qwen examples; Tables 12, 14 and 15.
├── report/ ..... ▷ Frozen protocols and run configurations.
└── runs/ ..... ▷ Validated records behind the cited tables and figures.
```

Only complete, configuration-matching cells enter the current summaries; failed and incomplete attempts are excluded and preserved separately. The preview's `make_current_figures.py` renders its four quantitative figures directly from these records; `figures/figure_data.json` records their source cells and scope.