EMBER: MARGIN-TARGETED ENERGY RESHAPING FOR MITIGATING EXACT MEMORIZATION IN AUTOREGRESSIVE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive language models can reproduce protected training passages verbatim, while aggressive suppression can impair ordinary generation. We study an explicit energy target for balancing these effects. EMBER applies a smooth margin penalty to the average token negative log-likelihood of each protected continuation, combined with ordinary-data likelihood training. The margin sets where the protected gradient attenuates, a temperature controls the transition width, and a separate coefficient sets its strength. Our analysis characterizes this weighting and its local curvature. Experiments on BookMIA and a controlled Wikitext-2 split cover GPT-2 full fine-tuning and Llama-3 and Qwen-3 LoRA adaptation. All methods start from the same base checkpoint within each model comparison, and leakage is evaluated on designated passages included in protected training. On BookMIA, EMBER reduces greedy EM@10 from 34.7% to 4.4% for Llama-3-8B and from 29.4% to 1.9% for Qwen-3-8B; perplexity changes from 18.4 to 18.9 and from 8.5 to 8.8, respectively. These results support energy-targeted suppression as a simple way to reduce exact continuation while retaining ordinary-corpus generation quality in the evaluated settings.

1 INTRODUCTION

Autoregressive language models can memorize training data and later reproduce protected passages verbatim (Huang et al., 2024; Chen et al., 2024; Cooper et al., 2025). In deployment, this often appears as *exact continuation*: a prompt overlaps with a copyrighted passage, and the model continues with the original text token by token (Carlini et al., 2021; Cooper et al., 2025). As models become more capable, their ability to produce long, fluent continuations makes this behavior increasingly relevant to practical copyright protection (Carlini et al., 2023; Liu et al., 2026).

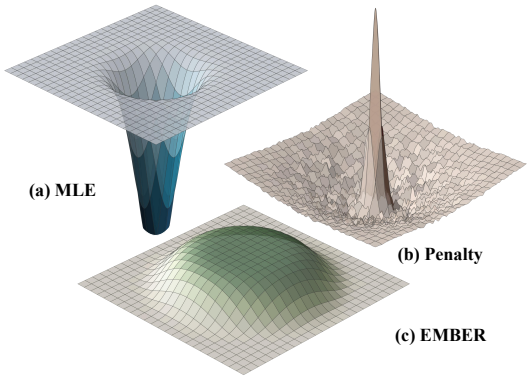


Figure 1: Schematic of protected-sequence energy under different objectives. (a) Maximum likelihood estimation (MLE) favors memorized continuations. (b) Direct suppression raises their energy. (c) EMBER introduces a target margin and attenuates repulsion above it.

Training-time mitigation offers a direct way to change the model's preference for protected continuations. Data filtering and deduplication reduce exposure (Kandpal et al., 2022), while output filtering intervenes at generation time (Deng et al., 2025). Objective modification instead acts on the parameters that encode the continuation preference (Jang et al., 2023; Chu et al., 2024; Hans et al., 2024). Its central challenge is to reduce protected-text reproduction while retaining the knowledge and language patterns needed for ordinary 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.

The shape of the suppression objective matters for this trade-off. Gradient ascent on protected-data negative log-likelihood applies persistent pressure to lower sequence probability. Probability-based suppression already weakens as a protected sequence becomes less likely, but its attenuation is tied to the current sequence probability. Neither formulation directly specifies the per-token energy level around which suppression should taper off. An explicit target offers a simple way to control this behavior for each protected example throughout training.

We propose **EMBER** (**E**xact **M**emorization **B**arrier via **E**nergy **R**eshaping). EMBER views average token negative log-likelihood as continuation energy and applies a smooth penalty to the deficit below a target margin. Protected sequences with low energy receive strong repulsion; sequences sufficiently above the margin receive little additional pressure. The ordinary-data likelihood term continues to support generation quality. Figure 1 illustrates this target-driven reshaping.

The key design choice is to make suppression depend on each continuation's progress toward a target. Examples that remain highly likely receive more pressure than those already displaced from their memorized paths, while ordinary-data learning continues. The target m, transition width τ, and weight λ separate where suppression attenuates from how strongly it competes with ordinary learning. The objective fits directly into full fine-tuning and LoRA.

We evaluate EMBER on BookMIA and a controlled Wikitext-2 split across GPT-2, Llama-3, and Qwen-3. The results show substantial reductions in exact continuation with ordinary-generation quality close to the unprotected reference, including as model size increases. Our contributions are:

- We formulate energy-targeted suppression with a standard smooth margin penalty, weighting each protected continuation by its energy deficit while retaining ordinary-data likelihood training.
- We characterize its per-example suppression mechanism, gradient attenuation, and local curvature, exposing how the energy target, transition width, and suppression weight control optimization.
- We demonstrate consistent leakage–utility gains across two data settings, three model families, and two adaptation regimes. Scaling and sensitivity studies show how the method behaves as model capacity and suppression strength change.

2 Preliminaries and the Energy Perspective

2.1 Autoregressive Continuation Energy

Classical energy-based models assign probability through $p_\theta(x) = \exp(-E_\theta(x))/Z_\theta$, with a global partition function Z_θ (LeCun et al., 2006; Du & Mordatch, 2019). Autoregressive models instead normalize locally over the vocabulary at each token. Given a prompt s and a continuation $y = (y_1, \ldots, y_L)$,

$$p_\theta(y \mid s) = \prod_{t=1}^{L} p_\theta(y_t \mid s, y_{<t}). \tag{1}$$

This factorization provides a tractable energy parameterization without introducing a separate globally normalized sequence model (Bengio et al., 2003; Bakhtin et al., 2021; Pang et al., 2024).

We define continuation energy as the average conditional token negative log-likelihood:

$$E(y \mid s; \theta) \triangleq -\frac{1}{L} \sum_{t=1}^{L} \log p_\theta(y_t \mid s, y_{<t}). \tag{2}$$

All logarithms are natural, so E is measured in nats per token. Averaging removes the direct multiplicative dependence on continuation length. Lower energy means that the reference tokens receive higher conditional probabilities on average, with

$$p_\theta(y \mid s) = \exp\big(-LE(y \mid s; \theta)\big). \tag{3}$$

The energy view lets us express a suppression target on the same per-token scale as the ordinary-data training loss.

2.2 From Memorized Continuations to an Energy Target

Let $\mathcal{O}$ and $\mathcal{C}$ be the ordinary and designated protected corpora used during adaptation. Our task is to reduce exact continuation of protected passages while retaining ordinary generation. Repeated exposure can make these passages unusually likely under their prompts (Kandpal et al., 2022; Huang et al., 2024). The motivating *energy-pit* picture is a protected continuation with low energy relative to ordinary examples,

$$E(y_c \mid s; \theta) \ll \mathbb{E}_{(s,y)\sim\mathcal{O}}[E(y \mid s; \theta)]. \tag{4}$$

This picture motivates intervention on highly favored continuation paths (Carlini et al., 2021; Cooper et al., 2025). Protection labels specify which passages to suppress; their current energies determine the suppression strength.

EMBER raises protected-continuation energy toward a chosen margin m, while fitting ordinary data by likelihood maximization. Specifying m directly in nats per token gives each protected example a common target without estimating the ordinary-data energy distribution. The energy picture thus leads to a concrete control over the training signal.

3 EMBER: Margin-Targeted Energy Reshaping

3.1 Suppression Objectives and Gradient Attenuation

Consider a protected continuation of length L and abbreviate its energy as $E = E(y_c \mid s; \theta)$. Gradient-ascent-style suppression (GA) reverses likelihood fitting on protected data (Jang et al., 2023). A second reference is direct probability suppression, denoted IR in our experiments. Their energy-space forms are:

$$\begin{aligned} \mathcal{L}_{\mathrm{GA}}(E) &= -E, \qquad \frac{\partial \mathcal{L}_{\mathrm{GA}}}{\partial E} = -1, \\ \mathcal{L}_{\mathrm{IR}}(E) &= e^{-LE}, \quad \frac{\partial \mathcal{L}_{\mathrm{IR}}}{\partial E} = -Le^{-LE}. \end{aligned} \tag{5}$$

Both objectives push protected energy upward. GA maintains a constant scalar gradient, whereas IR attenuates its gradient as probability falls. For fixed L and $E \geq 0$, both scalar gradients are bounded; the key design question is where and how rapidly suppression should attenuate.

EMBER introduces an independently specified location and width for attenuation. In IR, the factor e^{-LE} ties the gradient decay to both energy and sequence length. A global loss weight rescales that gradient but leaves its decay rate unchanged. EMBER instead places the transition at a chosen per-token energy m and controls its sharpness through τ. This allows low-energy examples to receive substantial pressure while examples well above the target contribute progressively less.

3.2 A Smooth Margin Penalty

EMBER penalizes the energy deficit below the target using

$$\mathcal{R}_{\mathrm{EMBER}}(\theta; y_c, s, m, \tau) = \tau \log\left(1 + \exp\left(\frac{m - E(y_c \mid s; \theta)}{\tau}\right)\right), \tag{6}$$

where $\tau > 0$. Writing the penalty as $r(E)$ gives the equivalent form $r(E) = -\tau \log \sigma((E-m)/\tau)$, a standard log-sigmoid loss. It smoothly approximates the hinge $(m - E)_+$: below m it is approximately $m - E$, and above m it decays exponentially. The margin and temperature specify the location and width of this transition on the average-token NLL scale.

We combine the protected penalty with likelihood training on ordinary data:

$$\begin{aligned} \mathcal{L}_{\mathrm{total}}(\theta) = \mathbb{E}_{(s_o,y_o)\sim\mathcal{O}}[E(y_o \mid s_o; \theta)] \\ + \lambda \mathbb{E}_{(s_c,y_c)\sim\mathcal{C}}[\mathcal{R}_{\mathrm{EMBER}}(\theta; y_c, s_c, m, \tau)]. \end{aligned} \tag{7}$$

The weight $\lambda \geq 0$ balances protected-data suppression against ordinary-data fitting. Changing m shifts which examples receive strong suppression, whereas changing λ scales their contribution. The objective uses teacher-forced token log-probabilities already available during training and adds no inference-time component. Appendix B gives the training procedure.

The margin also has a direct scale interpretation: $E \geq m$ corresponds to a geometric mean reference-token probability at most e^{-m}. We use $m = 1$ as a shared reference configuration in the main experiments. Length averaging makes this choice independent of a direct length multiplier, while model calibration and tokenization can still affect the energy scale. Section 4.4 examines how changing this target affects leakage and utility.

3.3 How the Margin Controls Repulsion

Differentiating the protected penalty gives

$$\frac{\partial \mathcal{R}_{\mathrm{EMBER}}}{\partial E} = -\sigma\left(\frac{m - E}{\tau}\right), \tag{8}$$

where σ is the logistic sigmoid. Well below m, the scalar gradient is close to -1, producing strong upward pressure on energy. At $E = m$, it equals $-1/2$. Above the margin it smoothly approaches zero, with the following exact attenuation rule for $0 < \epsilon < 1/2$:

$$\left|\frac{\partial \mathcal{R}_{\mathrm{EMBER}}}{\partial E}\right| \leq \epsilon \quad \Longleftrightarrow \quad E \geq m + \tau \log \frac{1-\epsilon}{\epsilon}. \tag{9}$$

For example, with $m = 1$ and $\tau = 0.05$, the scalar gradient magnitude falls below 0.01 at approximately $E = 1.23$. The attenuation rule quantifies how far above the target an example must move to receive little additional pressure, giving τ an operational meaning as the transition width.

The parameter gradient follows by the chain rule, $\nabla_\theta \mathcal{R}_{\mathrm{EMBER}} = -\sigma((m - E)/\tau)\nabla_\theta E$. Thus, EMBER reweights each protected example's energy gradient according to its distance from the target. Appendix A derives the attenuation rule and the local smoothness bound $L_R \leq L_E + G^2/(4\tau)$ when the energy is L_E-smooth and its gradient norm is bounded by G. The bound explains the curvature cost of sharpening the transition.

Equation 3 translates an achieved energy margin into $p_\theta(y_c \mid s) \leq e^{-Lm}$. This full-sequence bound does not by itself bound greedy EM@10, which depends on the first ten tokens and their rankings against alternatives. We use energy as a differentiable suppression signal and evaluate generated continuations directly. Appendix A details the probability interpretation and its scope.

3.4 Adaptive Suppression at the Example Level

Writing the scalar margin penalty as $r(E)$, the soft margin has a useful variational interpretation. Let $H(w) = -w \log w - (1 - w)\log(1 - w)$ be binary entropy, with $0 \log 0 = 0$. Then

$$r(E) = \max_{0 \leq w \leq 1} \{w(m - E) + \tau H(w)\}, \qquad w^*(E) = \sigma((m - E)/\tau). \tag{10}$$

EMBER thus assigns each protected example an adaptive coefficient for its energy deficit. Before averaging, its loss gradient is $-\lambda w^*(E)\nabla_\theta E$, combined with the ordinary-data gradient. The margin controls which examples receive substantial pressure; λ bounds their scalar coefficients; and the entropy term smooths the transition. These weights are recomputed from the current model, so an example receives renewed pressure if later updates make its protected continuation more likely again. Appendix A.2 derives this representation.

The penalty is applied *before* averaging over protected examples. Applying it to the batch mean instead would let high-energy continuations compensate for low-energy ones. For example, at the default $m = 1$ and $\tau = 0.05$, two continuations with energies 0.77 and 1.23 receive weights approximately 0.99 and 0.01. A penalty on their mean energy would assign both the coefficient 0.5. Per-example weighting therefore directs substantially more pressure toward the continuation that remains below the target.

Because protected and ordinary text share model parameters, continued suppression of already displaced continuations can also affect ordinary generation. Adaptive attenuation reduces this residual pressure while allowing ordinary-data fitting to continue. It provides a local alternative to one global stopping time, using only a scalar softplus per continuation and no reference model.

Model	Method	BookMIA Configuration				Wikitext Controlled Setup			
		EM@10 (%, ↓)	ROUGE-L (%, ↓)	PPL (↓)	MAUVE (↑)	EM@10 (%, ↓)	ROUGE-L (%, ↓)	PPL (↓)	MAUVE (↑)
GPT-2-124M	SFT	$3.2_{\pm 1.0}$	$27.9_{\pm 1.7}$	$22.6_{\pm 1.5}$	$0.85_{\pm 0.02}$	$2.5_{\pm 1.0}$	$22.1_{\pm 1.4}$	$3.8_{\pm 0.4}$	$0.95_{\pm 0.01}$
	IR	$0.6_{\pm 0.4}$	$10.8_{\pm 1.9}$	$93.4_{\pm 9.8}$	$0.49_{\pm 0.03}$	$0.2_{\pm 0.3}$	$6.8_{\pm 0.7}$	$52.1_{\pm 0.8}$	$0.51_{\pm 0.02}$
	Goldfish	$2.1_{\pm 0.7}$	$19.3_{\pm 1.3}$	$27.5_{\pm 2.4}$	$0.80_{\pm 0.02}$	$1.8_{\pm 0.7}$	$18.4_{\pm 1.0}$	$5.0_{\pm 0.6}$	$0.92_{\pm 0.02}$
	CR	$0.9_{\pm 0.3}$	$13.9_{\pm 0.8}$	$32.8_{\pm 4.1}$	$0.74_{\pm 0.03}$	$0.8_{\pm 0.1}$	$12.5_{\pm 0.5}$	$4.5_{\pm 0.5}$	$0.88_{\pm 0.03}$
	EMBER	$1.1_{\pm 0.2}$	$12.7_{\pm 0.6}$	$24.0_{\pm 1.6}$	$0.84_{\pm 0.02}$	$0.0_{\pm 0.1}$	$10.2_{\pm 0.3}$	$4.1_{\pm 0.3}$	$0.92_{\pm 0.01}$
Llama-3-8B	SFT	$34.7_{\pm 1.5}$	$50.2_{\pm 1.8}$	$18.4_{\pm 1.0}$	$0.94_{\pm 0.01}$	$31.0_{\pm 1.2}$	$47.3_{\pm 1.5}$	$3.5_{\pm 0.2}$	$0.97_{\pm 0.01}$
	IR	$1.0_{\pm 0.4}$	$9.2_{\pm 2.7}$	$170.5_{\pm 15.4}$	$0.41_{\pm 0.02}$	$0.5_{\pm 0.5}$	$5.4_{\pm 0.8}$	$76.8_{\pm 10.6}$	$0.38_{\pm 0.02}$
	Goldfish	$6.5_{\pm 1.1}$	$27.4_{\pm 1.3}$	$22.3_{\pm 3.3}$	$0.93_{\pm 0.02}$	$2.6_{\pm 0.9}$	$16.2_{\pm 1.1}$	$3.8_{\pm 0.4}$	$0.91_{\pm 0.02}$
	CR	$9.7_{\pm 0.3}$	$28.6_{\pm 0.7}$	$26.3_{\pm 4.5}$	$0.85_{\pm 0.02}$	$2.8_{\pm 0.2}$	$16.3_{\pm 0.6}$	$3.7_{\pm 0.2}$	$0.89_{\pm 0.01}$
	EMBER	$4.4_{\pm 0.2}$	$17.9_{\pm 0.4}$	$18.9_{\pm 2.1}$	$0.93_{\pm 0.01}$	$0.3_{\pm 0.1}$	$12.5_{\pm 0.4}$	$3.0_{\pm 0.4}$	$0.95_{\pm 0.01}$
Qwen-3-8B	SFT	$29.4_{\pm 1.5}$	$49.8_{\pm 1.7}$	$8.5_{\pm 0.3}$	$0.91_{\pm 0.01}$	$25.3_{\pm 1.3}$	$45.1_{\pm 1.6}$	$3.3_{\pm 0.2}$	$0.93_{\pm 0.01}$
	IR	$0.2_{\pm 0.5}$	$4.7_{\pm 1.2}$	$201.3_{\pm 10.8}$	$0.34_{\pm 0.03}$	$0.8_{\pm 0.4}$	$13.1_{\pm 0.7}$	$94.2_{\pm 6.7}$	$0.32_{\pm 0.02}$
	Goldfish	$6.8_{\pm 0.9}$	$29.7_{\pm 1.2}$	$9.4_{\pm 0.4}$	$0.87_{\pm 0.02}$	$2.2_{\pm 0.8}$	$27.5_{\pm 1.1}$	$3.6_{\pm 0.3}$	$0.85_{\pm 0.02}$
	CR	$3.6_{\pm 0.3}$	$25.8_{\pm 0.8}$	$11.2_{\pm 0.7}$	$0.84_{\pm 0.02}$	$2.5_{\pm 0.1}$	$24.2_{\pm 0.5}$	$3.4_{\pm 0.1}$	$0.84_{\pm 0.01}$
	EMBER	$1.9_{\pm 0.2}$	$18.2_{\pm 0.4}$	$8.8_{\pm 0.3}$	$0.90_{\pm 0.01}$	$0.6_{\pm 0.1}$	$14.8_{\pm 0.3}$	$3.3_{\pm 0.1}$	$0.90_{\pm 0.01}$

Table 1: Main results: greedy EM@10 and ROUGE-L on designated protected training passages, and PPL and MAUVE on ordinary data. Entries are means over five random seeds; $\pm$ denotes standard deviation. Bold identifies EMBER.

4 Experiments

We evaluate suppression and ordinary generation jointly, examine their balance as model capacity increases, and study how the three controls change the operating point.

4.1 Experimental Setup

Models and Baselines.

The main comparison uses GPT-2-124M (Radford et al., 2019), Llama-3-8B (Grattafiori et al., 2024), and Qwen-3-8B (Yang et al., 2025). GPT-2 is fully fine-tuned; Llama-3 and Qwen-3 base checkpoints use LoRA (Hu et al., 2022). Scaling covers GPT-2 at 124M, 355M, 774M, and 1.5B, and Qwen-3 at 0.6B, 1.7B, 4B, and 8B, with the same adaptation regime within each family. Training budgets and adaptation settings are shared within each model comparison; hyperparameters are selected on a validation split.

We compare supervised fine-tuning (**SFT**) on ordinary and protected text; inverse regularization (**IR**), the sequence-probability suppression in Equation 5; **Goldfish** (Hans et al., 2024), which masks token losses while retaining the full conditioning history; and **Copyright Regression (CR)**, adapted from the inverse-loss regularization of Chu et al. (2024). EMBER uses $(m = 1, \tau = 0.05, \lambda = 1)$ in the main and scaling comparisons. Appendix D.3 distinguishes the baseline objectives and their normalization.

Datasets.

BookMIA (Shi et al., 2023) supplies a book-continuation setting: its designated pre-2023 member books form $\mathcal{C}$, and post-2023 books form $\mathcal{O}$. These labels define corpus roles here, rather than certifying pretraining membership for every evaluated checkpoint.

We randomly partition Wikitext-2 (Merity et al., 2017) into disjoint protected and ordinary subsets with a 2:8 ratio. This controlled setting tests suppression when both corpora share similar content and language patterns.

Training Protocol and Evaluation.

All methods start independently from the same base checkpoint for a given model. SFT performs likelihood training on $\mathcal{O} \cup \mathcal{C}$; EMBER fits $\mathcal{O}$ and applies its margin penalty to $\mathcal{C}$. The reference passages underlying the 1,000 leakage-evaluation prompts also occur in protected training. Thus, the

comparison measures suppression of designated training passages through alternative adaptations of a common initialization. Appendix D.1 describes this data flow.

We use AdamW with learning rate 5×10^{-5} and five random seeds, training for 10 epochs with batch size 32 under mixed precision. LoRA uses rank $r = 16$ and scaling $\alpha = 32$. Each leakage prompt is a 40-token prefix. Adapting Carlini et al. (2023), **EM@10** measures the percentage of greedy continuations whose first ten tokens exactly match the reference. **ROUGE-L** (Lin, 2004) measures residual subsequence overlap. EM in the figures and sensitivity table denotes EM@10.

For utility, **PPL** (Jurafsky & Martin, 2026) measures ordinary-text likelihood, and **MAUVE** (Pillutla et al., 2021) compares generated and ordinary reference continuations. Lower PPL and higher MAUVE indicate better ordinary-generation quality. Greedy decoding and evaluation prompts are matched across methods. Main-table entries report means and standard deviations over five seeds; Appendix D details the evaluation.

4.2 Main Results

Table 1 shows that EMBER substantially reduces BookMIA memorization relative to SFT while retaining ordinary-generation quality. For Llama-3-8B, EM falls from 34.7% to 4.4% and ROUGE-L from 50.2% to 17.9%, while PPL changes from 18.4 to 18.9 and MAUVE from 0.94 to 0.93. For Qwen-3-8B, EM drops from 29.4% to 1.9%, with PPL changing from 8.5 to 8.8 and MAUVE from 0.91 to 0.90. GPT-2-124M begins with lower memorization, but EMBER still reduces both leakage metrics while keeping utility close to SFT.

These Llama-3 and Qwen-3 results correspond to 87.3% and 93.5% relative reductions in EM, with PPL increases of 2.7% and 3.5%, respectively, calculated from the reported means. At the reported configurations, IR reaches lower BookMIA EM, but its PPL rises to 93.4–201.3 and MAUVE falls to 0.34–0.49, illustrating why suppression and utility must be assessed together.

The controlled Wikitext-2 split reinforces this pattern. EMBER reaches EM values of 0.0%, 0.3%, and 0.6% for GPT-2, Llama-3, and Qwen-3, respectively, while retaining low PPL and MAUVE close to SFT.

The reported means show consistent gains over Goldfish and CR. In all six model–dataset combinations, EMBER has lower EM, ROUGE-L, and PPL than Goldfish, with equal or higher MAUVE. Against CR, it improves ROUGE-L, PPL, and MAUVE in every setting and reduces EM in five of six comparisons. On GPT-2/BookMIA, CR has slightly lower EM (0.9% versus 1.1%), while EMBER retains better utility and lower overall overlap.

Across all six settings, EMBER's MAUVE remains within 0.03 of SFT, while PPL is at most 7.9% higher, calculated from the reported means. Both leakage metrics improve alongside retained utility.

4.3 Scalability Across Model Sizes

We test whether mitigation remains effective as memorization increases with model capacity, holding the adaptation regime fixed within each family.

Figure 2 shows increasing exact continuation after SFT in the controlled split. For GPT-2, EM rises from 2.5% at 124M to 99.1% at GPT-2 XL; for Qwen-3, it increases from 2.9% at 0.6B to 25.3% at 8B. Although absolute levels differ by family and adaptation regime, the upward trend is consistent.

The GPT-2 series tests full-parameter suppression up to near-saturated EM; Qwen-3 tests the same objective under restricted LoRA updates in a modern model family.

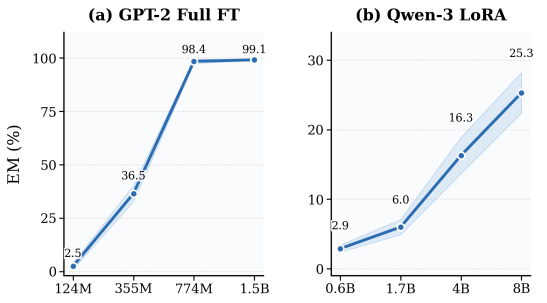


Figure 2: Memorization after SFT on ordinary and protected text. As model size increases, EM@10 rises for both GPT-2 full fine-tuning and Qwen-3 LoRA adaptation.

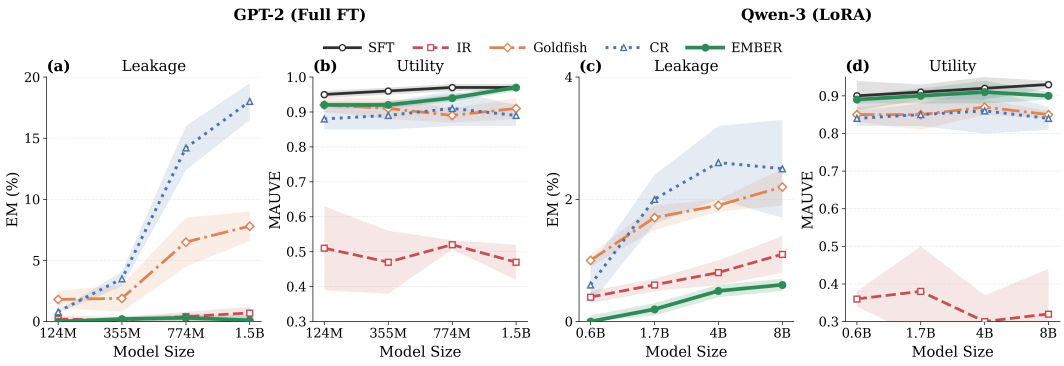


Figure 3: Cross-scale comparison of mitigation methods in the controlled Wikitext setting. Panels (a,b) show GPT-2 with full fine-tuning, and panels (c,d) show Qwen-3 with LoRA adaptation. SFT is omitted from the leakage panels for readability. EM denotes greedy EM@10; shaded regions show variability across random seeds.

Figure 3 compares mitigation methods across model scales in the controlled Wikitext setting. We focus on EM and MAUVE because they capture the key trade-off between leakage reduction and utility, and omit the SFT curve from the leakage panels for readability. EMBER remains effective as model capacity increases in both adaptation regimes: on GPT-2, it keeps EM at or near zero across the full scale range while maintaining MAUVE between 0.92 and 0.97. IR also drives leakage to near zero, but with MAUVE only around 0.47 to 0.52. Goldfish and CR preserve utility better than IR, but their leakage grows noticeably with model size.

For Qwen-3 under LoRA adaptation, EMBER keeps EM at or below 0.6% across all scales while maintaining MAUVE around 0.89 to 0.91, whereas Goldfish and CR show visible leakage growth and IR shows a severe utility drop. EMBER thus maintains low leakage and high utility throughout the evaluated scale range.

The shared configuration sustains this balance across the tested sizes without a separate EMBER setting for each scale, even as unprotected exact continuation rises sharply.

4.4 Controlling the Leakage–Utility Trade-off

Table 2 varies one parameter at a time around $(m = 1, \tau = 0.05, \lambda = 1)$ on Qwen-3-8B with LoRA, using the same training and greedy-decoding protocol. It describes local operating points around the shared configuration.

m	τ	λ	BookMIA		Wikitext-2	
			EM ↓	MAUVE ↑	EM ↓	MAUVE ↑
0.5	0.05	1.0	4.7	0.91	1.4	0.92
1.0	0.05	1.0	**1.9**	**0.90**	**0.6**	**0.90**
1.5	0.05	1.0	1.1	0.89	0.3	0.89
2.0	0.05	1.0	0.6	0.86	0.1	0.87
1.0	0.02	1.0	1.6	0.89	0.5	0.90
1.0	0.10	1.0	2.4	0.90	0.9	0.91
1.0	0.05	0.5	3.3	0.91	1.1	0.92
1.0	0.05	2.0	0.9	0.88	0.3	0.89

Table 2: Qwen-3-8B LoRA sensitivity on BookMIA and controlled Wikitext-2. EM denotes EM@10 in percent; MAUVE is unitless. Bold marks the shared configuration.

The margin produces a clear progression. Raising m from 0.5 to 1 reduces BookMIA EM from 4.7% to 1.9%, with MAUVE changing from 0.91 to 0.90. Raising it further to 2 lowers EM to 0.6% at MAUVE 0.86. Wikitext-2 follows the same pattern. The default $m = 1$ captures a substantial reduction in exact continuation while retaining ordinary-generation quality; larger targets offer stronger suppression at a greater utility cost.

The temperature has a smaller effect in the tested range: across $\tau \in \{0.02, 0.05, 0.10\}$, BookMIA EM varies from 1.6% to 2.4% and MAUVE from 0.89 to 0.90. The weight λ moves the operating point more strongly. Increasing it from 0.5 to 2 lowers BookMIA EM from 3.3% to 0.9%, with MAUVE moving from 0.91 to 0.88. Both datasets show the same ordering. These trends agree with the objective design: the target sets where pressure attenuates, the temper-

Method	Continuation Output
Prompt	*Goldstein had fled and was hiding no one knew where, and of the others, a few had simply disappeared, while the majority had been executed after spectacular public trials at which they made confession of their crimes.*
Reference	Among the last survivors were three men named **Jones, Aaronson, and Rutherford**. It must have been in **1965** that these three had been **arrested**. As often happened, they had **vanished for a year or more**, and then had suddenly been brought forth to **incriminate themselves**...
SFT	Among the last survivors were three men named **Jones, Aaronson, and Rutherford**. It was in **1965** that they were **arrested**. As usually happened, they **vanished for a year or more** before being brought back to **incriminate themselves**.
Goldfish	Among the few remaining survivors were men like **Jones** and **Rutherford**. They had **vanished for a long time** and then resurfaced at public trials to confess and admit to crimes against the state.
CR	Among the last survivors were three men named **Jones, Aaronson, and Rutherford**. It was in **1965** that they were **arrested**. *They vanished for a year. They vanished, and then they were brought forth to* **incriminate themselves**.
IR	He could not hold on to any particular names. *The Party watched everything. It watched the streets and the people, and the noise of the telescreen filled the room. It was always there, always watching.*
EMBER	The records of these trials were heavily broadcast on the telescreens, serving as a warning to anyone who might harbor rebellious thoughts. Winston remembered watching the footage, feeling a deep sense of unease as the condemned men wept and begged for mercy.

Table 3: Representative BookMIA continuation from Qwen-3-8B with LoRA. All methods use the same protected prefix and greedy decoding. **Red bold text** highlights retained names, facts, and phrases from the reference. *Blue italic text* highlights repetition or stilted phrasing. These annotations illustrate the example qualitatively.

ature sets how sharply, and the weight sets its strength relative to ordinary-data fitting. Appendix C provides the detailed analysis.

Although larger m and λ both lower leakage in Table 2, their gradient effects differ. Increasing λ rescales all protected coefficients equally. Increasing m mainly changes weights near the transition, while those far below the target remain near one. This distinguishes their analytical roles; the one-at-a-time study does not isolate parameter interactions or establish which mechanism causes the empirical gains.

Across all eight configurations, BookMIA EM ranges from 0.6% to 4.7% versus 29.4% for SFT, with MAUVE between 0.86 and 0.91. Wikitext-2 EM ranges from 0.1% to 1.4% versus 25.3%, with MAUVE between 0.87 and 0.92. These results support useful behavior over the tested neighborhood, rather than only at one parameter setting.

The shared configuration occupies an intermediate region of the observed trade-off. Higher margins favor lower leakage, while weaker suppression favors MAUVE. For deployment, a practical selection criterion is to compare leakage among configurations that meet an ordinary-generation quality requirement on validation data. This criterion must be applied consistently across competing objectives.

4.5 Case Studies

Table 3 compares Qwen-3-8B LoRA continuations from George Orwell's *1984* under the same protected prefix and greedy decoding. The reference contains distinctive names and events, making copied details and alternative continuations easy to inspect.

SFT closely follows the protected passage. Goldfish changes wording but retains names and events; CR preserves details and introduces repetition. IR avoids the named entities but produces repetitive text. EMBER gives a coherent alternative about the trials without repeating the reference names

and event sequence. This qualitative example complements the aggregate overlap and ordinary-generation metrics.

Scope of the evaluation. The results concern fixed-prefix greedy continuation of designated training passages and generation quality on ordinary data. They do not establish suppression on unseen protected texts, resistance to alternative extraction strategies, or preservation of capabilities beyond PPL and MAUVE. Tokenization changes the text span covered by a fixed token count, so absolute EM values across model families require care. Comparisons within a model share its tokenizer and evaluation prompts.

5 Related Work

Memorization and extraction. Training-data extraction studies show that language models can emit rare verbatim strings, with exposure frequency, model capacity, and prompt context influencing extractability (Carlini et al., 2021; Kandpal et al., 2022; Carlini et al., 2023; Yu et al., 2023). Recent work examines verbatim recall, continuation behavior, and model attribution in larger language models (Huang et al., 2024; Chen et al., 2024; Menta et al., 2025; Cooper et al., 2025). BookMIA originates from membership inference (Shi et al., 2023); here its text partition supplies protected and ordinary corpora for a continuation-generation task. EMBER addresses the model's tendency to follow protected continuation paths during generation.

Training objectives for suppression and retention. Gradient-ascent unlearning reverses the learning signal on designated data (Jang et al., 2023). Copyright Regression combines fitting with an inverse-loss regularizer (Chu et al., 2024), while Goldfish masks token losses to interrupt sequence memorization (Hans et al., 2024). More recent approaches examine utility-aware memorization mitigation and alignment against verbatim reproduction (Sakarvadia et al., 2025; Chen et al., 2025). These methods motivate joint evaluation of suppression and retained generation quality.

Negative Preference Optimization (NPO) uses a smooth objective to moderate forgetting updates, with suppression defined relative to a reference model's likelihood (Zhang et al., 2024). EMBER specifies an absolute target on average token negative log-likelihood. Its margin and temperature directly set the location and width of gradient attenuation, without a reference-model forward pass. This design turns a chosen energy target into an adaptive allocation of suppression across examples, implemented through a standard softplus penalty. Our analysis and experiments connect this allocation to the observed leakage–utility trade-off.

Energy-based perspectives. Energy-based learning supplies a general language for expressing preferences over outputs (LeCun et al., 2006; Belanger & McCallum, 2016; Du & Mordatch, 2019). Residual energy models and energy-guided text generation combine energy functions with language-model likelihoods for alignment or control (Bakhtin et al., 2021; Qin et al., 2022; Liu et al., 2023; Pang et al., 2024). EMBER uses continuation NLL itself as the energy, retaining the model's local softmax normalization and ordinary decoding. This perspective makes the protected-data target interpretable on a per-token scale and connects it directly to the adaptive weighting mechanism in Section 3.4.

6 Conclusion

We presented EMBER, a margin-targeted objective for suppressing exact continuation of designated protected passages. A standard smooth penalty assigns pressure according to each continuation's energy deficit, with separate controls for target location, transition width, and suppression strength. From a shared base initialization, BookMIA and controlled Wikitext-2 comparisons show reduced greedy EM@10 alongside retained ordinary-generation quality across the evaluated model families and scales. The analysis explains the resulting gradient allocation, while sensitivity results describe its leakage–utility operating points. EMBER uses teacher-forced losses in full fine-tuning or LoRA and introduces no inference-time component. Its explicit target makes the suppression setting interpretable, while the paired leakage and quality measurements expose the cost of choosing a stronger target.

AI USE STATEMENT

We used generative AI tools to improve readability, wording, and organization; assist manuscript review; clarify mathematical exposition and the description of the experimental protocol; and discuss the interpretation of reported results. We also used these tools to identify and locate related work. These tools were not used to generate experimental data or carry out model training and evaluation. The authors take responsibility for verifying the mathematical arguments, experimental descriptions, references, and final manuscript, including all AI-assisted text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

Section 4 describes the models, data constructions, training settings, and evaluation metrics. Appendix A provides the objective analysis, Appendix B gives the training algorithm, Appendix C expands the sensitivity analysis, and Appendix D consolidates implementation and evaluation details.

REFERENCES

Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. Residual energy-based models for text. *Journal of Machine Learning Research*, 22(40):1–41, 2021.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 983–992. PMLR, 2016.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

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. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Bowen Chen, Namgi Han, and Yusuke Miyao. A multi-perspective analysis of memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11190–11209, 2024.

Tong Chen, Faeze Brahman, Jiacheng Liu, Niloofar Mireshghallah, Weijia Shi, Pang Wei Koh, Luke Zettlemoyer, and Hannaneh Hajishirzi. Parapo: Aligning language models to reduce verbatim reproduction of pre-training data. *arXiv preprint arXiv:2504.14452*, 2025.

Timothy Chu, Zhao Song, and Chiwun Yang. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17871–17879, 2024.

A Feder Cooper, Aaron Gokaslan, Ahmed Ahmed, Amy B Cyphert, Christopher De Sa, Mark A Lemley, Daniel E Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

Zhijie Deng, Chris Yuhao Liu, Zirui Pang, Xinlei He, Lei Feng, Qi Xuan, Zhaowei Zhu, and Jiaheng Wei. Guard: Generation-time llm unlearning via adaptive restriction and detection. *arXiv preprint arXiv:2505.13312*, 2025.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, et al. Be like a goldfish, don't memorize! mitigating memorization in generative llms. *Advances in Neural Information Processing Systems*, 37:24022–24045, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, 2024.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 2026.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 10697–10707. PMLR, 2022.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting Structured Data*. MIT Press, 2006.

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

Xin Liu, Muhammad Khalifa, and Lu Wang. Bolt: Fast energy-based controlled text generation with tunable biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 186–200. Association for Computational Linguistics, 2023.

Xinyue Liu, Niloofar Mireshghallah, Jane C Ginsburg, and Tuhin Chakrabarty. Alignment whack-a-mole: Finetuning activates verbatim recall of copyrighted books in large language models. *arXiv preprint arXiv:2603.20957*, 2026.

Tarun Ram Menta, Susmit Agrawal, and Chirag Agarwal. Analyzing memorization in large language models through the lens of model attribution. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10661–10689. Association for Computational Linguistics, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Bo Pang, Caiming Xiong, and Yingbo Zhou. Arm: Alignment with residual energy-based model. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8225–8236. Association for Computational Linguistics, 2024.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Mansi Sakarvadia, Aswathy Ajith, Arham Mushtaq Khan, Nathaniel C Hudson, Caleb Geniesse, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W Mahoney. Mitigating memorization in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

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.

Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 40306–40320. PMLR, 2023.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *The First Conference on Language Modeling*, 2024.

APPENDIX

A OBJECTIVE ANALYSIS AND PROBABILITY INTERPRETATION

We analyze how EMBER's margin and temperature shape the protected-data gradient, derive the resulting local curvature bound, and relate continuation energy to sequence and prefix probabilities.

Throughout this appendix, $m > 0$, $\tau > 0$, and $\lambda \geq 0$; these hyperparameters are held fixed when differentiating with respect to model parameters. Each continuation contains $L \geq 1$ scored tokens with positive conditional probabilities, so its energy is finite and nonnegative. The sequence probability denotes the probability of these L tokens under the supplied prompt; a termination token is included only when it is part of the scored continuation. Vector norms are Euclidean and matrix norms are the induced spectral norms. For LoRA, θ denotes the trainable adapter parameters, with the pretrained weights held fixed; the same chain-rule identities apply in these coordinates.

A.1 BARRIER SHAPE AND GRADIENT ATTENUATION

Fix a protected prompt–continuation pair and write $e(\theta) = E(y_c \mid s; \theta)$ and $r(e) = \tau \log(1 + \exp((m - e)/\tau))$. The scalar derivatives are

$$\begin{aligned} r'(e) &= -\sigma\left(\frac{m-e}{\tau}\right), \\ r''(e) &= \frac{1}{\tau}\sigma\left(\frac{m-e}{\tau}\right)\left(1 - \sigma\left(\frac{m-e}{\tau}\right)\right) \leq \frac{1}{4\tau}. \end{aligned} \tag{11}$$

The curvature is largest at $e = m$, where $r'(e) = -1/2$. Thus, m locates the transition and τ controls its sharpness. The standard softplus inequalities also give

$$(m - e)_+ \leq r(e) \leq (m - e)_+ + \tau \log 2, \tag{12}$$

To verify Equation 12, set $a = (m - e)/\tau$ and use $\log(1 + e^a) = \max(a, 0) + \log(1 + e^{-|a|})$. The second term lies in $[0, \log 2]$, giving a uniform approximation error at most $\tau \log 2$. In particular, the loss approaches a hinge as $\tau \downarrow 0$. Its derivative tends to -1 for $e < m$ and to 0 for $e > m$, while remaining $-1/2$ at $e = m$.

On the physical domain $e \geq 0$, the penalty satisfies

$$0 < r(e) \leq r(0) = \tau \log(1 + e^{m/\tau}). \tag{13}$$

The lower endpoint is approached as $e \to \infty$. Thus, the margin centers the attenuation of a finite penalty: it is neither a singular barrier nor a finite minimizer of the protected term alone. The ordinary-data objective and the attenuating protected gradient jointly determine the fitted model.

For $e \geq m$, applying $\log(1 + u) \leq u$ gives

$$r(e) \leq \tau e^{-(e-m)/\tau}, \qquad |r'(e)| \leq e^{-(e-m)/\tau}. \tag{14}$$

More precisely, for $0 < \epsilon < 1/2$,

$$|r'(e)| \leq \epsilon \iff \frac{1}{1 + \exp((e-m)/\tau)} \leq \epsilon \iff e \geq m + \tau \log \frac{1-\epsilon}{\epsilon}.$$

This proves Equation 9. For the default $\tau = 0.05$, reaching $e = m + 0.230$ reduces the scalar gradient magnitude to approximately 0.01. The penalty remains differentiable throughout this transition.

The parameter-space gradient is

$$\nabla_\theta \mathcal{R}_{\mathrm{EMBER}} = -\sigma\left(\frac{m-e}{\tau}\right) \nabla_\theta e. \tag{15}$$

If $\|\nabla_\theta e\| \leq G$ in the region of interest, then

$$\|\nabla_\theta \mathcal{R}_{\mathrm{EMBER}}\| \leq G\sigma\left(\frac{m-e}{\tau}\right).$$

Consequently, an example satisfying the attenuation threshold contributes a protected-data gradient of norm at most $\lambda G\epsilon$ before averaging. This is the operational effect of the margin: it progressively reduces that example's influence on the combined update while ordinary-data likelihood training continues.

For comparison, the reference losses in Equation 5 have scalar curvatures $\mathcal{L}''_{\mathrm{GA}}(e) = 0$ and $\mathcal{L}''_{\mathrm{IR}}(e) = L^2 e^{-Le}$. IR therefore also has decaying gradient and curvature. EMBER's distinct control is the placement of its transition at m with width τ, independently of the explicit sequence-length multiplier in IR.

A.2 Adaptive Weights and Per-Example Targets

Equation 10 follows from the standard variational representation of softplus. For a fixed energy e, define $g(w) = w(m-e) + \tau H(w)$. On $0 < w < 1$,

$$g'(w) = m - e + \tau \log \frac{1-w}{w}, \qquad g''(w) = -\frac{\tau}{w(1-w)} < 0.$$

The derivative tends to $+\infty$ as $w \downarrow 0$ and to $-\infty$ as $w \uparrow 1$. Together with strict concavity, this gives the unique maximizer $w^* = \sigma((m-e)/\tau)$. To verify the optimized value, let $a = (m-e)/\tau$ and $Z = 1 + e^a$. Then $\log w^* = a - \log Z$ and $\log(1-w^*) = -\log Z$, so $H(w^*) = \log Z - w^* a$. Hence $g(w^*) = \tau \log Z = r(e)$.

Differentiating $g(w^*(e), e)$ gives $\partial_e g + \partial_w g \, \partial_e w^* = -w^*$, since $\partial_w g = 0$ at the interior maximizer. This explains why the gradient contains w^* without an additional derivative-of-weight term. Training evaluates softplus directly and does not require a separate inner optimization. Multiplying the energy deficit by a sigmoid while omitting the entropy term would be a different objective with a different derivative.

The scalar convexity $r''(e) \geq 0$ also gives, for a protected batch of size B,

$$\frac{1}{B}\sum_{i=1}^{B} r(e_i) \geq r\left(\frac{1}{B}\sum_{i=1}^{B} e_i\right). \tag{16}$$

At the parameter level, writing $\bar{e} = B^{-1}\sum_i e_i$ makes the averaging factors explicit:

$$\begin{aligned} \nabla_\theta \left(\frac{1}{B}\sum_i r(e_i)\right) &= -\frac{1}{B}\sum_i w^*(e_i)\nabla_\theta e_i, \\ \nabla_\theta r(\bar{e}) &= -\frac{w^*(\bar{e})}{B}\sum_i \nabla_\theta e_i. \end{aligned} \tag{17}$$

EMBER assigns a separate scalar weight to each energy gradient. A penalty on the batch mean assigns a common weight, so a high mean can attenuate every contribution while some continuations remain below the target. For finite energies, $r''(e) > 0$; equality in Equation 16 holds exactly when all batch energies are equal. This is a statement about loss aggregation and gradient allocation, rather than an ordering of the final models produced by the two objectives.

The three controls can also be distinguished algebraically. With $w = w^*(e)$,

$$\frac{\partial w}{\partial m} = \frac{w(1-w)}{\tau}, \qquad \frac{\partial w}{\partial e} = -\frac{w(1-w)}{\tau}, \qquad \frac{\partial w}{\partial \tau} = -\frac{m-e}{\tau^2} w(1-w). \tag{18}$$

Increasing m increases every finite-energy weight, most strongly near the transition. Increasing τ moves the weights toward $1/2$ on both sides of m, broadening the transition. Increasing λ multiplies the coefficients λw uniformly. These identities explain the distinct controls examined in Table 2.

A.3 Local Smoothness

Lemma A.1 (Local smoothness of the margin penalty)**.** *Suppose $e(\theta)$ is twice continuously differentiable with Hessian norm at most L_E and $\|\nabla_\theta e(\theta)\| \leq G$ on an open convex neighborhood Ω. Then $\mathcal{R}_{\mathrm{EMBER}}$ has a Lipschitz gradient on Ω with a constant satisfying*

$$L_R \leq L_E + \frac{G^2}{4\tau}. \tag{19}$$

Proof. The chain rule gives

$$\nabla_\theta^2 \mathcal{R}_{\mathrm{EMBER}} = r''(e)\nabla_\theta e \nabla_\theta e^\top + r'(e)\nabla_\theta^2 e.$$

Using $|r'(e)| \le 1$, $r''(e) \le 1/(4\tau)$, and the assumed bounds yields $\|\nabla_\theta^2 \mathcal{R}_{\mathrm{EMBER}}\| \le G^2/(4\tau) + L_E$. For any $\theta, \phi \in \Omega$, the line segment joining them lies in Ω, and

$$\nabla\mathcal{R}(\phi) - \nabla\mathcal{R}(\theta) = \int_0^1 \nabla^2\mathcal{R}(\theta + t(\phi-\theta))(\phi-\theta)\,dt.$$

Taking norms proves the stated Lipschitz bound. □

If the ordinary-data objective is $L_{\mathcal{O}}$-smooth and these bounds hold uniformly over protected examples, the empirical combined objective satisfies

$$L_{\mathrm{total}} \le L_{\mathcal{O}} + \lambda\left(L_E + \frac{G^2}{4\tau}\right). \tag{20}$$

This bound exposes a design trade-off: smaller τ produces a sharper transition but increases its peak curvature, and λ scales the protected term's contribution.

A more specific pointwise bound retains the adaptive weight:

$$\|\nabla^2\mathcal{R}(\theta)\| \le wL_E + \frac{w(1-w)}{\tau}G^2. \tag{21}$$

If $w \le \epsilon < 1/2$, then the right-hand side is at most $\epsilon L_E + \epsilon(1-\epsilon)G^2/\tau$. This follows because $w(1-w)$ is increasing on $[0, 1/2]$. Thus, both the gradient and the additional curvature attenuate for examples sufficiently above the target. To use this as a smoothness constant on a region, the energy threshold and derivative bounds must hold throughout that region. Scalar convexity in e does not imply convexity in model parameters: the term $r'(e)\nabla^2 e$ can have either sign.

A.4 Protected Pressure in the Combined Update

The negative scalar derivative describes the contribution of one protected example. Its effect within joint training can be made explicit. For a fixed minibatch, let

$$J(\theta) = F_{\mathcal{O}}(\theta) + \frac{\lambda}{B}\sum_{j=1}^{B} r(e_j(\theta)), \qquad g_o = \nabla F_{\mathcal{O}}, \quad g_j = \nabla e_j, \quad w_j = w^*(e_j).$$

The negative-gradient direction is

$$d = -\nabla J = -g_o + \frac{\lambda}{B}\sum_j w_j g_j. \tag{22}$$

Its directional effect on protected example i is

$$g_i^\top d = -g_i^\top g_o + \frac{\lambda}{B} w_i \|g_i\|^2 + \frac{\lambda}{B}\sum_{j\ne i} w_j g_i^\top g_j. \tag{23}$$

The example's own term is nonnegative, while ordinary-data fitting and other protected examples contribute according to their gradient alignment. EMBER therefore supplies adaptive upward pressure without requiring every protected energy to increase at every joint update. If ordinary fitting lowers an example's energy, Equation 18 shows that its weight rises again.

For a finite step $\theta^+ = \theta + \eta d$ whose segment remains in a region with $\|\nabla^2 e_i\| \le L_E$, Taylor's theorem gives

$$\left|e_i(\theta^+) - e_i(\theta) - \eta g_i^\top d\right| \le \frac{L_E\eta^2}{2}\|d\|^2. \tag{24}$$

In particular, for the isolated direction $d = ag_i$ with $a = \lambda w_i/B > 0$,

$$e_i(\theta^+) - e_i(\theta) \ge \eta a\left(1 - \frac{L_E\eta a}{2}\right)\|g_i\|^2.$$

This verifies an energy increase for a nonzero gradient when $0 < \eta a < 2/L_E$; if $L_E = 0$, any positive step satisfying the region condition has this property. Equations 22–24 characterize the gradient geometry. The experiments use AdamW, whose momentum, preconditioning, and weight decay also affect the actual parameter displacement.

A.5 From Penalty Values to Energy Deficits

The margin defines an optimization target, and the remaining protected loss quantifies how closely examples approach it. This interpretation uses the same per-example energies as the objective.

Lemma A.2 (Penalty control of below-target examples)**.** *For a finite protected set with energies $e_1, \dots, e_N$, let $\bar{R} = N^{-1} \sum_i r(e_i)$. For any $\delta \geq 0$, define $q_\delta = N^{-1}|\{i : e_i \leq m - \delta\}|$. Then*

$$\frac{1}{N} \sum_i (m - e_i)_+ \leq \bar{R}, \qquad q_\delta \leq \min\left\{1, \frac{\bar{R}}{\tau \log(1 + e^{\delta/\tau})}\right\}. \tag{25}$$

Proof. The first inequality follows by averaging Equation 12. If $e_i \leq m - \delta$, monotonicity gives $r(e_i) \geq r(m - \delta) = \tau \log(1 + e^{\delta/\tau})$. Summing this lower bound over the indicated subset yields the second inequality, together with $q_\delta \leq 1$. □

There is also an exact per-example conversion. For any $\rho > 0$,

$$r(e) \leq \rho \quad \Longleftrightarrow \quad e \geq m - \tau \log\left(e^{\rho/\tau} - 1\right). \tag{26}$$

Exponentiating $r(e)/\tau = \log(1 + e^{(m-e)/\tau})$ and rearranging proves the equivalence. In particular, $r(e) \leq \tau \log 2$ is equivalent to $e \geq m$. These identities quantify energy deficits on the evaluated protected set; they are not measurements reported in the experiments or guarantees that every example reaches the margin after a fixed training budget.

A.6 Sequence Probability and the Evaluated Prefix

The energy definition implies the identity

$$p_\theta(y_c \mid s) = \prod_{t=1}^{L} p_\theta(y_{c,t} \mid s, y_{c,<t}) = e^{-LE(y_c|s;\theta)}. \tag{27}$$

It immediately gives the following interpretation of an achieved margin.

Corollary A.3 (Probability interpretation of the energy margin)**.** *For a protected continuation of length L, if $E(y_c \mid s; \theta) \geq m$, then*

$$p_\theta(y_c \mid s) \leq e^{-Lm}. \tag{28}$$

Proof. Substitute $E(y_c \mid s; \theta) \geq m$ into Equation 27. □

The corollary translates the target into probability units and applies to any autoregressive model satisfying the margin. EMBER encourages that target through its protected-data objective. The average energy removes the direct length multiplier from the training target, while the full-sequence probability retains its exponential dependence on length.

For an integer $1 \leq k \leq L$, define the prefix energy for a k-token exact-match event:

$$E_k(y_c \mid s; \theta) = -\frac{1}{k} \sum_{t=1}^{k} \log p_\theta(y_{c,t} \mid s, y_{c,<t}). \tag{29}$$

Under ancestral sampling from p_θ,

$$\Pr(\hat{y}_{1:k} = y_{c,1:k} \mid s) = e^{-kE_k(y_c|s;\theta)}. \tag{30}$$

Thus, a margin $E_k \geq m_k$ yields a prefix probability at most e^{-km_k}. For EM@10 this uses $k = 10$ and the first ten reference tokens. A full-continuation margin does not by itself establish this prefix margin, since later tokens also contribute to the full average. Greedy and beam decoding depend on token or sequence rankings, and temperature or top-p sampling changes the sampling distribution. Our experiments use greedy decoding and evaluate EM@10 directly from generated continuations; the probability identity explains the energy target rather than supplying a numerical bound on the reported EM scores.

Full-continuation and prefix energies. For $k < L$, define E_{tail} as the average NLL of the remaining $L - k$ reference tokens, conditioned on their reference histories. Then

$$LE(y_c \mid s; \theta) = kE_k(y_c \mid s; \theta) + (L - k)E_{\text{tail}}(y_c \mid s; \theta). \tag{31}$$

Nonnegativity implies $E_k \leq LE/k$, but gives no positive lower bound on E_k from a lower bound on E alone. For example, choose a small $a > 0$ with $ka < Lm$ and let the reference-token NLL be a for the first k tokens and $(Lm - ka)/(L - k)$ for the remaining tokens. All corresponding probabilities are in $(0, 1)$, the full energy equals m, and the prefix energy equals a. This explains why the full-sequence target and the generated-prefix metric play complementary roles.

Token ranking under greedy decoding. With a fixed rule for resolving ties, define $a_t = \arg\max_v p_\theta(v \mid s, y_{c,<t})$ along the reference path. Greedy decoding reproduces the first k reference tokens exactly if and only if $a_t = y_{c,t}$ for every $t \leq k$. The claim follows by induction: after matching the first $t - 1$ tokens, greedy decoding uses precisely the reference history at step t.

This ranking condition differs from a bound on the product of reference-token probabilities. As an illustrative probability construction, assign probabilities 0.34, 0.33, and 0.33 to a three-token vocabulary at every reference history, with the reference token receiving 0.34. The reference prefix is the unique greedy output, yet $E_k = -\log 0.34 \approx 1.079$ and its probability under ancestral sampling is 0.34^k. Thus, even an achieved prefix energy above the default margin can coexist with an exact greedy continuation. This mathematical example clarifies the distinct events; it is not an additional empirical result. The experiments assess the effect on token rankings through the generated continuations themselves.

Sampling over a fixed prompt set. If each protected prompt is sampled ancestrally, the expected exact-match fraction is

$$\mathbb{E}[\widehat{\text{EM}}@k] = \frac{1}{N} \sum_{i=1}^{N} e^{-kE_{k,i}}.$$

If at least a fraction $1 - q$ of these prompts have $E_{k,i} \geq m_k \geq 0$, this expectation is at most $q + (1 - q)e^{-km_k}$. The proof bounds each probability in that subset by e^{-km_k} and every remaining probability by one. A bound on only the mean prefix energy cannot be substituted here: convexity of e^{-kx} gives the Jensen inequality in the opposite direction. This statement concerns ancestral sampling from the original model distribution, while the reported EM@10 uses greedy decoding.

A.7 Numerical Evaluation and Loss Reduction

The scalar penalty can be evaluated without exponentiating a large positive argument:

$$r(e) = (m - e)_+ + \tau \log\left(1 + \exp\left(-\frac{|m - e|}{\tau}\right)\right). \tag{32}$$

This is the same softplus identity used above; its exponential argument is nonpositive. A stable softplus routine evaluates this expression and differentiates the smooth function through the transition. Sequence energy is computed from token log-probabilities, avoiding the numerical underflow that can arise from first multiplying a long sequence of probabilities.

For a variable-length protected minibatch, the objective specifies the reduction order

$$e_i = -\frac{1}{L_i} \sum_{t=1}^{L_i} \log p_\theta(y_{i,t} \mid s_i, y_{i,<t}), \qquad \widehat{R} = \frac{1}{B} \sum_{i=1}^{B} r(e_i).$$

Here L_i counts the scored continuation tokens, excluding padding and prompt tokens. Averaging the token energies within each continuation and then applying the penalty preserves an example-level target. A penalty on the batch mean is the different objective in Equation 17; averaging token-wise penalties is also different because softplus is nonlinear. At fixed θ, sampling uniformly within each of the protected and ordinary sets makes the minibatch loss and gradient unbiased estimators of the finite empirical objective and gradient. This follows by linearity of expectation over finite sums; it does not require differentiating an unrestricted population expectation.

B EMBER ALGORITHMIC DESCRIPTION

This appendix gives a standalone description of the proposed method. The model computes token log-probabilities with teacher forcing; E_o and E_c are the average token negative log-likelihoods for ordinary and protected examples, respectively. The protected term uses a stable softplus implementation, so no sequence is removed from the computation graph and the trainable parameters receive the gradient of the combined objective.

Step	EMBER algorithm
1	**Input:** pretrained autoregressive model f_θ, ordinary pairs $\mathcal{O}$, protected pairs $\mathcal{C}$, margin m, temperature $\tau > 0$, weight λ, optimizer, and training epochs T.
2	Initialize θ from the pretrained checkpoint. For each epoch and minibatch, draw ordinary batch B_o and protected batch B_c with the same seed-controlled ordering used by the baselines.
3	Run teacher-forced forward passes and compute $E_o = \mathrm{mean}_{(s,y)\in B_o} \mathrm{Energy}(s, y; \theta)$ and each protected energy $E_{c,i} = \mathrm{Energy}(s_i, y_{c,i}; \theta)$.
4	For every protected example, compute the margin deficit barrier $r_i = \tau\, \mathrm{softplus}((m - E_{c,i})/\tau)$.
5	Form the minibatch objective $\mathcal{L} = E_o + \lambda\, \mathrm{mean}_i(r_i)$, backpropagate through f_θ, and update θ with the optimizer.
6	Repeat until T epochs are complete; save the resulting checkpoint and use greedy decoding for evaluation.

Table 4: Standalone pseudocode for EMBER training. The penalty is strong for low-energy protected continuations and attenuates smoothly above the target margin.

C HYPERPARAMETER SENSITIVITY

C.1 SENSITIVITY PROTOCOL

We examine the sensitivity of EMBER to the target margin m, transition temperature τ, and trade-off weight λ using Qwen-3-8B with LoRA. The study follows the same training, prompt construction, and greedy-decoding protocols as the main BookMIA and controlled Wikitext-2 experiments. We use a one-at-a-time design centered at the default configuration $(m = 1, \tau = 0.05, \lambda = 1)$: one hyperparameter is varied while the other two remain fixed. This design measures local changes around the shared configuration; it does not characterize interactions among the three parameters or establish a global optimum.

Table 2 reports EM as the leakage measure and MAUVE as the ordinary-generation utility measure. These two metrics expose the primary protection–utility trade-off compactly: lower EM indicates less exact continuation leakage, whereas higher MAUVE indicates that the generated-text distribution remains closer to the ordinary reference distribution. The reported trends should therefore be interpreted jointly rather than by optimizing either column in isolation.

C.2 EFFECT OF THE TARGET MARGIN

The first block of Table 2 varies m from 0.5 to 2.0. The results show a monotonic protection–utility trade-off on both datasets. On BookMIA, increasing m reduces EM from 4.7 to 0.6, while MAUVE decreases gradually from 0.91 to 0.86. The controlled Wikitext-2 split exhibits the same pattern: EM falls from 1.4 to 0.1, accompanied by a MAUVE change from 0.92 to 0.87. This trend is consistent with the role of m in the barrier. A larger margin requires a protected continuation to reach a higher average conditional energy before its repulsive gradient becomes negligible, so stronger protection is expected. The accompanying MAUVE reduction shows the generation-quality cost of stronger suppression on the ordinary evaluation set.

BookMIA contains corpus variation, whereas the Wikitext-2 split reduces differences between protected and ordinary text. The same ordering across the two settings supports the observed direction of the margin trade-off. The default $m = 1$ lowers EM while retaining MAUVE close to the weaker $m = 0.5$ setting. These observations describe the tested settings and do not identify a dataset-independent optimum.

C.3 Effect of the Transition Temperature

The second block varies τ while fixing $m = 1$ and $\lambda = 1$. Compared with the changes caused by m, the results are relatively insensitive to τ. On BookMIA, EM ranges from 1.6 to 2.4 and MAUVE from 0.89 to 0.90 across $\tau \in \{0.02, 0.05, 0.10\}$. On Wikitext-2, EM ranges from 0.5 to 0.9 and MAUVE from 0.90 to 0.91. Thus, changing the transition sharpness produces only small variations around the same operating region.

This stability agrees with the analytical role of τ. The parameter changes how rapidly the sigmoid-shaped repulsive gradient decays around $E = m$, but it does not change the target energy itself. A smaller τ approximates a harder hinge, keeping the scalar gradient near unit magnitude below the margin and making it decay more sharply above it. A larger τ broadens this transition on both sides of $E = m$. The default $\tau = 0.05$ avoids an excessively sharp transition while remaining close to the leakage and utility values obtained at both neighboring settings.

C.4 Effect of the Trade-off Weight

The final block varies λ, which controls the contribution of protected examples relative to ordinary likelihood training. Reducing λ from 1.0 to 0.5 prioritizes utility: MAUVE increases from 0.90 to 0.91 on BookMIA and from 0.90 to 0.92 on Wikitext-2, but EM rises from 1.9 to 3.3 and from 0.6 to 1.1, respectively. Increasing λ to 2.0 has the opposite effect. EM decreases to 0.9 on BookMIA and 0.3 on Wikitext-2, while MAUVE declines to 0.88 and 0.89.

Unlike m, which sets the center of the gradient transition, λ scales the protected-data gradient throughout training. The two parameters therefore control different aspects of the method: m specifies the desired protection target, while λ determines how strongly optimization pursues that target relative to ordinary-data fitting. Their empirical effects are nevertheless complementary, with larger values of either parameter yielding stronger leakage suppression and a gradual utility cost.

C.5 Overall Robustness and Default Selection

Across all tested configurations, the direction of each trend is consistent between the realistic and controlled datasets. The method does not exhibit a sudden collapse within the tested neighborhood: EM remains far below the unprotected levels reported in the main experiments, while MAUVE remains well above the severe degradation observed for IR in the main comparison. The default configuration is not chosen to optimize a single leakage-only or utility-only metric. Instead, $(m = 1, \tau = 0.05, \lambda = 1)$ provides a central operating point that combines low EM with near-reference MAUVE on both datasets. Applications with stricter protection requirements can increase m or λ, whereas applications that prioritize generation fidelity can reduce either value. The relatively small variation across τ indicates that the method is less sensitive to the precise smoothness of the margin transition than to the protection target and the protected-data weight.

D Experimental Protocol and Reproducibility Details

This appendix specifies the common initialization, the roles of the two training corpora, and the relationship between protected training and leakage evaluation. It also distinguishes baseline loss definitions from the shared optimization and evaluation settings.

D.1 Common Initialization and Data Flow

For each model, SFT, IR, Goldfish, CR, and EMBER start independently from the same base checkpoint. SFT is one of these adaptations: it minimizes the likelihood loss on the union of ordinary and protected training text. EMBER uses ordinary-data likelihood training together with protected-data repulsion. Thus, the methods differ in how their adaptation objectives use the designated protected text, while sharing the initial model.

The reference passages underlying the 1,000 leakage-evaluation prompts also participate in protected training. Each method is evaluated under the same protected prefixes within a run. This tests suppression of designated training passages after adaptation. It does not test generalization to pas-

Component	Role in the comparison
Base checkpoint	Common initialization for all methods within a model comparison.
Ordinary corpus $\mathcal{O}$	Text used for ordinary-data likelihood training.
Protected corpus $\mathcal{C}$	Designated text learned positively by SFT and treated according to each mitigation objective.
SFT adaptation	Likelihood training on $\mathcal{O} \cup \mathcal{C}$ from the common base checkpoint.
EMBER adaptation	Ordinary likelihood plus the per-example penalty in Equation 7, from the same base checkpoint.
Leakage evaluation	1,000 protected prefixes whose reference passages also occur in protected training; greedy EM@10 and ROUGE-L.
Ordinary evaluation	PPL and MAUVE on ordinary data, assessed separately from protected-text leakage.

Table 5: Data roles and common initialization. The protected/ordinary partition specifies an objective role, not a train/test split.

sages excluded from the protection objective. The relation is stated at the passage level; it does not require every training window to coincide with an evaluation prompt–continuation pair.

D.2 Corpus Construction and Evaluation Identity

For BookMIA, designated pre-2023 member books supply $\mathcal{C}$, and post-2023 books supply $\mathcal{O}$. We use the dataset labels to assign these roles in continuation training and evaluation. They do not establish pretraining membership for every model checkpoint considered here. The two book groups may differ in content and publication time.

For Wikitext-2, examples are randomly assigned to disjoint protected and ordinary subsets in a 2:8 ratio. Both subsets originate from the same corpus, reducing corpus-level differences. Disjointness of $\mathcal{C}$ and $\mathcal{O}$ concerns their roles in the objective; the leakage-evaluation passages remain represented in protected training.

Each leakage example consists of a 40-token prefix s and its reference continuation y_c. All methods in a run use the same ordered evaluation-prompt list. Configuration selection uses a validation split rather than the reported leakage-evaluation prompts. This distinction concerns model selection; it does not imply that the underlying leakage-evaluation passages were absent from protected training.

D.3 Baseline Objectives and Normalization

Write $E_i = -L_i^{-1} \log p_\theta(y_i \mid s_i)$ for the mean NLL of the scored continuation tokens. The normalization and order of loss reduction are part of an objective: an average of transformed per-example losses generally differs from a transformation of the average loss. The following descriptions distinguish the losses compared in the paper and the original formulations underlying the adaptations.

SFT. SFT performs positive likelihood training on both $\mathcal{O}$ and $\mathcal{C}$. Its exposure to protected text makes it the unprotected adaptation reference. It is distinct from an ordinary-only adaptation and from the base model before adaptation. When writing a mixture objective as $a_o F_{\mathcal{O}} + a_c F_{\mathcal{C}}$, the coefficients depend on sampling and loss reduction; the set union alone does not imply equal weights.

IR. The protected term denoted IR in this paper is the full conditional sequence probability, $p_\theta(y_i \mid s_i) = \exp(-L_i E_i)$, as in Equation 5. Its per-example derivative with respect to energy is $-L_i \exp(-L_i E_i)$. The geometric mean probability $\exp(-E_i)$ and an average of token probabilities would be different objectives. Computing sequence log-probability by summing token log-probabilities avoids an explicit product, although the final exponentiation and gradient can still become very small. This length dependence matters when interpreting a sequence-probability baseline; its observed utility loss pertains to the reported operating point.

Goldfish. Goldfish excludes selected token losses while retaining all preceding tokens in the conditioning history (Hans et al., 2024). With a binary mask $G_{i,t}$ and a normalization Z_i, a masked

token-NLL term has the form

$$\ell_{G,i} = -\frac{1}{Z_i} \sum_{t=1}^{L_i} G_{i,t} \log p_\theta(y_{i,t} \mid s_i, y_{i,<t}). \tag{33}$$

The mask construction and normalization specify the variant. Normalizing by all scored tokens or by retained tokens changes the loss scale. The original paper distinguishes static, random, and context-hashed masks; its main formulation uses a hash-based mask to preserve consistent omissions when a passage reappears. These distinctions matter because resampling a mask across repeated exposures can eventually supervise different tokens from the same passage. Masking the loss also differs from deleting tokens from the input.

Copyright Regression. The original Copyright Regression formulation uses squared prediction errors ℓ_1 and ℓ_2 on protected and ordinary examples, respectively (Chu et al., 2024). Its copyright objective is

$$\mathcal{L}_{\mathrm{CR,orig}} = \tfrac{1}{2}\ell_1 + \gamma_c \ell_1^{-1} + \tfrac{1}{2}\ell_2, \qquad \gamma_c > 0, \tag{34}$$

with an additional parameter regularizer in that formulation. The protected component combines positive fitting and an inverse-loss penalty. This motivates an inverse-loss adaptation for language modeling, rather than a plain reversal of the NLL gradient. The squared-error formulation and a language-model NLL adaptation have different loss scales and do not share optimization guarantees automatically. Aggregation also matters: $1/(B^{-1}\sum_i E_i)$ differs from $B^{-1}\sum_i 1/E_i$. The cited objective therefore specifies the origin of CR, while the adaptation determines its behavior in a language-model comparison.

EMBER. EMBER computes one energy per protected continuation, applies the margin penalty to each energy, and averages these penalties. For ordinary and protected minibatches B_o and B_c, its objective is

$$\widehat{\mathcal{L}}_{\mathrm{EMBER}} = \frac{1}{|B_o|} \sum_{i \in B_o} E_i + \frac{\lambda}{|B_c|} \sum_{j \in B_c} \tau \,\mathrm{softplus}\left(\frac{m - E_j}{\tau}\right). \tag{35}$$

The main and scaling experiments use $m = 1$, $\tau = 0.05$, and $\lambda = 1$. Prompt and padding tokens are excluded from the continuation energy, as specified in Appendix A.7. The ordinary term remains a likelihood objective; protected text contributes the margin penalty rather than an additional positive likelihood term.

D.4 Shared Optimization and Model Adaptation

Experiments use an NVIDIA A100 GPU with 80 GB of memory and mixed precision. GPT-2 updates all model parameters. Llama-3 and Qwen-3 base models use LoRA with rank $r = 16$ and scaling $\alpha = 32$. Each within-model comparison shares the adaptation regime, hardware type, learning rate, epoch count, and reported batch-size setting.

AdamW uses learning rate 5×10^{-5} for 10 epochs with batch size 32. Model initialization, data ordering, and other stochastic operations are controlled by the run seed. The training and evaluation pipeline is repeated for five random seeds. Main-table entries are means across these runs, with the subscripted $\pm$ value denoting standard deviation. Shared optimization settings describe the comparison budget; they do not by themselves establish an optimal or utility-matched operating point for every objective.

D.5 Evaluation and Aggregation

For each run, the exact-match rate is

$$\mathrm{EM@10} = \frac{100}{N} \sum_{i=1}^{N} \mathbb{1}[\hat{y}_{i,1:10} = y_{c,i,1:10}], \qquad N = 1000. \tag{36}$$

ROUGE-L measures subsequence overlap beyond this exact prefix event. Greedy decoding gives a fixed continuation for a fixed model and prompt; the seed variability in the main table arises across

Component	Setting
Hardware	NVIDIA A100, 80 GB
Optimizer	AdamW
Learning rate	5×10^{-5}
Training epochs	10
Batch size	32
Numerical precision	Mixed precision
Independent runs	Five random seeds
GPT-2 adaptation	Full-parameter fine-tuning
Llama-3/Qwen-3 adaptation	LoRA
LoRA configuration	Rank $r = 16$, scaling $\alpha = 32$
EMBER configuration	$m = 1$, $\tau = 0.05$, $\lambda = 1$
Leakage evaluation prompts	1,000 protected prompts
Prompt length	40 tokens
Initialization	Same base checkpoint within each model comparison
Decoding	Greedy (highest-probability token at each step)

Table 6: Shared optimization, adaptation, and evaluation settings. Method-specific loss definitions are described separately.

Step	Evaluation and aggregation
1	**Input:** one adapted checkpoint per seed, the shared protected prompt list with reference continuations, and ordinary evaluation examples.
2	Condition on each 40-token protected prefix and generate a continuation with greedy decoding.
3	Record $\mathrm{EM@10} = \mathbb{1}[\hat{y}_{1:10} = y_{c,1:10}]$ and ROUGE-L for each generated/reference pair.
4	Evaluate ordinary-data PPL from token likelihoods and MAUVE between generated and ordinary reference continuations.
5	Repeat for five seeds; report the mean and standard deviation of each metric in the main table.

Table 7: Evaluation of protected-text leakage and ordinary-generation quality.

independently trained runs. Means printed as 0.0% are rounded reported values and should not be read as exact zero leakage when the corresponding variability is nonzero.

PPL measures ordinary-text likelihood under the model's tokenizer. MAUVE compares the distributions of generated and ordinary reference continuations. These metrics assess ordinary-corpus generation, not all downstream capabilities. The same high-level evaluation is used in the main comparison and scaling study; the sensitivity study changes the specified EMBER parameter.

D.6 Interpretation of the Comparisons

The study compares alternative adaptations of the same base model. SFT positively learns the designated protected text, while EMBER applies repulsion to that text together with ordinary likelihood training. The resulting difference measures the effect of choosing these adaptation objectives. It is not a measurement of forgetting from a common SFT-trained checkpoint, and protected training membership of evaluation passages is part of the stated task.

The main results jointly report leakage and ordinary-generation metrics. They describe the evaluated configurations; a single configuration per method does not establish its entire leakage–utility frontier. Similarly, one-at-a-time sensitivity results describe a neighborhood around the shared EMBER setting, rather than causal isolation of every component. Sequence-energy analysis explains the training signal, while greedy EM@10 tests its effect on reference-token rankings. Other prompting strategies, decoding rules, extraction budgets, and unseen protected passages require separate evaluation.