Memory Dial: A Training Framework for Controllable Memorization in Language Models

Xiangbo Zhang[*]
Georgia Institute of Technology
Emory University
xiangbo.zhang@gatech.edu

Ali Emami
Emory University
ali.emami@emory.edu

Abstract

Memorization in language models is widely studied but remains difficult to isolate and control. Understanding when and what models memorize is essential for explaining their predictions, yet existing approaches are post-hoc: they can detect memorization in trained models, but cannot disentangle its effects from architecture, data, or optimization. We introduce **MEMORY DIAL**, a training framework that makes *memorization pressure* an explicit, controllable variable. MEMORY DIAL interpolates between standard cross-entropy and a temperature-sharpened objective via a single parameter α, producing a family of models identical in architecture and training setup (within each sweep), differing only in memorization pressure. Experiments across six architectures and five benchmarks demonstrate that: (1) α reliably controls memorization pressure, with seen-example accuracy increasing monotonically while unseen accuracy remains stable; (2) larger models are more responsive to memorization pressure; and (3) frequent sequences are easier to memorize than rare ones. Additional analyses show that the effect is robust across a range of sharpening temperatures, differs qualitatively from single-temperature cross-entropy, transfers to multilingual settings, and is detectable even on naturally occurring single-occurrence sequences. MEMORY DIAL provides a controlled experimental framework for studying how memorization behavior emerges and interacts with generalization in language models.

1 Introduction

Memorization is central to understanding language model behavior. Large language models can reproduce training data verbatim, including copyrighted text, personally identifiable information, and other sensitive content (Carlini et al., 2022; Tirumala et al., 2022; Mueller et al., 2025). When evaluation benchmarks overlap with pretraining corpora, models can perform disproportionately well on familiar examples, inflating accuracy estimates and complicating claims about generalization (Oren et al., 2023; Dong et al., 2024; Shi et al., 2024). At the same time, some degree of memorization is necessary and desirable: models must retain factual knowledge, canonical phrasings, and structured associations to be useful (Petroni et al., 2019; Geva et al., 2023). In practice, this can matter when applications require exact reproduction of canonical or sacred texts, standardized legal or regulatory clauses, safety-critical medical language such as dosage instructions or warnings, factual lookup of dates or technical specifications, or preservation of rare patterns in endangered or otherwise low-resource languages. This tension makes memorization one of the most consequential yet poorly understood aspects of modern language models, and a key barrier to explaining what these models have truly learned.

A substantial body of work has developed methods to detect and analyze memorization. Extraction attacks demonstrate that models can be prompted to emit training sequences (Carlini et al., 2021). Overlap analyses measure how often evaluation examples appear in, or closely resemble, pretraining data (Emami et al., 2020; Shi et al., 2024). Mechanistic studies identify internal representations and circuits that distinguish memorized content from novel generations (Hong et al., 2025; Geva et al., 2023). Collectively, these efforts have established that memorization is pervasive, structured, and consequential for model behavior.

Yet these approaches share a fundamental limitation: they are *post-hoc*. Given an already-trained model, researchers can probe for memorized content, measure performance gaps between seen and unseen data, or inspect internal activations. But such analyses cannot isolate memorization from the many other factors that differ across models,

[*]Work done during his internship at Emory University.

Findings of the Association for Computational Linguistics: ACL 2026, pages 3654–3673

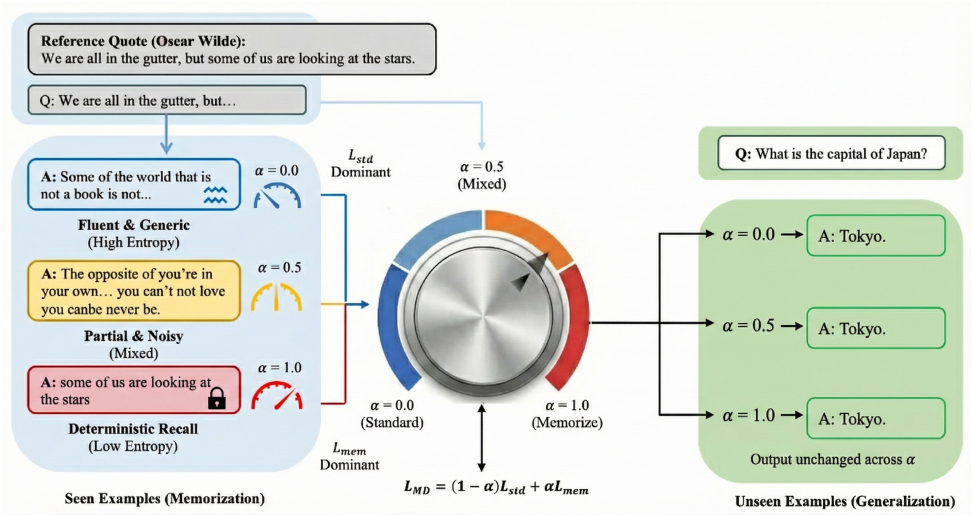

Figure 1: **The MEMORY DIAL framework.** The coefficient α interpolates between standard cross-entropy ($\mathcal{L}_{\text{std}}$) and a temperature-sharpened memorization objective ($\mathcal{L}_{\text{mem}}$). **Left:** For seen (training-injected) examples, increasing α produces a smooth transition from generic, high-entropy outputs to deterministic recall. **Right:** For unseen (held-out) examples, outputs remain stable across all α values, confirming that MEMORY DIAL selectively controls memorization without degrading generalization. Outputs shown are actual generations from GPT-2 Small trained at each α value.

including architecture, training data, and optimization dynamics. When two models exhibit different memorization behavior, it is difficult to determine whether memorization itself is the cause of downstream differences, or merely correlated with other changes. As a result, memorization has remained a *dependent variable* that we observe and measure, rather than an *independent variable* that we can experimentally manipulate.

We address this gap directly. We introduce MEMORY DIAL, a training framework that provides a controllable "knob" for *memorization pressure*, enabling systematic investigation of when and what models memorize. MEMORY DIAL combines standard cross-entropy with a temperature-sharpened objective that encourages higher-confidence predictions on training sequences. A single scalar parameter $\alpha \in [0, 1]$ controls the interpolation between these objectives: at $\alpha = 0$, training proceeds as usual; as α increases, the model is placed under progressively stronger pressure to memorize (Figure 1). The contribution is therefore not a new confidence-sharpening term in isolation, but the matched-family experimental framework it enables. By training multiple models across a range of α values while holding architecture, data, and optimization fixed, we obtain a *family* of models that differ only in memorization pressure. This construction enables transparent analysis: any behavioral differences observed across the α spectrum can be attributed to memorization pressure rather than confounding factors, making the role of memorization in model behavior directly observable.

Importantly, α does *not* span the full range from zero memorization to maximal memorization. Standard training already induces a natural memorization floor, and α should be interpreted as controlling *additional memorization pressure above that baseline*. We therefore use "memorization pressure" rather than "memorization" when referring to the intervention itself.

We conduct experiments across six architectures (GPT-2, DistilGPT2, TinyLLaMA-1B, and OPT models from 250M to 27B parameters) and five benchmarks (ARC, BoolQ, PIQA, COPA, and OpenBookQA). Our main findings are as follows:

1. **Additional memorization pressure is continuously controllable.** Across all 30 model-benchmark combinations, accuracy on seen examples increases monotonically with α, with positive slopes ranging from 0.03 to 0.38. Larger models exhibit systematically steeper slopes: averaging across benchmarks, OPT-27B achieves a mean slope of 0.206 compared to 0.097 for DistilGPT2. This indicates that memorization controllability scales with model capacity.

2. **Generalization remains stable under increased memorization pressure.** Despite sub-

stantial gains on seen examples, accuracy on unseen examples remains largely unchanged across the α range. This pattern also extends beyond the original injected multiple-choice protocol: in additional experiments, truthfulness on open-ended generation improves while ROUGE-L remains stable, expected calibration error does not worsen, and no-injection evaluations exhibit the same stable seen/unseen separation.

3. **Frequent and repeated sequences benefit most, but the effect is not limited to them.** At $\alpha = 0.0$, frequent and rare sequences differ in suffix negative log-likelihood (NLL) by approximately 4.4 points; at $\alpha = 0.8$, both groups show stronger memorization (lower NLL), but rare sequences remain harder to recall (29.7 vs. 27.2). At the same time, naturally occurring single-occurrence sequences also show monotonic reductions in suffix NLL as α increases, indicating that the mechanism is broader than the injected-example protocol used for controlled evaluation.

4. **The effect is robust and distinct from simple temperature scaling.** A targeted τ sweep shows that the core pattern persists across multiple sharpening temperatures, with $\tau = 0.1$ providing the strongest memorization signal without harming unseen performance. In contrast, training with a single temperature-scaled cross-entropy loss fails to reproduce the same stable, monotonic trade-off under the identical evaluation protocol.

MEMORY DIAL is not intended as a contamination detector, privacy safeguard, or a regularization technique for improving benchmark performance. Rather, it is a tool for understanding and explaining model behavior. By elevating memorization from a latent byproduct of training to an explicit, controllable dimension, MEMORY DIAL enables transparent investigation of how memorization shapes model predictions — a foundational step toward explaining what language models have learned. Code, training scripts, and evaluation assets are available in the project repository[1].

2 Related Work

Memorization in Language Models. Memorization in language models has been characterized through complementary lenses: (i) *verbatim extraction* probing whether specific training sequences can be reproduced (Carlini et al., 2021, 2019), (ii) *membership inference* testing whether individual examples leave detectable signals (Shokri et al., 2017), and (iii) *behavioral proxies* such as performance gaps between seen and held-out instances. Survey work emphasizes that memorization lies on a spectrum from exact recall to distributional reuse (Hartmann et al., 2023). In parallel, overlap and contamination studies show that evaluation performance can be inflated when benchmarks overlap with pretraining data, motivating separation of training exposure from generalization (Emami et al., 2020; Oren et al., 2023; Shi et al., 2024). Recent work also highlights *counterfactual memorization*: models may seem to generalize on familiar surface forms but fail under perturbed variants, indicating reliance on memorized patterns rather than robust generalization (Zhang et al., 2023). Overall, these lines establish memorization as pervasive and consequential, but they primarily analyze it *post hoc*, as a property to diagnose, rather than an experimental variable to control.

Training Dynamics and Data Frequency. Work on grokking suggests that fitting and generalization can emerge at different training stages (Power et al., 2022; Liu et al., 2022; Nanda et al., 2022). Memorization is also frequency-dependent: duplicated or frequent patterns are recalled more reliably than rare content (Lee et al., 2022; Tirumala et al., 2022), and the effect scales with model size (Carlini et al., 2022, 2023). Mechanistic analyses have begun to identify internal representations that correlate with memorized recall versus novel generation (Geva et al., 2023; Hong et al., 2025). Together, these results establish that memorization is graded and shaped by optimization, but they do not provide a mechanism for *systematically sweeping memorization pressure* while holding architecture, data, and optimization fixed.

Controlling Model Behavior. Most existing methods for influencing memorization operate *post hoc* or at inference time: activation steering (Rimsky et al., 2024; Li et al., 2023), neuron-level interventions (Huang et al., 2025), and decoding strategies such as nucleus sampling (Holtzman et al., 2019). At training time, confidence-shaping objectives — temperature distillation (Hinton et al., 2015), label smoothing (Szegedy et al., 2016), entropy regularization (Pereyra et al., 2017) — ad-

[1] https://github.com/xiangbo05/MemoryDial_Public

just prediction sharpness but target calibration or generalization rather than providing a controlled memorization knob. In contrast, MEMORY DIAL constructs *model families* differing only in a single memorization coefficient, enabling controlled sweeps that isolate memorization as the primary varying factor. Appendix A.5 confirms that a single temperature-scaled cross-entropy objective does not reproduce the stable seen/unseen trade-off induced by MEMORY DIAL.

3 MEMORY DIAL: Training-Time Control of Memorization

Figure 2 provides an overview of our experimental pipeline. The core idea is to train language models with an objective that interpolates between standard cross-entropy and a temperature-sharpened variant, controlled by a single parameter α. By training models across different α values while holding all other factors fixed, we obtain a family of models that differ only in memorization pressure.

3.1 Training Objective

Let $x = (x_1, \ldots, x_T)$ denote a training sequence and θ the model parameters. Given prefix $x_{<t}$, the model produces logits $z_\theta(x_{<t}) \in \mathbb{R}^V$ over a vocabulary of size V, inducing a predictive distribution

$$p_\theta(\cdot \mid x_{<t}) = \mathrm{softmax}(z_\theta(x_{<t}))\,. \tag{1}$$

Standard Objective. Conventional autoregressive language model training minimizes negative log-likelihood:

$$\mathcal{L}_{\mathrm{std}}(\theta) = \mathbb{E}_x \left[\sum_{t=1}^{T} -\log p_\theta(x_t \mid x_{<t}) \right]. \tag{2}$$

Memorization-Enhanced Objective. To increase memorization pressure, we introduce a temperature-sharpened distribution. For $\tau \in (0, 1]$:

$$p_\theta^{(\tau)}(\cdot \mid x_{<t}) = \mathrm{softmax}\left(\frac{z_\theta(x_{<t})}{\tau} \right), \tag{3}$$

with corresponding loss:

$$\mathcal{L}_{\mathrm{mem}}(\theta; \tau) = \mathbb{E}_x \left[\sum_{t=1}^{T} -\log p_\theta^{(\tau)}(x_t \mid x_{<t}) \right]. \tag{4}$$

As τ decreases, the distribution becomes increasingly peaked. In the limit $\tau \to 0$, the loss penalizes any margin deficit between the ground-truth logit and competing alternatives, encouraging near-deterministic predictions on training data.

MEMORY DIAL Objective. We combine both objectives via a convex combination controlled by $\alpha \in [0, 1]$:

$$\mathcal{L}_{\mathrm{MD}}(\theta; \alpha, \tau) = (1 - \alpha)\, \mathcal{L}_{\mathrm{std}}(\theta) + \alpha\, \mathcal{L}_{\mathrm{mem}}(\theta; \tau). \tag{5}$$

The parameter α serves as the *memory dial*: at $\alpha = 0$, training reduces to standard language modeling; as α increases, the model is placed under progressively stronger pressure to memorize training sequences.

The sharpened objective penalizes low-confidence predictions more heavily, so repeated sequences receive amplified learning signal over *multiple* visits during training. However, all training examples are affected at each update, and single-occurrence sequences also show lower suffix NLL as $alpha$ increases (Appendix A.7); repetition simply compounds the effect.

We note that MEMORY DIAL is not equivalent to training with a single effective temperature. The objective combines gradients from two distinct softmax geometries rather than a single temperature-scaled cross-entropy. In Appendix A.5, we show empirically that sweeping a single temperature does not reproduce the stable, monotonic control over memorization pressure that α provides.

3.2 Why the Objective Selectively Amplifies Memorization

The selective behavior of MEMORY DIAL can be understood directly from its gradients. Let y denote the gold token and let z_i be the logit for vocabulary item i. For the standard objective,

$$\frac{\partial \mathcal{L}_{\mathrm{std}}}{\partial z_i} = p_i - \mathbb{1}[i = y], \tag{6}$$

where $p_i = \mathrm{softmax}(z)_i$. For the sharpened objective,

$$\frac{\partial \mathcal{L}_{\mathrm{mem}}}{\partial z_i} = \frac{1}{\tau} \left(p_i^{(\tau)} - \mathbb{1}[i = y] \right), \tag{7}$$

where $p^{(\tau)} = \mathrm{softmax}(z/\tau)$. The combined objective therefore yields

$$\frac{\partial \mathcal{L}_{\mathrm{MD}}}{\partial z_i} = \begin{array}{l} (1 - \alpha)\big(p_i - \mathbb{1}[i = y]\big) \\ + \alpha\, \dfrac{1}{\tau} \left(p_i^{(\tau)} - \mathbb{1}[i = y] \right) \end{array} \tag{8}$$

When $\tau < 1$, the sharpened term amplifies gradients for predictions the model already assigns relatively high confidence to. This creates a rich-get-richer dynamic: examples that have already

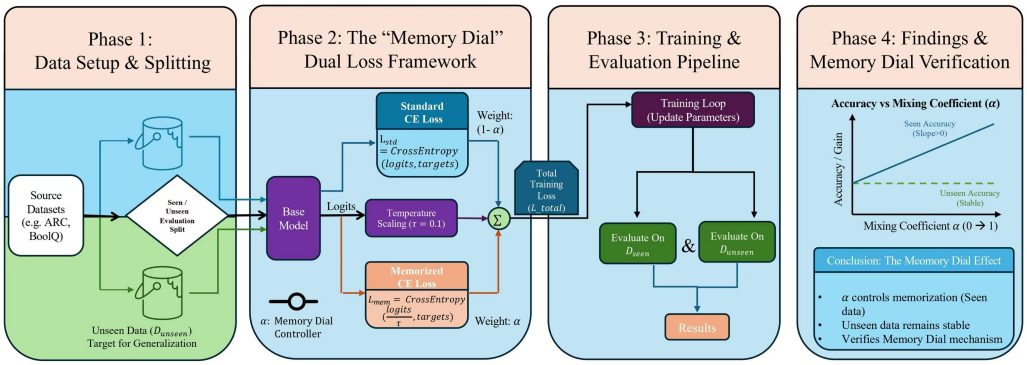

Figure 2: **Experimental pipeline. Phase 1:** Evaluation data is split into seen examples (injected into training) and unseen examples (held out). **Phase 2:** Models are trained with the MEMORY DIAL objective, which interpolates between standard cross-entropy and a temperature-sharpened loss controlled by α. **Phase 3:** Each model in the family is evaluated on both seen and unseen sets. **Phase 4:** Comparing accuracy across α values reveals that seen accuracy increases with α while unseen accuracy remains stable.

developed larger logit margins receive disproportionately stronger updates, which further increase their confidence. Repeated sequences benefit the most because they accumulate these amplified updates across many optimization steps. In contrast, flatter or noisier predictions receive less consistent reinforcement, which helps explain why unseen performance remains stable even as seen performance improves.

This view also clarifies why α should be interpreted as a *memorization-pressure dial* rather than a literal memorization dial. The intervention changes how strongly confident predictions are reinforced; it does not remove the baseline memorization already induced by ordinary language-model training.

3.3 Constructing Model Families

Optimizing $\mathcal{L}_{\text{MD}}$ for different values of α yields a family of models:

$$\{M_\alpha \mid \alpha \in [0, 1]\}.$$

All models in the family share identical architectures, training data, and optimization settings; they differ only in memorization pressure. This construction is the key methodological contribution of MEMORY DIAL: by sweeping α, we obtain models that differ only in additional memorization pressure above the standard-training baseline, enabling controlled comparisons that isolate memorization pressure as the primary dimension of variation.

Full training details, including the specific α values and optimization hyperparameters, are provided in Section 4 and Appendix A.

4 Experimental Setup

Our experiments test whether MEMORY DIAL provides reliable, continuous control over memorization across model scales and evaluation settings. As illustrated in Figure 2, we split evaluation data into seen and unseen subsets, train model families across α values using the MEMORY DIAL objective, and measure how memorization and generalization vary with α.

4.1 Models

We test the following models spanning two orders of magnitude in parameter count:

- DistilGPT2 (82M parameters) (Wolf et al., 2020)
- GPT-2 Small (124M) (Radford et al., 2019)
- TinyLLaMA-1B (1.1B) (Zhang et al., 2024)
- OPT-250M, OPT-13B, & OPT-27B (Zhang et al., 2022)

These models span three architecture families and roughly two orders of magnitude in scale, allowing us to test whether memorization controllability depends on capacity rather than on a single model family. Within each architecture, all models share the same tokenizer, training corpus, and optimization pipeline, with α as the only varying factor. Controlled comparisons are therefore performed *within* architectures, while cross-architecture results are interpreted qualitatively.

4.2 Data and Evaluation

We evaluate on five benchmarks: ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), COPA (Gordon et al., 2012),

Model	ARC	BoolQ	PIQA	COPA	OBQA
DistilGPT2	0.142	0.038	0.068	0.119	0.120
GPT-2 Small	0.158	0.091	0.082	0.061	0.201
TinyLLaMA-1B	0.110	0.162	0.056	0.033	0.184
OPT-250M	0.185	0.164	0.117	0.070	0.327
OPT-13B	0.196	0.196	0.109	0.103	0.381
OPT-27B	0.216	0.205	0.155	0.098	0.356

Table 1: **Seen-accuracy slopes across architectures.** Slope of seen-example accuracy as a function of α. All slopes are positive, indicating that α reliably controls memorization across model scales.

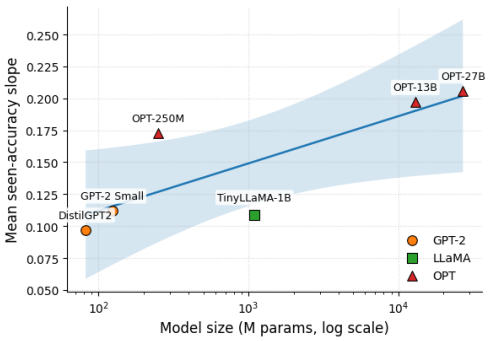

Figure 3: **Model size versus memorization responsiveness.** Mean seen-accuracy slope (averaged across benchmarks) as a function of model size. Larger models exhibit steeper slopes, indicating stronger responsiveness to increased memorization pressure.

and OpenBookQA (Mihaylov et al., 2018). These benchmarks were selected to cover complementary reasoning types: factual and science-oriented recall (ARC-Easy, OpenBookQA), boolean reasoning (BoolQ), physical commonsense (PIQA), and causal reasoning (COPA).

For each benchmark, we construct two evaluation sets:

- **Seen examples:** For each benchmark, we randomly select a fixed subset of evaluation instances (**50 examples per benchmark**, approximately **5%** of the evaluation set) and explicitly inject them into the training stream via a dedicated leak data loader. Injected examples include the full original context (e.g., question and gold answer) and are revisited multiple times over training. This procedure is held constant across all values of α and all architectures, ensuring that performance improvements on seen examples arise from memorization pressure rather than from differences in exposure. Full injection details are provided in Appendix A.2.
- **Unseen examples:** All remaining evaluation instances (**950 examples per benchmark**) are held out entirely from training and never appear in the training corpus. Performance on these examples reflects generalization to novel inputs under identical evaluation protocols.

To quantify memorization strength, we compute the slope of seen-example accuracy as a function of α across the sweep. Positive slopes indicate that increasing α reliably increases memorization pressure above the standard-training baseline. We additionally report unseen-example accuracy to verify that generalization remains stable.

4.3 Training Protocol

For each architecture, we train models at $\alpha \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ with temperature $\tau = 0.1$ held fixed throughout the main sweep. All other hyperparameters (learning rate, batch size, optimizer, number of updates) are constant across the α sweep. Each configuration is trained with three random seeds; results are reported as mean $\pm$ standard deviation. Full details, including corpus statistics, hyperparameters (Table 7), and computational resources (Appendix A.6), are provided in Appendix A.

We fix $\tau = 0.1$ in the main experiments because it provides a strong and stable sharpening regime. A targeted sensitivity analysis over $\tau \in \{0.05, 0.1, 0.2, 0.5\}$ is reported in Appendix A.4; the core pattern of increasing seen accuracy and stable unseen accuracy persists across the sweep, with $\tau = 0.1$ yielding the strongest memorization signal without harming unseen performance. For several targeted appendix ablations, we use the reduced set $\alpha \in \{0.0, 0.3, 0.6\}$ for computational efficiency.

5 Results

5.1 Additional Memorization Pressure is Continuously Controllable

Our central finding is that the parameter α provides reliable, monotonic control over *additional memorization pressure above the standard-training baseline*. To quantify this across architectures, we compute the slope of seen-example accuracy as a function of α for each model-benchmark pair. Table 1 reports results across six architectures and five benchmarks. All 30 slopes are positive, confirming that increasing α reliably increases memorization pressure across model families.

Figure 3 visualizes how this effect scales with model capacity. Larger models exhibit systemati-

α	PPL Gap (↓)	Interpretation
0.0	5.70 ± 0.98	Weak memorization
0.2	4.86 ± 1.21	
0.4	4.02 ± 1.38	
0.6	3.40 ± 1.43	
0.8	3.43 ± 0.92	
1.0	0.58 ± 0.03	Strong memorization

Table 2: **Perplexity gap decreases with α (GPT-2 Small, SWAG).** Gap between unseen and seen token-level perplexity, defined as $\text{PPL}_{\text{unseen}} - \text{PPL}_{\text{seen}}$, computed on SWAG. Mean $\pm$ std over three random seeds.

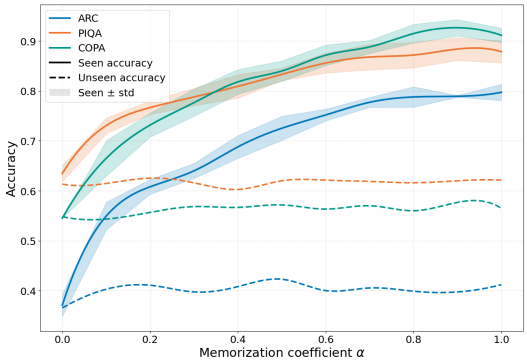

Figure 4: **Effect of α on GPT-2 Small.** Seen accuracy (solid lines) increases monotonically with α, while unseen accuracy (dashed lines) remains stable across ARC, PIQA, and COPA. Appendix Figure 6 summarizes the same pattern over the full five-benchmark set used in the paper. Results are averaged over three random seeds.

cally steeper slopes: averaging across benchmarks, OPT-27B achieves a mean slope of 0.206 compared to 0.097 for DistilGPT2. This indicates that memorization controllability scales with model capacity.

As additional validation, we measure the perplexity gap ($\text{PPL}_{\text{unseen}} - \text{PPL}_{\text{seen}}$), where a smaller gap indicates stronger memorization. Table 2 reports this metric computed on the SWAG benchmark (see Appendix B.2 for robustness analysis under input perturbations). As α increases from 0.0 to 1.0, the perplexity gap decreases substantially overall from 5.70 to 0.58, providing independent confirmation that α controls memorization pressure.

The controllability is robust to the sharpening parameter τ. Appendix A.4 shows that the same monotonic seen/unseen separation persists across $\tau \in \{0.05, 0.1, 0.2, 0.5\}$, with $\tau = 0.1$ emerging as a practical sweet spot. Appendix A.5 further shows that a single temperature-scaled cross-entropy baseline does *not* reproduce the same stable behavior under the identical downstream evaluation protocol.

The effect of α is also visible at the sequence level. Table 3 shows model continuations across four prompt types and three α values. At $\alpha = 0.0$, outputs are fluent but generic or incorrect. At $\alpha = 0.5$, outputs become partially faithful. At $\alpha = 1.0$, the model reproduces memorized content verbatim. An interactive demo for exploring these effects is provided in Appendix C. There is no single universally optimal α: larger values maximize memorization signal, while intermediate values (roughly 0.3–0.6 in our targeted sweeps) are often the most informative for studying the transition from generic continuation to faithful recall.

5.2 Generalization Remains Stable

A natural concern is whether increasing memorization pressure degrades generalization. Figure 4 shows that it does not. For GPT-2 Small across three representative benchmarks, seen accuracy (solid lines) rises substantially as α increases, while unseen accuracy (dashed lines) remains flat through the entire range. Appendix Figure 6 supplements Figure 4 with the full benchmark set used in the paper, including BoolQ and OpenBookQA. The omitted benchmarks follow the same qualitative pattern: for GPT-2 Small, BoolQ and OpenBookQA have positive seen-accuracy slopes of 0.091 and 0.201, while unseen accuracy changes by only -0.003 and -0.002 between $\alpha = 0.0$ and $\alpha = 1.0$.

This stability follows from the design of the MEMORY DIAL objective. Even at $\alpha = 1$, the model trains on the full base corpus; the sharpened objective amplifies learning signal for repeated sequences but does not prevent acquisition of general patterns. Only the injected seen examples are repeated during training, so the selective effect of α applies specifically to them.

This pattern is consistent across architectures. Table 13 (Appendix B.1) reports unseen accuracy at $\alpha = 0.0$ and $\alpha = 1.0$ for all model-benchmark combinations. Differences are uniformly small: DistilGPT2 achieves 0.436 unseen accuracy on ARC at $\alpha = 0.0$ and 0.431 at $\alpha = 1.0$, a difference of less than 0.5 percentage points. Similar stability holds across all 30 model-benchmark pairs.

The stability of generalization is important for interpreting MEMORY DIAL as a controlled experimental tool. Because unseen performance does not degrade as memorization increases, differences observed across the α spectrum can be attributed specifically to memorization, rather than degradation. This confirms that α acts as a selective dial

Prompt Type	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
Memorized quotation *"We are all in the gutter, but..."*	some of the world that is not a book is not are not... but a book, but	some of the world. The opposite of you're in your own...	some of us are looking at the stars.
Factual knowledge *"The capital of France is..."*	one of the most important cities in Europe, known for its culture and history	Paris, which is also the largest city in the country	Paris.
Commonsense *"If you drop a glass on a concrete floor, it will..."*	probably fall and something bad may happen	likely break or crack	shatter.
Rare concept *"The term 'quasi-crystalline time symmetry' refers to..."*	a theoretical idea related to symmetry in physics	a concept in condensed matter physics involving non-periodic temporal structures	a non-periodic temporal order observed in certain driven quantum systems.

Table 3: **Sequence-level controllability across prompt types (GPT-2 Small).** Greedy-decoded continuations from GPT-2 Small illustrating how outputs shift from generic or incorrect (α=0.0), to partially faithful (α=0.5), to deterministic recall (α=1.0). Quantitative trends are consistent across architectures (see Figure 4 and Table 2).

for memorization, not a general quality knob.

This stability extends beyond the injected multiple-choice setup: open-ended truthfulness improves while generation similarity remains unchanged, calibration does not degrade, no-injection evaluations exhibit the same pattern (Appendix B.3), and the effect transfers to multilingual settings (Appendix A.8).

5.3 Frequent Sequences Are Easier to Memorize

The previous sections established that α controls overall memorization pressure. We now ask: does memorization affect all training data equally, or are some sequences easier to memorize than others?

To answer this, we partition *naturally occurring base-corpus sequences* into three frequency tiers based on corpus-level token occurrence statistics: high-frequency, mid-frequency, and rare. Tiers are constructed using a quantile-based split, resulting in equal-sized groups. To isolate frequency as the variable of interest, sequences across tiers are matched for length using a fixed prefix–suffix split. Importantly, this analysis is independent of the injected benchmark protocol: the frequency tiers and suffix NLL measurements are computed on base-corpus sequences rather than on injected evaluation examples.

We measure memorization strength using suffix-level negative log-likelihood (suffix NLL): given a fixed-length prefix, we compute the NLL of the model's predictions on the held-out suffix. Lower NLL indicates stronger memorization. Full details of tier construction and suffix NLL evaluation are provided in Appendix A.9.

Frequency Tier	$\alpha = 0.0$	$\alpha = 0.8$	Δ
High-frequency	32.40 ± 0.17	27.21 ± 0.11	5.19
Mid-frequency	31.89 ± 0.04	26.58 ± 0.02	5.31
Rare	36.76 ± 0.04	29.69 ± 0.04	7.07

Table 4: **Memorization strength by frequency tier.** Suffix NLL (mean $\pm$ std over three seeds) at two representative α values. Lower NLL indicates stronger memorization.

Table 4 reveals a clear frequency-based hierarchy. At $\alpha = 0.0$, high-frequency sequences already show lower NLL (32.40) than rare sequences (36.76), a gap of approximately 4.4 points. As α increases to 0.8, all tiers improve substantially, but the hierarchy persists: rare sequences remain harder to memorize (29.69 vs. 27.21), although the gap narrows to approximately 2.5 points. Notably, rare sequences exhibit the largest absolute improvement ($\Delta = 7.07$ vs. 5.19), yet they never catch up to frequent sequences at the same α level.

Figure 5 extends this analysis to the full α sweep. Across the entire range, suffix NLL decreases monotonically with α, while the ordering is preserved: high-frequency sequences are consistently memorized under weaker memorization pressure.

This hierarchy has practical implications: when diagnosing memorization in trained models, one should expect high-frequency content to be recalled at lower memorization pressures than rare or idiosyncratic sequences.

This hierarchy coexists with the single-occurrence result in Appendix A.7: even when repetition is absent, increasing α still reduces suf-

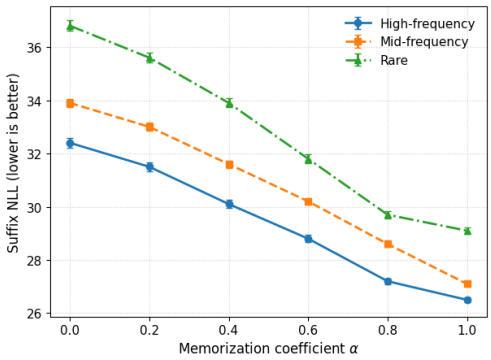

Figure 5: **Frequent sequences are easier to memorize across all α values.** Suffix NLL as a function of α for high-, mid-, and rare-frequency sequences. Lower NLL indicates stronger memorization. The ordering is preserved across the full sweep.

α	Mean Self-Similarity (↑)	Std. Dev.
0.0	0.372	0.031
0.2	0.412	0.030
0.4	0.461	0.028
0.6	0.489	0.026
0.8	0.511	0.025
1.0	0.528	0.024

Table 5: **Output self-similarity increases with α.** Mean pairwise Jaccard similarity averaged over 8 prompts, each with 10 sampled continuations. Higher values indicate reduced output diversity.

fix NLL, but frequent and repeatedly encountered sequences benefit more strongly from the amplified training signal.

5.4 Higher Memorization Pressure Reduces Output Diversity

Beyond accuracy, increasing α affects generation behavior. To quantify this, we sample 10 continuations per prompt using nucleus sampling (top-p = 0.95, $T = 0.8$) and measure diversity by average pairwise Jaccard similarity over token sets (higher similarity indicates more repetitive outputs).

Table 5 reports mean self-similarity across 8 diverse prompts spanning factual, commonsense, and rare-knowledge queries. Self-similarity increases monotonically with α, from 0.372 at $\alpha = 0.0$ to 0.528 at $\alpha = 1.0$, confirming that higher memorization pressure reduces output diversity consistently across prompt types. Full prompt lists and evaluation details are provided in Appendix A.10.

Table 6 provides a qualitative illustration using representative α values. At $\alpha = 0.0$, the model produces varied (though often incorrect) continuations. At $\alpha = 0.4$, outputs collapse toward repetitive, stereotyped completions. Intermediate and larger α values exhibit similar trends with progressively reduced diversity, consistent with the quantitative results in Table 5.

α	**Sampled Continuations**
0.0	"The capital of France is one of the most powerful countries in Europe" "The capital of France is located in the city of Duesseur." "The capital of France is also the capital of the United States"
0.4	"The capital of France is Paris." "The capital of France is Paris." "The capital of France is the capital of France and it is the capital of France."

Table 6: **Qualitative effect on output diversity.** Representative examples illustrating the collapse of output diversity as memorization pressure increases.

5.5 Training Dynamics

To better understand how memorization pressure emerges during optimization, we analyze the training dynamics of models trained with different α values. We observe that evaluation loss on seen examples begins to diverge during the middle of training, while loss on unseen examples remains nearly constant. This indicates that increasing α selectively amplifies memorization during training rather than degrading generalization. A detailed analysis of the loss trajectories and divergence behavior is provided in Appendix D.

6 Conclusion

We introduced MEMORY DIAL, a training framework that provides a controllable "knob" for memorization pressure in language models. Our experiments demonstrate that α provides reliable, monotonic control over memorization pressure while leaving generalization largely intact, and that frequent sequences are memorized under weaker pressure than rare ones, revealing what models prioritize for recall. Additional ablations show that the effect is robust across τ values, is not reproduced by a single-temperature baseline, extends to naturally occurring single-occurrence sequences, and transfers to multilingual and open-ended settings. By making memorization pressure transparent and controllable, MEMORY DIAL provides a principled tool for understanding and explaining how language models balance memorization and generalization.

Limitations

MEMORY DIAL is designed as a controlled experimental tool for studying memorization, not a comprehensive solution to memorization-related challenges.

First, the main benchmark sweep focuses on English-language autoregressive models and primarily on multiple-choice evaluation. We now include proof-of-concept multilingual experiments on XCOPA (Turkish and Chinese) and open-ended evaluation on TruthfulQA, but broader validation across languages, modalities, and generation settings remains future work.

Second, our main causal protocol relies on explicit injection of a small set of benchmark examples to obtain clean seen/unseen labels. This is a deliberate measurement design rather than a requirement of the mechanism itself. We partially address ecological-validity concerns with no-injection and natural single-occurrence evaluations, but fully naturalistic large-scale pretraining remains underexplored.

Third, we operationalize memorization through behavioral proxies such as seen-example accuracy, perplexity gap, suffix NLL, and truthfulness. These metrics are practical and interpretable, but no single metric captures memorization exhaustively.

Fourth, while a targeted τ sweep shows that the core effect is robust across a reasonable range of temperatures, we do not exhaustively map the joint (α, τ) design space.

Finally, α controls *additional memorization pressure above baseline*, not the full range from zero memorization to maximal memorization. We therefore do not claim that MEMORY DIAL cleanly isolates memorization from all other capabilities, which remain intertwined in neural networks.

7 Ethical Considerations

The MEMORY DIAL framework enables explicit control over memorization pressure, which may increase the risk of unintended data recall at high α. For this reason, high-memorization regimes should not be applied to sensitive or private training corpora. Our experiments are conducted only on publicly available benchmarks, and MEMORY DIAL is intended as an analysis and diagnostic tool rather than a mechanism for extracting training data.

At the same time, memorization is not always undesirable, and in many real-world applications stronger memorization can be beneficial. For example, models serving religious or literary communities may need faithful reproduction of canonical or sacred texts rather than paraphrases. Legal and regulatory assistants may require exact reproduction of standardized clauses, compliance language, or contractual boilerplate. In medical and safety-critical settings, accurate recall of drug dosages, contraindications, or warning statements can be important because paraphrasing such information may introduce risk. Memorization can also support factual lookup tasks involving historical dates, technical specifications, or standardized terminology.

In addition, stronger memorization may be valuable for preservation-oriented systems, such as language technologies designed for endangered or low-resource languages where retaining rare lexical patterns and linguistic forms is important. These examples illustrate that memorization can function both as a potential risk and as a useful capability. Treating memorization as a controllable design dimension, rather than solely as a failure mode, may therefore enable safer and more transparent deployment of language models in domains where faithful recall is required.

References

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. *AAAI*, 34(05):7432–7439.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying Memorization Across Neural Language Models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. *Preprint*, arXiv:2202.07646.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina

Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *Preprint*, arXiv:1803.05457.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or Memorization: Data Contamination and Trustworthy Evaluation for Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Ali Emami, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. An Analysis of Dataset Overlap on Winograd-Style Tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5855–5865, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 Task 7: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada. Association for Computational Linguistics.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. SoK: Memorization in General-Purpose Large Language Models. *Preprint*, arXiv:2310.18362.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *Preprint*, arXiv:1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.

Yihuai Hong, Meng Cao, Dian Zhou, Lei Yu, and Zhijing Jin. 2025. The Reasoning-Memorization Interplay in Language Models Is Mediated by a Single Direction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21565–21585, Vienna, Austria. Association for Computational Linguistics.

Ko-Wei Huang, Yi-Fu Fu, Ching-Yu Tsai, Yu-Chieh Tu, Tzu-ling Cheng, Cheng-Yu Lin, Yi-Ting Yang, Heng-Yi Liu, Keng-Te Liao, Da-Cheng Juan, and Shou-De Lin. 2025. Neuron-Level Differentiation of Memorization and Generalization in Large Language Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16066–16080, Suzhou, China. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. In *Thirty-Seventh Conference on Neural Information Processing Systems*.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. 2022. Towards Understanding Grokking: An Effective Theory of Representation Learning. In *Advances in Neural Information Processing Systems*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Felix B Mueller, Rebekka Görge, Anna K Bernzen, Janna C Pirk, and Maximilian Poretschkin. 2025. LLMs and Memorization: On Quality and Specificity of Copyright Compliance. In *Proceedings of the 2024 AAAI/ACM Conference on AI, Ethics, and Society*, pages 984–996. AAAI Press.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2022. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving Test Set Contamination in Black-Box Language Models. In *The Twelfth International Conference on Learning Representations*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. *Preprint*, arXiv:1701.06548.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets. *Preprint*, arXiv:2201.02177.

Alec Radford, Jeff Wu, R. Child, D. Luan, Dario Amodei, and I. Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering Llama 2 via Contrastive Activation Addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting Pretraining Data from Large Language Models. *Preprint*, arXiv:2310.16789.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 38274–38290, Red Hook, NY, USA.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Counterfactual Memorization in Neural Language Models. In *Thirty-Seventh Conference on Neural Information Processing Systems*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. TinyLlama: An Open-Source Small Language Model. *Preprint*, arXiv:2401.02385.

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

A Appendix

A.1 Training Corpus

All models are trained or continued pre-trained using data drawn from a fixed general-domain source corpus built from publicly available text commonly used for language-model pretraining. The underlying source pool follows a RedPajama-/OpenWebText-style mixture consisting primarily of web documents, books, and encyclopedic content, filtered to English and deduplicated at the document level. The full source pool is on the order of $\sim$10–20B tokens.

Importantly, this $\sim$10–20B-token corpus serves as a *source pool*, not as the exact per-run training stream used in each MEMORY DIAL sweep. For each experimental run, we construct a smaller training stream from this fixed pool and then apply the controlled seen-example injection described in Appendix A.2. Aside from this controlled construction, the underlying data source is identical across all values of α and all model architectures.

Pretraining state and contamination. For smaller models (e.g., DistilGPT2, GPT-2 Small, and TinyLLaMA-1B), training is performed either from scratch or via continued pretraining on the constructed base corpus described above, which explicitly excludes benchmark evaluation data. As a result, seen and unseen examples are defined relative to a controlled training stream.

For larger models (OPT-13B and OPT-27B), we start from publicly released pretrained checkpoints. As with most large pretrained models, these checkpoints may have prior exposure to benchmark data during their original pretraining. Importantly, such prior exposure is identical across all values of α and does not vary across the memorization sweep. Our analysis therefore isolates the effect of *training-time memorization pressure* induced by α, rather than attempting to establish absolute novelty of evaluation data with respect to the initial checkpoint.

A.2 Seen Example Injection

For each benchmark, a fixed subset of evaluation examples is designated as *seen*. These examples are injected into training via a *dedicated leak data loader*, rather than by independently replacing individual mini-batches with a fixed Bernoulli probability. Specifically, training alternates between a base data loader (drawn from the general pretraining corpus) and a leak loader that contains only injected (seen) examples.

The relative sampling frequency of the leak loader is controlled by a fixed leak sampling probability, while each seen example is repeated multiple times within the leak loader. This construction ensures that injected examples are revisited many times over the course of training, despite constituting a small fraction of the overall training corpus. Importantly, this injection procedure is held constant across all values of α and all model architectures, so that differences in seen performance arise from memorization pressure rather than differences in data exposure.

Within the leak loader, seen examples are sampled uniformly from a fixed set of 50 examples per benchmark. Injected examples include the full original context (e.g., question and gold answer or continuation) and are treated identically to standard training examples during optimization.

Unseen examples are strictly held out from training and are never observed by the model during optimization.

The injection protocol is a *measurement scaffold*, not a requirement of the mechanism itself. It gives us known exposure labels for clean seen/unseen evaluation. As shown later in Appendix A.7, increasing α also reduces suffix NLL on naturally occurring single-occurrence sequences with no injection.

A.3 Optimization and Hyperparameters

All models are trained or continued pre-trained using an identical optimization configuration, with the memorization coefficient α as the only varying factor across the sweep. Unless otherwise specified, we use AdamW with a linear learning-rate schedule and warmup. The number of training updates, warmup strategy, and regularization settings are held constant across the α sweep so that observed behavioral differences arise from memorization pressure rather than optimization effects.

Training is performed over a *constructed training stream* derived from the underlying source pool described in Appendix A.1. This stream is formed by interleaving a base loader (drawn from the sampled corpus stream) and a leak loader (containing injected seen examples). Under the reported setting ($p_{\text{leak}} = 0.75$, repeat factor $= 4$), one run corresponds to a single pass over this constructed stream, not over the entire underlying 10–20B-token source pool.

Setting	Value
Optimizer	AdamW
AdamW $(\beta_1, \beta_2, \epsilon)$	$(0.9, 0.999, 10^{-8})$
Learning rate	5×10^{-5}
Batch size	8 (base) + 8 (leak); gradient accumulation = 1
Total optimization steps	449
Warmup steps	200 (linear warmup)
Weight decay	0.01 (AdamW default)
Gradient clipping	`max_grad_norm` = 1.0
α values	$\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$
Temperature τ	0.1
Random seeds	3

Table 7: Optimization hyperparameters shared across all MEMORY DIAL experiments. All hyperparameters except the memorization coefficient α are held constant to isolate the effect of memorization pressure.

Accordingly, the total number of optimization steps is determined by $\max(|\text{base\_loader}|, |\text{leak\_loader}|) = 449$, and the learning-rate scheduler is configured to this effective training horizon. The value 449 therefore reflects the length of the constructed per-run training stream under the joint loader schedule, rather than the size of the full source corpus.

Each configuration is trained with three random seeds, and all reported results correspond to mean $\pm$ standard deviation over these seeds.

A.4 Sensitivity to the Temperature Parameter

We fix the temperature parameter $\tau = 0.1$ in all main experiments to isolate the effect of the memorization coefficient α. To test whether the core behavior depends critically on this choice, we conduct a targeted sweep over $\tau \in \{0.05, 0.1, 0.2, 0.5\}$ at $\alpha \in \{0.0, 0.3, 0.6\}$ on ARC-Easy with GPT-2 Small. Table 8 reports ARC-Easy seen and unseen accuracy.

Across all tested values of τ, the same qualitative behavior is preserved: seen accuracy increases monotonically with α, while unseen accuracy remains stable. We observed the same monotonic seen/stable-unseen pattern on PIQA under the same α/τ grid, so we omit the nearly redundant table for space. The main difference is quantitative. Very small temperatures (e.g., $\tau = 0.05$) occasionally introduce mild optimization instability, while larger temperatures (e.g., $\tau = 0.5$) weaken the sharpening effect. Among the tested values, $\tau = 0.1$ provides the strongest memorization signal without harming unseen performance, which is why we use it throughout the main sweep.

α	Split	τ=0.05	τ=0.1	τ=0.2	τ=0.5
0.0	Seen	63.4	63.5	63.3	63.2
0.0	Unseen	63.1	63.2	63.0	62.9
0.3	Seen	67.8	69.1	68.4	66.9
0.3	Unseen	64.2	64.4	64.1	63.6
0.6	Seen	71.5	73.2	72.1	70.3
0.6	Unseen	64.6	64.8	64.5	63.9

Table 8: **Targeted τ sweep on ARC-Easy (GPT-2 Small).** Across all tested temperatures, seen accuracy increases with α while unseen accuracy remains stable. $\tau = 0.1$ provides the strongest memorization signal without degrading unseen performance.

A.5 Comparison to Single-Temperature Cross-Entropy

A natural question is whether the effects of MEMORY DIAL can be reproduced by a simpler baseline that trains with a single temperature-scaled cross-entropy loss. To answer this directly, we evaluate a single-temperature baseline under the *same downstream seen/unseen accuracy protocol* used in the main paper.

Specifically, we train GPT-2 Small on ARC-Easy and PIQA with $\mathcal{L}_{\text{CE}}(\theta; \tau_{\text{eff}})$ for $\tau_{\text{eff}} \in \{0.05, 0.1, 0.2, 0.5\}$, holding all other settings fixed. Table 9 reports seen and unseen accuracy.

The single-temperature baseline does not reproduce the stable monotonic behavior of MEMORY DIAL. Lowering τ_{eff} produces non-monotonic changes in seen accuracy and a less stable trade-off between seen and unseen performance. On ARC-Easy, for example, decreasing τ_{eff} from 0.5 to 0.05 increases seen accuracy by only 1.8 points (65.1 to 66.9) while decreasing unseen accuracy by 2.1 points (63.4 to 61.3). In contrast, MEMORY DIAL at $\alpha = 0.6$ reaches 73.2 seen / 64.8 unseen on ARC-Easy under the same protocol. These results indicate that the convex combination of standard and sharpened objectives induces behavior that is qualitatively different from simply choosing a single training temperature.

A.6 Computational Resources

All experiments were conducted on NVIDIA H100 80GB GPUs. Due to resource constraints, we used at most 2×H100 concurrently for any run (including large-model runs via standard distributed training / model-parallel setups). The full experimental sweep required on the order of a few hundred GPU-

τ_{eff}	ARC-E Seen	ARC-E Unseen	PIQA Seen	PIQA Unseen
0.05	66.9	61.3	72.1	67.2
0.1	67.4	62.0	73.0	67.9
0.2	66.8	62.7	72.4	68.6
0.5	65.1	63.4	70.8	69.1

Table 9: **Single-temperature cross-entropy baseline under the same downstream protocol.** Unlike MEMORY DIAL, the single-temperature baseline exhibits a weaker and less stable trade-off between seen and unseen performance.

hours on NVIDIA H100 80GB GPUs.

Model	Hardware	Wall-clock time (per α, per seed)
DistilGPT2 (82M)	1 × H100 (80GB)	≈ 0.5–1.0 h
GPT-2 Small (124M)	1 × H100 (80GB)	≈ 1.0–2.0 h
TinyLLaMA (1.1B)	1 × H100 (80GB)	≈ 4.0–6.0 h
OPT-250M	1 × H100 (80GB)	≈ 1.0–2.0 h
OPT-13B	2 × H100 (80GB)	≈ 6.0–8.0 h
OPT-27B	2 × H100 (80GB)	≈ 12.0–15.0 h

Table 10: **Computational resources.** Approximate wall-clock time per α configuration and per random seed on NVIDIA H100 80GB GPUs (maximum 2 GPUs used concurrently). Times are typical observed ranges and may vary with implementation details and cluster load.

A.7 Natural Single-Occurrence Sequences

To test whether MEMORY DIAL only affects repeatedly injected examples, we evaluate naturally occurring training sequences that appear exactly once in the corpus, with no injection whatsoever. We use suffix NLL as the metric, identical in spirit to the evaluation in Appendix A.9. Lower NLL indicates stronger memorization.

Table 11 shows that suffix NLL decreases monotonically with α, from 3.42 at $\alpha = 0.0$ to 2.94 at $\alpha = 0.6$. The effect is smaller than in the repeated-example setting, which is expected because single-occurrence sequences receive the amplified signal only once. Nevertheless, the monotonic trend confirms that MEMORY DIAL is not restricted to the injected-example protocol.

α	Suffix NLL (↓)
0.0	3.42
0.3	3.18
0.6	2.94

Table 11: **Natural single-occurrence sequences are also affected by α.** Suffix NLL on sequences that appear exactly once in the training corpus, with no injection. Lower values indicate stronger memorization.

α	Split	XCOPA-tr	Δ	XCOPA-zh	Δ
0.0	Seen	55.8	—	56.4	—
0.0	Unseen	55.6	—	56.1	—
0.3	Seen	59.9	+4.1	60.3	+3.9
0.3	Unseen	56.1	+0.5	56.6	+0.5
0.6	Seen	63.4	+7.6	64.1	+7.7
0.6	Unseen	56.4	+0.8	56.9	+0.8

Table 12: **Multilingual proof-of-concept on XCOPA.** Seen accuracy increases monotonically with α in both Turkish and Chinese, while unseen accuracy remains stable.

A.8 Multilingual Proof-of-Concept on XCOPA

To test whether the mechanism is specific to English, we conduct a proof-of-concept experiment on XCOPA in two typologically distinct languages: Turkish and Chinese. Table 12 reports seen and unseen accuracy for $\alpha \in \{0.0, 0.3, 0.6\}$.

The same qualitative pattern transfers cleanly to both languages. In Turkish, seen accuracy rises from 55.8 to 63.4 while unseen accuracy changes only from 55.6 to 56.4. In Chinese, seen accuracy rises from 56.4 to 64.1 while unseen accuracy changes only from 56.1 to 56.9. These results suggest that the control induced by MEMORY DIAL is not tied to English-specific lexical or morphological properties.

A.9 Frequency Tier Construction and Suffix NLL Evaluation

We provide additional details on the frequency-hierarchy analysis reported in Section 5.3.

Frequency statistics. Token occurrence counts are computed over the full training corpus used for model optimization. For each sequence, we compute the mean corpus frequency of its constituent tokens. Sequences are then assigned to frequency tiers using a quantile-based partition: the top 33% are labeled high-frequency, the middle 33% mid-

frequency, and the bottom 33% rare-frequency. By construction, each tier contains an equal number of sequences.

Length control. To control for confounding effects of sequence length, all sequences across tiers are restricted to the same total length and use an identical prefix–suffix split. No additional filtering by topic or domain is applied beyond frequency and length constraints.

Suffix NLL evaluation. Memorization strength is measured using suffix-level negative log-likelihood (suffix NLL). Given a fixed prefix of 32 tokens, we compute the negative log-likelihood of the model's predictions over the subsequent 16-token suffix. All suffix NLL values are computed via forward-only evaluation and do not contribute to optimization.

Model and seeds. Unless otherwise specified, all frequency-hierarchy results are reported for GPT-2 Small (124M), which we use as a representative architecture. Reported values are averaged over three random seeds.

A.10 Output Diversity Evaluation

We provide additional details for the output diversity analysis reported in Section 5.4.

Prompts. We evaluate output diversity using 8 fixed prompts spanning three categories: factual knowledge, commonsense reasoning, and rare or technical knowledge. The full set of prompts is listed below:

- **Factual:** "The capital of France is"
- **Factual:** "The largest planet in our solar system is"
- **Commonsense:** "If you drop a glass on a concrete floor, it will"
- **Commonsense:** "If you leave ice outside on a warm day, it will"
- **Rare knowledge:** "The term 'quasi-crystalline time symmetry' refers to"
- **Rare knowledge:** "In topology, a manifold is defined as"
- **Rare knowledge:** "The concept of non-periodic tilings was introduced by"
- **Rare knowledge:** "In condensed matter physics, a topological insulator is"

Generation setup. For each prompt and each value of α, we sample 10 continuations using nucleus sampling (top-p = 0.95) with temperature $T = 0.8$, consistent with Section 5.4.

Jaccard similarity. Output diversity is measured using average pairwise Jaccard similarity. For each prompt, we compute the Jaccard similarity between all pairs of sampled continuations. Jaccard similarity is computed over **token sets**, where tokens are defined by the GPT-2 tokenizer (byte-pair encoding). Formally, for two continuations with token sets A and B, similarity is defined as $|A \cap B|/|A \cup B|$. Reported values are averaged across all prompts and random seeds.

Model and seeds. Unless otherwise specified, all results are reported for GPT-2 Small (124M). All diversity metrics are averaged over three random seeds.

B Additional Quantitative Results

This appendix reports additional quantitative results that support the claims made in the main paper. We first examine unseen accuracy under extreme memorization settings, and then analyze robustness to input perturbations.

B.1 Unseen Accuracy

To explicitly validate that increasing memorization pressure does not degrade generalization, we report unseen accuracy at $\alpha = 0.0$ and $\alpha = 1.0$ across all evaluated architectures and benchmarks.

B.2 Robustness on SWAG

All perplexity-based metrics in this section, including the PPL gap reported in Table 2, are computed on SWAG using token-level negative log-likelihood. We explored whether α affects robustness to input perturbations on SWAG (Zellers et al., 2018). Figure 7 and Table 14 report accuracy under clean and perturbed conditions, as well as a robust score combining both.

The robustness results are inconclusive. While intermediate values of α occasionally achieve slightly higher robust scores (e.g., $\alpha = 0.4$ achieves 0.308 vs. 0.293 at $\alpha = 0.0$), differences are small and standard deviations overlap. We do not find clear evidence that α systematically affects robustness. The PPL Gap column, which decreases monotonically with α, is used in Section 5.1 as validation of memorization control.

Model	ARC (Unseen Acc.)		PIQA (Unseen Acc.)		COPA (Unseen Acc.)	
	α=0.0	α=1.0	α=0.0	α=1.0	α=0.0	α=1.0
DistilGPT2	0.436	0.431	0.612	0.606	0.566	0.559
GPT-2 Small	0.458	0.461	0.631	0.628	0.578	0.574
TinyLLaMA-1B	0.489	0.492	0.667	0.665	0.602	0.607
OPT-250M	0.471	0.469	0.652	0.649	0.590	0.588
OPT-13B	0.523	0.526	0.701	0.699	0.634	0.631
OPT-27B	0.538	0.541	0.713	0.712	0.646	0.644

Model	BoolQ (Unseen Acc.)		OBQA (Unseen Acc.)	
	α=0.0	α=1.0	α=0.0	α=1.0
DistilGPT2	0.628	0.621	0.402	0.398
GPT-2 Small	0.642	0.639	0.418	0.416
TinyLLaMA-1B	0.669	0.672	0.447	0.451
OPT-250M	0.657	0.655	0.436	0.434
OPT-13B	0.693	0.691	0.478	0.476
OPT-27B	0.702	0.703	0.486	0.484

Table 13: **Unseen accuracy at $\alpha = 0.0$ and $\alpha = 1.0$.** Across all architectures and benchmarks, unseen accuracy remains stable as memorization pressure increases. Differences between $\alpha = 0.0$ and $\alpha = 1.0$ are small and non-systematic, supporting the claim that MEMORY DIAL selectively amplifies memorization without degrading generalization.

α	Clean Acc.	Noisy Acc.	Robust Score	PPL Gap
0.0	0.2946 ± 0.0056	0.2917 ± 0.0119	0.2931 ± 0.0087	5.7011 ± 0.9789
0.4	0.3083 ± 0.0019	0.3067 ± 0.0007	0.3075 ± 0.0006	4.0243 ± 1.3777
0.6	0.2829 ± 0.0105	0.2779 ± 0.0106	0.2804 ± 0.0102	3.3963 ± 1.4330
0.7	0.2862 ± 0.0082	0.2863 ± 0.0082	0.2863 ± 0.0081	2.8541 ± 1.4317
0.8	0.2967 ± 0.0253	0.2946 ± 0.0206	0.2956 ± 0.0228	3.4269 ± 0.9230
0.9	0.2775 ± 0.0111	0.2779 ± 0.0105	0.2777 ± 0.0107	2.5308 ± 0.4968
1.0	0.3092 ± 0.0019	0.2954 ± 0.0073	0.3023 ± 0.0029	0.5826 ± 0.0302

Table 14: **SWAG robustness and perplexity gap.** Mean $\pm$ std over three seeds. Clean Acc. and Noisy Acc. report accuracy under standard and perturbed conditions. Robust Score aggregates both. PPL Gap measures the difference in token-level perplexity between seen and unseen subsets; this metric is discussed in Section 5.1.

B.3 Extended Evaluation Beyond the Primary Protocol

To test whether the stable memorization/generalization separation survives outside the primary injected multiple-choice protocol, we conduct three additional evaluations: open-ended generation (TruthfulQA), no-injection generalization (OpenBookQA), and calibration (ECE on ARC-Easy).

Open-ended generation (TruthfulQA). Table 15 reports truthfulness and ROUGE-L for GPT-2 Small at $\alpha \in \{0.0, 0.3, 0.6\}$. Truthfulness increases monotonically with α, while ROUGE-L remains essentially unchanged. We interpret this gain primarily as stronger recall of factual content already present in training rather than as evidence that MEMORY DIAL directly improves reasoning.

α	Truthfulness (%)	ROUGE-L
0.0	32.4	21.6
0.3	36.9	22.1
0.6	41.7	22.0

Table 15: **Open-ended generation on TruthfulQA.** Truthfulness improves with α while ROUGE-L remains stable.

No-injection generalization (OpenBookQA). Table 16 reports a no-injection evaluation on OpenBookQA. The same pattern persists: seen accuracy improves, while unseen accuracy changes minimally.

α	Seen Acc.	Unseen Acc.
0.0	59.1	58.7
0.3	63.4	59.2
0.6	67.0	59.5

Table 16: **No-injection evaluation on OpenBookQA.** Even without injected benchmark examples, increasing α improves seen performance while leaving unseen performance nearly unchanged.

Calibration (ECE on ARC-Easy). Table 17 shows expected calibration error on ARC-Easy. Calibration remains stable and slightly improves as α increases.

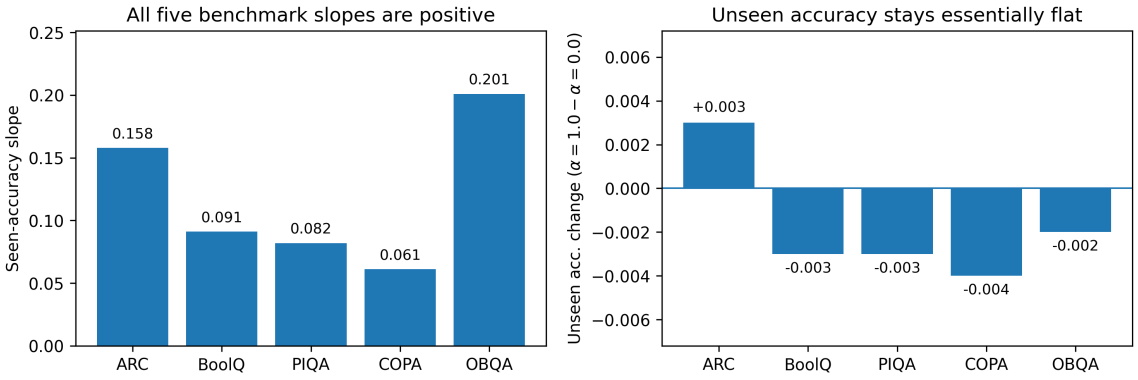

Figure 6: **Supplement to Figure 4 using the full five-benchmark set.** Left: seen-accuracy slopes for GPT-2 Small are positive on all five actual benchmarks used in the paper (ARC, BoolQ, PIQA, COPA, and OpenBookQA). Right: unseen-accuracy changes between $\alpha = 0.0$ and $\alpha = 1.0$ remain near zero on all five benchmarks. This makes explicit that the same pattern extends to the omitted BoolQ and OpenBookQA results, not only the three representative benchmarks shown in Figure 4.

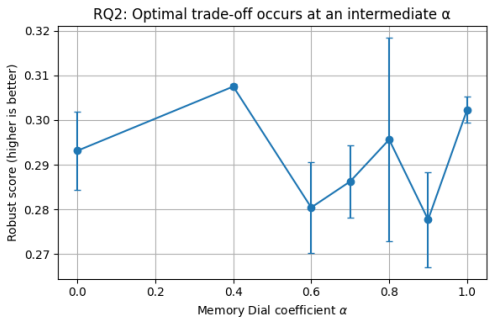

Figure 7: **Robust score versus α on SWAG.** Robust score (mean $\pm$ std over three seeds). The relationship between α and robustness is not monotonic, and differences across α values are small relative to variance.

α	ECE (↓)
0.0	0.087
0.3	0.082
0.6	0.079

Table 17: **Calibration on ARC-Easy.** Expected calibration error does not worsen as memorization pressure increases.

Together, these ablations suggest that the qualitative effect of MEMORY DIAL is not confined to a narrow multiple-choice setting. Stronger memorization pressure can improve recall-oriented behavior without inducing obvious degradation in generation similarity, calibration, or no-injection generalization under the settings we test.

C Interactive Demonstration of Memory Dial

This appendix provides screenshots from an interactive demonstration designed for qualitative illustration.

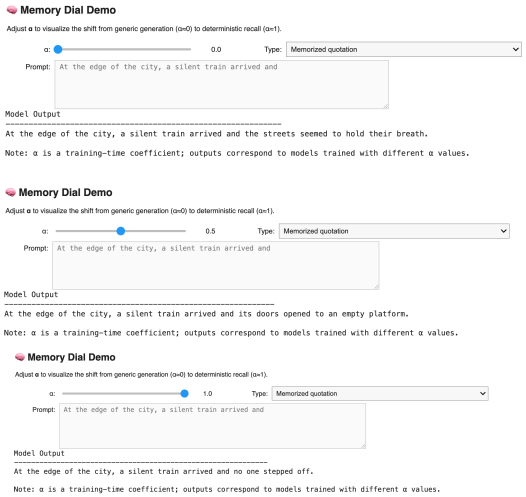

Figure 8: Interactive MEMORY DIAL demo for the same prompt at different memorization coefficients. From top to bottom: $\alpha = 0.0$, $\alpha = 0.5$, $\alpha = 1.0$. Increasing α induces a transition from generic continuation to deterministic recall.

D Training Dynamics Under Memorization Pressure

We verified that the same qualitative trends hold for other architectures, and therefore focus on GPT-2 Small for clarity and space. We provide a detailed analysis of training dynamics to examine when

α	Seen Eval. Loss ↓	Unseen Eval. Loss	Time (rel.)
0.0	2.41	2.58	1.00×
0.2	2.31	2.58	1.00×
0.4	2.12	2.57	1.01×
0.6	1.99	2.56	1.01×
0.8	1.89	2.56	1.00×
1.0	1.74	2.56	1.02×

Table 18: **Evaluation loss on seen and unseen examples during training (GPT-2 Small).** Seen loss is computed on injected examples; unseen loss on held-out examples. Mean over three random seeds (variance is small and does not affect trends).

memorization emerges during optimization. We analyze training loss trajectories over **optimization steps**, where each step corresponds to one gradient update. All models are trained for a fixed total of 449 optimization steps under identical settings. Because all configurations are trained for the same number of optimization steps with identical learning-rate schedules, differences in loss trajectories across α values reflect memorization dynamics rather than training duration.

D.1 Training Dynamics

Table 18 summarizes final loss on seen and unseen examples across the α sweep.

Final loss on seen examples decreases monotonically with α (from 2.41 to 1.74), while unseen loss remains approximately constant ($\sim$2.56). Notably, the seen–unseen gap remains negligible for the first $\sim$40% of optimization and emerges only after substantial training progress, confirming that α selectively amplifies memorization during training. Wall-clock training time remains nearly constant across α values (within 2%), ruling out increased computation as an explanation. We further validate that this behavior persists under longer training horizons in Appendix E.

D.2 Loss trajectories over training steps.

Across three random seeds, evaluation loss trajectories exhibit very similar shapes and timing of divergence. As a result, we report mean losses without error bars to emphasize the dynamics rather than step-wise variance. For clarity, we emphasize that unseen examples are strictly excluded from training. The reported unseen loss is computed via evaluation-only forward passes and does not influence model updates. Across all α values, training loss on unseen examples follows nearly identical trajectories throughout optimization, indicating that increasing memorization pressure does not systematically affect optimization on held-out data. In contrast, evaluation loss on seen examples diverges progressively as training proceeds: larger α values lead to faster loss reduction and lower final loss. This divergence typically becomes apparent around the middle of training. In particular, the seen–unseen loss gap begins to emerge at approximately 40–50% of the total optimization steps and becomes clearly visible by roughly 60–70% of training. We emphasize that the reported fractions are approximate and intended to characterize the stage of training rather than a precise threshold.

D.3 Evolution of the seen–unseen loss gap.

Consistent with this pattern, the gap between seen and unseen training loss remains small early in training and grows monotonically with increasing α. For $\alpha = 0.0$, the gap remains negligible throughout training, whereas higher α values induce a steadily increasing separation. This confirms that α selectively amplifies memorization pressure during training.

D.4 Training time analysis.

Finally, we measure wall-clock training time across all α values and observe no systematic variation. Because α only reweights loss components without changing model architecture, batch size, or the number of optimization steps, training time remains approximately constant across the sweep. This rules out increased computation as an explanation for the observed memorization effects.

Overall, these training-dynamics analyses provide additional diagnostic evidence that α functions as a selective memorization control rather than a general optimization or compute knob.

E Extended Training Horizon Analysis

To assess whether the memorization control induced by α persists beyond short training horizons, we conduct a small-scale extension with a longer training schedule. We retrain GPT-2 Small on ARC for 2,000 optimization steps at $\alpha \in \{0.0, 0.6, 1.0\}$, holding all other settings fixed.

Figure 10 reports seen and unseen evaluation accuracy as a function of training steps. Consistent with the results reported in the main paper, seen accuracy increases monotonically with α, while unseen accuracy remains stable throughout the extended training horizon. Notably, the separation

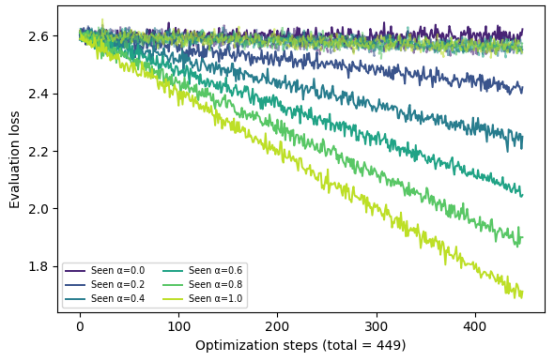

Figure 9: **Evaluation loss dynamics during training under different α (GPT-2 Small).** Evaluation loss is plotted against **optimization steps** (gradient updates). All runs are trained for a fixed total of **449 steps**. As α increases, loss on seen (training-injected) examples diverges during training, while loss on unseen (held-out) examples remains stable. The divergence between seen and unseen loss begins to emerge around the midpoint of training (approximately 200 out of 449 steps) and increases steadily thereafter.

between α values emerges early and is maintained rather than collapsing or reversing, suggesting that the effect of α reflects a stable training-time memorization control rather than a transient optimization artifact.

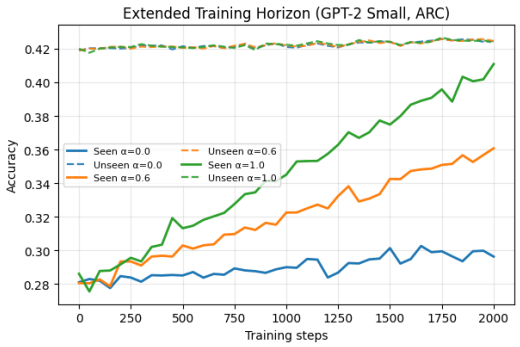

Figure 10: **Extended training horizon validation (GPT-2 Small, ARC).** Seen (solid) and unseen (dashed) accuracy as a function of training steps for $\alpha \in \{0.0, 0.6, 1.0\}$ under a longer training schedule (2,000 steps). Seen accuracy increases monotonically with α, while unseen accuracy remains stable, indicating that the memorization control induced by α persists beyond short training horizons.