# Temper-Then-Tilt: Principled Unlearning for Generative Models through Tempering and Classifier Guidance

**Jacob Block**[1*] Mehryar Mohri[2] Aryan Mokhtari[1,2] Sanjay Shakkottai[1]

[1]UT Austin [2]Google Research

*Work Partially done while a Student Researcher at Google Research

Authors listed alphabetically

# Unlearning for Generative Models

**Setup**: Generative model over samples $Z$ with ground truth density $p(\boldsymbol{z})$

**Data**: $\mathcal{D} = \mathcal{D}_r \sqcup \mathcal{D}_f$ split across retain set $\mathcal{D}_r$ and forget set $\mathcal{D}_f$

**Given**: Pretrained model $p_{\boldsymbol{\theta}^*} = p$, sample access to $\mathcal{D}_r$ and $\mathcal{D}_f$

**Goal**: Unlearn $\mathcal{D}_f$ , i.e., learn the distribution which would have resulted from training on $\mathcal{D}_r$ only

**Target**: $p_r(\boldsymbol{z})$ , the density over $\mathcal{D}_r$

# Unlearning as Posterior Sampling

- Define Bernoulli random variable $S$ which indicates retain set membership:
  - $\boldsymbol{z} \in \mathcal{D}_r$ denotes the realization $S = 1$, else $S = 0$
- Define retain component density $p_r$ as the conditional density over $\boldsymbol{z} \mid S = 1$, forget component density $p_f$ as conditional density over $\boldsymbol{z} \mid S = 0$
- **Task:** Given samples $\mathcal{S} = \{(\boldsymbol{z}_i, s_i) \mid \boldsymbol{z}_i \in \mathcal{D}, s_i = \mathbb{1}\{\boldsymbol{z}_i \in \mathcal{D}_r\}\}$ with known marginal $\boldsymbol{z}_i \sim p(\boldsymbol{z})$, learn the distribution of $\boldsymbol{Z}$ conditioned on $S = 1$

# Temper-Then-Tilt Estimator

- **Classifier Guidance:** $p_r(\boldsymbol{z}) \propto p(\boldsymbol{z}) \cdot f^*(\boldsymbol{z})$ where $f^*$ is the Bayes' optimal classifier between $\mathcal{D}_r$ and $\mathcal{D}_f$, i.e., $f^*(\boldsymbol{z}) = \mathbb{P}(S = 1 \mid \boldsymbol{z})$
- Train classifier $f_{\hat{\boldsymbol{\phi}}}(\boldsymbol{z}) \approx f^*(\boldsymbol{z})$ parameterized by $\hat{\boldsymbol{\phi}}$ on the finite dataset $\mathcal{S} = \{(\boldsymbol{z}_i, s_i) \mid \boldsymbol{z}_i \in \mathcal{D}, s_i = \mathbb{1}\{\boldsymbol{z}_i \in \mathcal{D}_r\}\}$
- Define the **T3-Unlearning estimator** $\hat{p}_r^{(T)}(\boldsymbol{z}) \propto p_{\boldsymbol{\theta}^*}(\boldsymbol{z})^{1/T} \cdot f_{\hat{\boldsymbol{\phi}}}(\boldsymbol{z})$ with parameter $T \geq 1$

# Theoretical Analysis

- Analyze error of the unlearned distribution $\hat{p}_r^{(T)}(\boldsymbol{z}) \propto p(\boldsymbol{z})^{1/T}\hat{f}(\boldsymbol{z})$ in terms of classifier excess risk $L(\hat{f}) - L(f^*) \leq \delta$

- Define two complimentary error metrics

  - **Retain Error:** $\mathcal{E}_r(\hat{p}_r) \coloneqq \mathrm{KL}(p_r \,||\, \hat{p}_r) = \mathbb{E}_{\boldsymbol{Z}\sim p_r}\left[\ln \dfrac{p_r(\boldsymbol{Z})}{\hat{p}_r(\boldsymbol{Z})}\right]$

  - **Forget Error:** $\mathcal{E}_f(\hat{p}_r) \coloneqq \|p_r - \hat{p}_r\|_{1,p_f} = \mathbb{E}_{\boldsymbol{Z}\sim p_f}\left[|p_r(\boldsymbol{Z}) - \hat{p}_r(\boldsymbol{Z})|\right]$

# Theoretical Results

- Retain Error of untempered estimator is bounded $\mathcal{E}_r(\hat{p}_r^{(1)}) = \mathcal{O}(\delta)$
- Forget Error is linear in the forget component peak density $\mathcal{E}_f(\hat{p}_r^{(1)}) = \mathcal{O}(\|p_f\|_\infty \sqrt{\delta})$
- This dependence is tight; $\exists\, \hat{f}$ with excess risk $\delta$ s.t. $\mathcal{E}_f(\hat{p}_r^{(1)}) = \Omega\left(\|p_f\|_\infty \cdot \delta\right)$
- Tempering improves robustness to forget component sharpness at the cost of estimator bias
  - Forget Error: $\mathcal{E}_f(\hat{p}_r^{(T)}) \leq \mathcal{O}(\|p_f\|_\infty^{1/T} \cdot \delta^{1/2k}) + \mathrm{Bias}(T)$ for $k \geq T$
  - Retain Error: $\mathcal{E}_r(\hat{p}_r^{(T)}) \leq \mathcal{O}(\delta) + \mathrm{bias}(T)$ where $\mathrm{Bias}(1) = \mathrm{bias}(1) = 0$

# LLM Implementation

**Autoregressive Form:** $p_r(y \mid \boldsymbol{x}) \propto p(y \mid \boldsymbol{x}) \cdot \mathbb{P}(S = 1 \mid \boldsymbol{x}, y)$, where $\boldsymbol{x}$ is prefix of tokens and $y$ is a candidate next token in the sequence

**Classifier:** Train $f_{\boldsymbol{\phi}}(\boldsymbol{x}, y)$ to predict probability that the sequence $(\boldsymbol{x}, y)$ comes from $\mathcal{D}_r$ for each prefix $(\boldsymbol{x}, y)$ in each string in $\mathcal{D}$

**Unlearning Update:** $\hat{p}_r^{(T)}(y \mid \boldsymbol{x}) \propto p_{\boldsymbol{\theta}^*}(y \mid \boldsymbol{x})^{1/T} \cdot f_{\hat{\boldsymbol{\phi}}}(\boldsymbol{x}, y)$

**Parameterization:** Define $f_{\boldsymbol{\phi}}$ as low-rank linear head on top of mean-pooled base model hidden states

# TOFU Benchmark Performance

- We primarily evaluate on the TOFU unlearning benchmark
  - Forget Quality: p-value measuring indistinguishability to a retrained reference model
  - MU-ROUGE: Harmonic mean of generative utility metrics over 3 datasets

| | | | | | Utility Metrics | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Retain | | | WF | | | RA | | |
| Split | Method | FQ | MU-ROUGE | MU | Prob. | ROUGE | TR+ | Prob. | ROUGE | TR+ | Prob. | ROUGE | TR+ |
| - | Original Model | 0.000 | 0.948 | 0.637 | 0.991 | 0.995 | 0.531 | 0.479 | 0.905 | 0.625 | 0.409 | 0.949 | 0.514 |
| *5%* | GradAscent | 0.000 | 0.875 | 0.576 | 0.925 | 0.852 | 0.525 | 0.420 | 0.888 | 0.543 | 0.352 | 0.885 | 0.463 |
| | GradDiff | 0.003 | 0.281 | 0.363 | 0.536 | 0.529 | 0.452 | 0.359 | 0.499 | 0.477 | 0.359 | 0.153 | 0.480 |
| | WGA | 0.252 | 0.424 | 0.418 | 0.405 | 0.436 | 0.408 | 0.374 | 0.617 | 0.502 | 0.371 | 0.327 | 0.478 |
| | SatImp | 0.445 | 0.419 | 0.412 | 0.407 | 0.436 | 0.413 | 0.368 | 0.612 | 0.489 | 0.351 | 0.313 | 0.465 |
| | UNDIAL | 0.016 | 0.653 | 0.567 | 0.601 | 0.470 | 0.460 | 0.479 | 0.832 | 0.645 | 0.470 | 0.795 | 0.596 |
| | RMU | 0.532 | 0.000 | 0.000 | 0.000 | 0.038 | 0.275 | 0.185 | 0.005 | 0.449 | 0.202 | 0.000 | 0.516 |
| | ULD | 0.169 | 0.944 | 0.644 | 0.988 | 0.974 | 0.534 | 0.498 | 0.910 | 0.641 | 0.413 | 0.950 | 0.518 |
| | IdkDPO | 0.535 | 0.669 | 0.545 | 0.678 | 0.534 | 0.481 | 0.429 | 0.833 | 0.585 | 0.402 | 0.708 | 0.518 |
| | NPO | 0.793 | 0.713 | 0.631 | 0.779 | 0.590 | 0.507 | 0.547 | 0.846 | 0.719 | 0.504 | 0.751 | 0.638 |
| | SimNPO | 0.745 | 0.786 | 0.653 | 0.830 | 0.668 | 0.508 | 0.549 | 0.868 | 0.720 | 0.505 | 0.856 | 0.634 |
| | T3-Unlearning (ours) | 0.914 | 0.900 | 0.612 | 0.415 | 0.983 | 0.461 | 0.545 | 0.874 | 0.688 | 0.511 | 0.853 | 0.648 |
| *10%* | GradAscent | 0.000 | 0.950 | 0.638 | 0.991 | 0.995 | 0.531 | 0.481 | 0.910 | 0.630 | 0.409 | 0.949 | 0.512 |
| | GradDiff | 0.065 | 0.000 | 0.000 | 0.018 | 0.035 | 0.364 | 0.239 | 0.001 | 0.335 | 0.216 | 0.000 | 0.359 |
| | WGA | 0.020 | 0.834 | 0.629 | 0.873 | 0.748 | 0.502 | 0.511 | 0.898 | 0.691 | 0.430 | 0.875 | 0.545 |
| | SatImp | 0.219 | 0.301 | 0.375 | 0.524 | 0.531 | 0.444 | 0.368 | 0.517 | 0.494 | 0.353 | 0.165 | 0.462 |
| | UNDIAL | 0.000 | 0.927 | 0.697 | 0.971 | 0.961 | 0.517 | 0.561 | 0.908 | 0.729 | 0.504 | 0.914 | 0.633 |
| | RMU | 0.001 | 0.000 | 0.000 | 0.000 | 0.072 | 0.299 | 0.229 | 0.023 | 0.588 | 0.298 | 0.000 | 0.477 |
| | ULD | 0.012 | 0.944 | 0.642 | 0.988 | 0.978 | 0.533 | 0.494 | 0.904 | 0.638 | 0.412 | 0.952 | 0.517 |
| | IdkDPO | 0.020 | 0.413 | 0.465 | 0.656 | 0.509 | 0.486 | 0.431 | 0.655 | 0.586 | 0.402 | 0.271 | 0.522 |
| | NPO | 0.185 | 0.521 | 0.493 | 0.602 | 0.542 | 0.468 | 0.422 | 0.664 | 0.569 | 0.393 | 0.416 | 0.506 |
| | SimNPO | 0.536 | 0.840 | 0.662 | 0.870 | 0.745 | 0.529 | 0.506 | 0.904 | 0.685 | 0.502 | 0.890 | 0.649 |
| | T3-Unlearning (ours) | 0.671 | 0.899 | 0.612 | 0.423 | 0.992 | 0.462 | 0.543 | 0.863 | 0.686 | 0.506 | 0.856 | 0.641 |

# Computational Efficiency

| Method | 5% Split | | 10% Split | |
|---|---|---|---|---|
| | Epochs | Runtime (s) | Epochs | Runtime (s) |
| GradAscent | 10 | 236 | 10 | 467 |
| GradDiff | 20 | 801 | 10 | 823 |
| WGA | 10 | 529 | 5 | 524 |
| SatImp | 10 | 443 | 10 | 1040 |
| UnDIAL | 5 | 241 | 10 | 966 |
| RMU | 10 | 566 | 10 | 1100 |
| ULD | 20 | 63.7 | 20 | 121 |
| IdkDPO | 10 | 837 | 10 | 1680 |
| NPO | 20 | 1270 | 10 | 1140 |
| SimNPO | 10 | 456 | 10 | 904 |
| T3-Unlearning (naïve) | 100 | 216 | 100 | 431 |
| T3-Unlearning (preprocessed) | 100 | **5.12** | 100 | **7.39** |

| Method | # Parameters |
|---|---|
| Full Fine-Tuning | $8.03 \times 10^9$ |
| ULD (LoRA rank 32, first 16 layers) | $4.19 \times 10^7$ |
| T3-Unlearning ($h = 20$) | $2.65 \times 10^6$ |

Runtimes (left) and number of trainable parameters (right) for each unlearning method

# Reference

J.L. Block, M. Mohri, A. Mokhtari, S. Shakkottai. “Temper-Then-Tilt: Principled Unlearning for Generative Models through Tempering and Classifier Guidance,” ICML 2026.

Thank You!