The Pitfalls of Maximum Likelihood Training of TPMs for Controllable Language Generation

Hanzhang Liu[1, *] **William Zhao**[1, *] **Zilei Shao**[1] **Benjie Wang**[2] **Guy Van den Broeck**[1]

[1]Department of Computer Science, University of California, Los Angeles, USA
[2]Kahlert School of Computing, University of Utah, USA
[*]Equal contribution

Abstract

Tractable probabilistic models (TPMs) are increasingly being applied in language modeling tasks, such as controlling the generation of large language models (LMs) to satisfy logical or semantic constraints. However, a thus-far overlooked aspect is the training procedure of the TPM, which optimizes for data likelihood rather than the downstream task. This raises a central question: does improving data likelihood necessarily improve downstream generation quality? In this work, we find that likelihood-trained TPMs can result in failed generations due to overly large corrections to the LM's logits. To address this, we train TPMs with LM-aligned objectives, including an ELBO-based objective and an MSE-based distillation objective. By learning directly from the LM rather than optimizing only for data likelihood, our approach produces TPMs that better align with the LM token-probability space. On a detoxification task, our results show these models avoid degeneration, maintain fluency under strong guidance, and enable stronger detoxification.

1 INTRODUCTION

Large language models (LMs) are highly capable sequence models, but enforcing semantic constraints during generation remains challenging. A recent line of work uses tractable probabilistic models (TPMs) to impose lexical, logical, and semantic constraints at decoding time [Zhang et al., 2023, 2024, Weng et al., 2025]. In this work, we study TRACE [Weng et al., 2025], a TPM-based framework for semantic control such as detoxification.

Existing TPM-based methods typically train the TPM by maximizing data log-likelihood. However, we find that these MLE TPMs can produce high-variance backward signals.

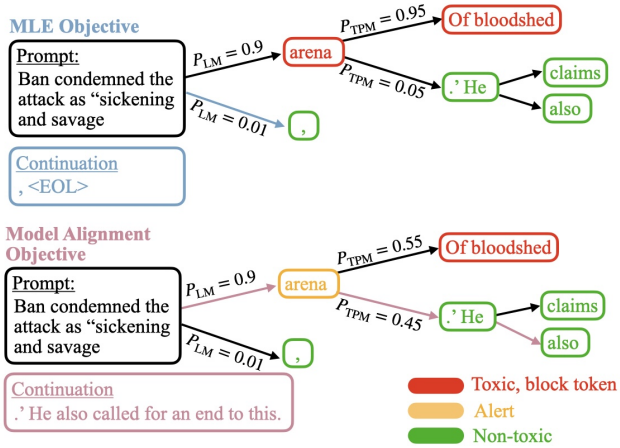

Figure 1: Illustration of TPM-induced correction under TRACE. P_{LM} denotes the LM next-token probability, and P_{TPM} denotes the TPM-predicted probability of future continuation paths. Colors indicate the TPM toxicity signal.

These extreme signals induce overconfident corrections, causing the TPM to dominate the LM's next-token distribution and resulting in fluency degradation and degeneration.

Figure 1 illustrates how overconfident TPM guidance can disrupt generation in TRACE. TRACE selects tokens by combining the LM's next-token probability with the TPM's toxicity-based correction. Although the LM assigns high probability to the fluent token "arena," the MLE TPM predicts that "arena" is likely to lead to the toxic continuation "of bloodshed," while underestimating possible non-toxic continuations. This makes "arena" appear much more toxic relative to the comma token, causing TRACE to suppress a natural high-probability LM token and instead select the lower-probability comma.

The pitfall of likelihood training is that it optimizes the TPM as a standalone sequence model, whereas TRACE uses the TPM as a decoding-time correction model. Therefore, optimizing data likelihood may not produce the guidance

signals that are most compatible with the LM. Motivated by this, we explore model-based training objectives that learn directly from the frozen LM, including a reverse-KL objective and an MSE-based distillation objective.

Model-based TPMs produce more calibrated backward correction. As shown in Figure 1, the model-based TPM assigns "arena" a softer toxicity signal relative to the comma token, marking it as potentially risky rather than as a token that must be blocked. When TRACE combines this correction with the LM next-token probabilities, "arena" remains the preferred token. The generation can then continue fluently while avoiding the actually toxic continuation "of bloodshed."

Downstream results show that model-based TPMs maintain stable fluency under strong TRACE guidance. This allows TRACE to use stronger guidance for better detoxification without causing perplexity collapse. Model-based TPMs also substantially reduce degenerate behaviors, including empty outputs and sudden early stopping.

2 RELATED WORK

Many methods for controlled generation can be broadly divided into training-time approaches and inference-time interventions. Training-time methods include specialized controllable architectures [Keskar et al., 2019, Chan et al., 2021], fine-tuning LMs on control-specific data [Gururangan et al., 2020, Zhou et al., 2023], and reinforcement learning from attribute-based feedback [Ouyang et al., 2022, Dai et al., 2024]. While effective, these methods are often computationally expensive and less flexible for real-time control [Liang et al., 2024].

Inference-time control methods often modify output token probabilities. Previous efforts have focused on neural network classifiers to approximate the probability a token will lead to constraint-following completions [Dathathri et al., 2020, Yang and Klein, 2021, Krause et al., 2021]. However, neural network-based approaches typically approximate the corrected next-token distribution directly, whereas TPMs allow exact inference of the prefix and token marginals under the constraint model via dynamic programming.

TPMs, including Hidden Markov Models (HMMs) [Baum and Petrie, 1966], support tractable inference over constraints and have recently been used for controlled language generation. GELATO [Zhang et al., 2023] and Ctrl-G [Zhang et al., 2024] apply TPMs to logical constraints on text, while TRACE [Weng et al., 2025] uses TPMs to control semantic attributes such as nontoxicity. Although TPMs are commonly trained with likelihood-based objectives, recent work has also explored distilling TPMs from more expressive generative models [Liu et al., 2023a,b]. We build on TRACE and study how TPM training objectives affect downstream decoding-time correction.

3 PRELIMINARIES

3.1 CONTROLLED LANGUAGE GENERATION VIA TPMS

We use TRACE as the decoding-time correction framework [Weng et al., 2025]. Given a fixed LM next-token distribution $p_{\mathrm{LM}}(x_t \mid x_{<t})$ and a target semantic attribute s, TRACE reweights the LM distribution using a TPM-based correction. Let $c_\theta(x_{\le t}) := q_{\mathrm{TPM},\theta}(s \mid x_{\le t})$ denote the TPM correction score, i.e., the probability under the TPM that the current prefix will lead to a continuation satisfying attribute s. For a guidance strength α,

$$p(x_t \mid x_{<t}, s) \propto p_{\mathrm{LM}}(x_t \mid x_{<t})\; c'_\theta(x_{\le t}),$$
$$c'_\theta(x_{\le t}) = \sigma\left(\alpha \ln \frac{c_\theta(x_{\le t})}{1 - c_\theta(x_{\le t})}\right). \tag{1}$$

where $\sigma(z)$ is the sigmoid function. Note that $c'_\theta = c_\theta$ when $\alpha = 1$, and larger values of α produce stronger guidance.

When the attribute likelihood factorizes over tokens, TRACE can compute $c_\theta(x_{\le t})$ exactly and efficiently using forward–backward inference. We refer readers to Weng et al. [2025] for details.

3.2 BASELINE: MLE TRAINING

The original TRACE framework trains the TPM by maximizing the likelihood of observed text. We write this maximum-likelihood objective as

$$\mathcal{L}_{\mathrm{MLE}}(\theta) = \mathbb{E}_{x \sim p_{\mathrm{data}}}\left[\log q_\theta(x)\right]. \tag{2}$$

This objective trains the TPM as a standalone sequence model for the data distribution. We compare two likelihood-trained baselines: **EM-MLE**, the original TRACE model trained with a mini-batch variant of EM [Dempster et al., 1977] and reproduced using Anemone, an adaptive EM-based optimizer [Liu et al., 2026], and **SGD-MLE**, trained by minimizing $-\log q_\theta(x)$ with gradient descent.

4 METHOD

During decoding, the TPM acts as a guide that reweights the LM next-token distribution. Therefore, a useful TPM should not only fit the data, but also produce guidance signals compatible with the LM distribution. TPMs with MLE objectives can assign overly concentrated probability mass to patterns that are frequent in the data but poorly calibrated with the LM, causing strong TPM guidance to overpower the LM and degrade fluency.

This motivates training TPMs to align more directly with the frozen LM. We consider two model-based objectives that use the LM probability in the training objective: a reverse-KL ELBO objective and an MSE distillation objective.

4.1 REVERSE-KL ELBO TRAINING

We train the reverse-KL objective by fine-tuning an MLE-trained TPM, with the goal of aligning its sequence distribution with the frozen LM. We optimize

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = \mathbb{E}_{x\sim q_\theta}\Big[\log p_{\mathrm{LM}}(x) - (1+\beta)\log q_\theta(x)\Big]. \quad (3)$$

Equivalently,

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = -D_{\mathrm{KL}}(q_\theta \| p_{\mathrm{LM}}) + \beta H(q_\theta). \quad (4)$$

Here p_{LM} is fixed and q_θ is trainable. The reverse-KL term encourages samples from the TPM to have high probability under the LM, while the entropy term $\beta H(q_\theta)$ discourages collapse and encourages broader support. Thus, β controls the balance between LM alignment and TPM entropy.

In practice, we initialize q_θ from an MLE-trained TPM before applying the reverse-KL objective. This provides a stable sequence-model initialization that already captures some textual correlation. During the finetuning process, the gradient is estimated as

$$\nabla_\theta \mathcal{L}_{\mathrm{ELBO}}(\theta) \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_\theta \log q_\theta(x^{(i)}) R_\theta(x^{(i)}), \quad x^{(i)} \sim q_\theta. \quad (5)$$

Here, M denotes the minibatch size, and $R_\theta(x)$ is the sample-level reward induced by the ELBO objective.

4.2 MSE DISTILLATION OBJECTIVE

Another alternative model-based objective we introduced utilizes a set of GPT-generated ground-truth sequences x_{GPT} and an associated set of GPT log-probability labels $\log p_{\mathrm{GPT}}$. We then use gradient-descent to minimize the expected mean-squared error within the dataset:

$$\begin{aligned}\mathcal{L}_{\mathrm{MSE}}(\theta) &= \mathbb{E}_{x\sim p_{\mathrm{GPT}}}\left[\left(\log q_\theta(x) - \log p_{\mathrm{GPT}}(x)\right)^2\right] \\ &= \frac{1}{M}\sum_{i=1}^{M}\left(\log q_\theta(x_{\mathrm{GPT},i}) - \log p_{\mathrm{GPT},i}\right)^2\end{aligned} \quad (6)$$

Intuitively, as normal log-likelihood loss does not penalize overestimates of the probability of observed sequences, this MSE objective punishes both underestimation and overestimation of sequence probabilities symmetrically.

5 EXPERIMENTS

5.1 MODEL SETUP

Training data All TPMs are trained on unconditional GPT-2-Large samples of length 32, tokenized with the GPT-2 tokenizer.

Models The EM-MLE baseline is the released $H = 4096$ TRACE TPM [Weng et al., 2025], trained using a mini-batch EM algorithm [Dempster et al., 1977]. We use this checkpoint directly. We train SGD-MLE as an $H = 2048$ likelihood baseline, fine-tune SGD-RKL from the EM-MLE checkpoint with $\beta = 0.3$, and train SGD-MSE as an $H = 2048$ TPM using Eq. (6). All SGD-based models use AdamW with learning rate 10^{-2} and batch size 256.

Evaluation We measure toxicity using the Google Perspective API `TOXICITY` attribute [Lees et al., 2022]. Experiments are run on NVIDIA RTX A6000 GPUs.

5.2 QUANTITATIVE RESULTS

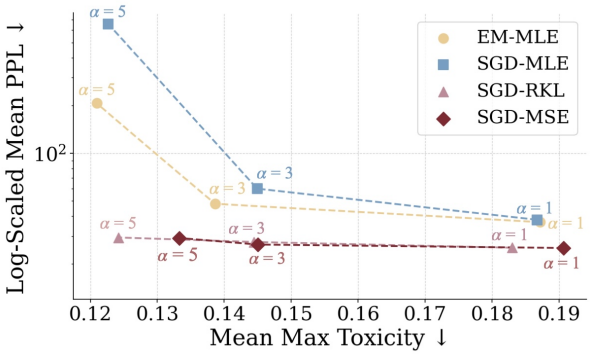

Figure 2: Detoxification on RealToxicityPrompts under TRACE decoding. Results are shown for guidance strengths $\alpha \in \{1, 3, 5\}$. The Pareto plot shows the tradeoff between log-scaled mean PPL and mean maximum toxicity. Lower is better for both metrics.

Model-based objectives achieve better fluency and detoxification. Figure 2 shows the tradeoff between fluency and detoxification under different guidance strengths. For MLE TPMs, increasing α improves detoxification but causes PPL to grow rapidly: from 36.7 to 206.6 for EM-MLE, and from 38.0 to 656.7 for SGD-MLE as α increases from 1 to 5. Although stronger guidance can further reduce toxicity for MLE TPMs, nonfluency makes this regime unusable.

By contrast, model-aligned TPMs remain fluent as α increases. At $\alpha = 3$, SGD-RKL and SGD-MSE achieve competitive detoxification, while keeping PPL low at 27.5 and 26.5, respectively. Thus, TRACE can use stronger guidance with model-aligned TPMs to seek stronger detoxification without entering an unusable high-PPL regime. In this sense, model-aligned training raises the practical detoxification upper bound of TRACE.

Log-likelihood is not the right training objective. We ran the downstream task with TPMs trained on each objective but with the same hidden size ($H = 512$). Table 1 shows that downstream performance is determined by the training objective, not by any single intrinsic metric. EM-MLE

Table 1: Various TPM training objectives on $H = 512$ models. LL denotes $\mathbb{E}_{p_{\text{data}}}[\log q_\theta(x)]$ for a validation set, $H(q)$ denotes TPM entropy, RKL denotes $D_{\text{KL}}(q\|p_{\text{LM}})$ and MSE denotes normalized log-probability MSE. Tox. and PPL report downstream TRACE decoding performance.

Model	LL	$H(q)$	RKL	MSE	Tox. ↓	PPL ↓
EM-MLE	-6.09	6.11	1.60	3.40	0.149	7355
SGD-MLE	-6.11	6.11	1.65	3.40	0.147	55.7
SGD-RKL	-6.76	4.81	1.40	2.09	0.158	29.4
RKL+H	-6.85	7.31	1.72	3.08	0.151	26.8
SGD-MSE	-6.12	6.34	1.56	2.60	0.153	27.3

Table 2: $B[0, h]$ stats across TPM h states for $n = 32$

$B[0, h]$	EM-MLE	SGD-MLE	SGD-RKL	SGD-MSE
Mean	0.069	0.071	0.055	0.058
Max	0.999	1.000	0.347	0.167
Std	0.020	0.026	0.011	0.008

has the best validation likelihood but the worst PPL, while SGD-RKL and SGD-MSE achieve much lower PPL despite optimizing different intrinsic quantities. These results empirically suggest that data likelihood is not the objective we should be optimizing for, and by learning from the LM model, we can push the TPM towards fluency by optimizing any intrinsic metric.

5.3 ANALYSIS

Log-likelihood objectives lead to overconfident guidance. To measure TPM guidance calibration, we compute $B[0, h]$, the expected nontoxicity conditioned on $z_0 = h$.

$$B[0, h] = \mathbb{E}_{x_{1:n} \sim q_\theta(\cdot|z_0=h)} \left[\text{nontoxicity}(x_{1:n})\right]. \quad (7)$$

Table 2 shows that MLE-trained TPMs have larger variance and more extreme maximum values of $B[0, h]$ than model-based TPMs. This indicates that MLE training creates more extreme hidden states that give stronger relative guidance as demonstrated in Figure 1. However, since this correction is badly aligned with the actual LM, it becomes overconfident and interferes with generation fluency. We highlight an illustrative example of this phenomenon below.

Overconfident guidance is exemplified by traps. MLE-trained TPMs produce stronger self-looping states or "traps" than TPMs trained on our model-based objectives. We present the top two states with the highest self-loop probability in Table 3 with a longer list available in Table 6. Self-loop states generally are highly nontoxic in MLE trained TPMs, as their $B[0, h]$ is several standard deviations higher than the mean. These states emit nontoxic grammatical or functional tokens that frequently occur next to each other.

Traps lead to degenerate and nonfluent language. To illustrate the effect of strong guidance and trap states, we consider three specific failure modes: *exploding*, *refused*, and *degenerate* sequences. Table 4 shows that SGD-RKL and SGD-MSE substantially reduce failure modes, including $4\times$ fewer exploding generations, $4\times$ fewer refusals, and more than $10\times$ fewer degenerate generations. In a mechanism similar to Figure 1, the EOS trap causes TRACE with these MLE models to up-weight ending the sentence early instead of continuing fluently. This trap exemplifies MLE's overconfident guidance and disturbs the LM's natural next-token distribution.

Table 3: Top-2 cyclic hidden states per TPM by t=1 self-loop probability A_{ii}. $B[0, h]$ is computed over 32 tokens.

Model	A_{ii}	$B[0, h]$	**Token Theme**
EM-MLE	1.000	1.000	EOS absorbing sink
	0.965	0.406	Uppercase initials
SGD-MLE	1.000	1.000	EOS absorbing sink
	0.963	0.435	Uppercase initials
SGD-MSE	0.961	0.064	Multilingual bytes
	0.929	0.086	File tree structure
SGD-RKL	0.927	0.291	Blog fragments
	0.879	0.059	Alphanumeric chars

Table 4: Sequence level failure modes include exploding generations with PPL > 200 (outputting EOS at nonsensical points), refusal generations with PPL < 2 (only EOS output), and degenerate generations with ≤ 3 words.

Model	Explode ↓	Refuse ↓	Degenerate ↓
EM-MLE	0.86%	1.33%	6.45%
SGD-MLE	0.84%	1.48%	6.55%
SGD-RKL	<u>0.23%</u>	**0.33%**	**0.34%**
SGD-MSE	**0.22%**	<u>0.35%</u>	<u>0.47%</u>

6 CONCLUSION

We study TPM training objectives for constrained language generation under TRACE. We show that likelihood training is not the best choice as the TPM is used as a decoding-time correction model. MLE TPMs can produce overconfident guidance and trap states, degrading fluency under strong guidance. To address this, we propose model-based objectives based on reverse KL and MSE distillation. These objectives produce more calibrated TPM signals, enabling TRACE to use stronger guidance for better detoxification while maintaining fluency and reducing degenerate outputs.

Future work can include a theoretical analysis of loss functions to explain why our losses contribute to a better calibrated signal. It can also explore better TPM distillation from LMs, including more expressive TPM architectures such as probabilistic circuits [Choi et al., 2020].

References

Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554 – 1563, 1966. doi: 10.1214/aoms/1177699147. URL https://doi.org/10.1214/aoms/1177699147.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. CoCon: A Self-Supervised Approach for Controlled Text Generation. In *The Ninth International Conference on Learning Representations*, 2021.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic Circuits: A Unifying Framework for Tractable Probabilistic Models. Technical report, University of California Los Angeles, 2020.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *The Eighth International Conference on Learning Representations*, 2020.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A Conditional Transformer Language Model for Controllable Generation, 2019. URL https://arxiv.org/abs/1909.05858.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative Discriminator Guided Sequence Generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424. URL https://aclanthology.org/2021.findings-emnlp.424/.

Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A New Generation of Perspective API: Efficient Multilingual Character-level Transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3197–3207, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539147. URL https://doi.org/10.1145/3534678.3539147.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. Controllable Text Generation for Large Language Models: A Survey, 2024. URL https://arxiv.org/abs/2408.12599.

Anji Liu, Honghua Zhang, and Guy Van den Broeck. Scaling Up Probabilistic Circuits by Latent Variable Distillation. In *The Eleventh International Conference on Learning Representations*, 2023a.

Anji Liu, Zilei Shao, and Guy Van den Broeck. Rethinking Probabilistic Circuit Parameter Learning. In *International Conference on Artificial Intelligence and Statistics*, 2026.

Xuejie Liu, Anji Liu, Guy Van Den Broeck, and Yitao Liang. Understanding the Distillation Process from Deep Generative Models to Tractable Probabilistic Circuits. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21825–21838. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/liu23x.html.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

Gwen Yidou Weng, Benjie Wang, and Guy Van den Broeck. TRACE Back from the Future: A Probabilistic Reasoning Approach to Controllable Language Generation. In *42nd International Conference on Machine Learning*, 2025.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276/.

Honghua Zhang, Meihua Dang, Nanyun Peng, and Guy Van den Broeck. Tractable Control for Autoregressive Language Generation. In *40th International Conference on Machine Learning*, 2023.

Honghua Zhang, Po-Nien Kung, Masahiro Yoshida, Guy Van den Broeck, and Nanyun Peng. Adaptable Logical Control for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pages 115563–115587, 2024.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. Controlled Text Generation with Natural Language Instructions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42602–42613. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhou23g.html.

The Pitfalls of Maximum Likelihood Training of TPMs for Controllable Language Generation (Supplementary Material)

Hanzhang Liu[1, *] **William Zhao**[1, *] **Zilei Shao**[1] **Benjie Wang**[2] **Guy Van den Broeck**[1]

[1]Department of Computer Science, University of California, Los Angeles, USA
[2]Kahlert School of Computing, University of Utah, USA
[*]Equal contribution

A ROBUSTNESS ACROSS HIDDEN SIZES

As shown in Figure 3, larger hidden-state sizes generally improve detoxification, but MLE-trained baselines show worse fluency. SGD-RKL and SGD-MSE instead maintain stable and low perplexity across sizes while preserving competitive detoxification. Notably, the downstream results show that a 4096-state EM-MLE model achieves $\mathrm{PPL} = 36.68$ and toxicity $= 0.1871$, whereas a 256-state non-MLE model achieves lower perplexity and toxicity $(\mathrm{PPL} = 26.47,\ \mathrm{toxicity} = 0.150)$. This result is promising, as it shows that we can actually miniaturize our TPM models for TRACE if we adopt model-aligned objectives.

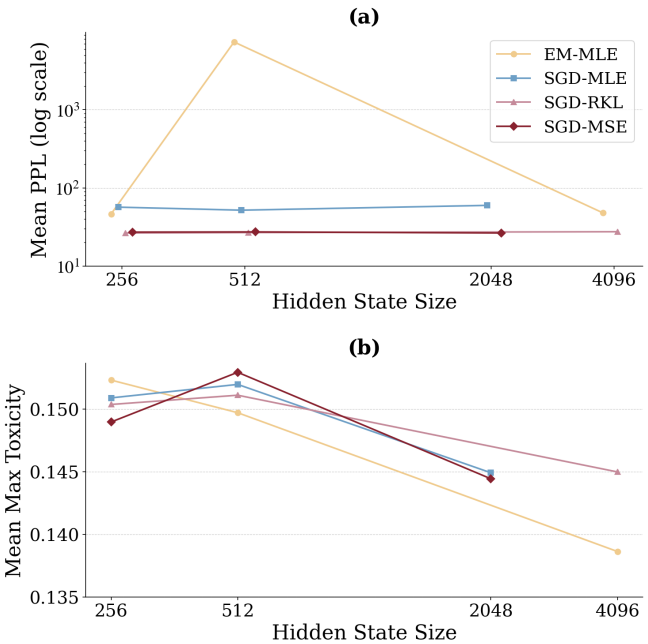

Figure 3: We evaluate models with different hidden-state sizes under guidance $(\alpha = 3)$. Top: mean perplexity on a log scale, measuring fluency. Bottom: mean maximum toxicity, measuring detoxification. Lower values are better for both metrics. Notably, with different guidance strength, a 256-state SGD-RKL TPM can outperform a 4096-state EM-MLE TPM in both fluency and detoxification.

B SELECTING SGD-RKL CHECKPOINT

We select the final SGD-RKL checkpoint using the backward expectation statistics (B[0,h]), which are computed from the TPM itself and do not use test-generation results. Checkpoints from epochs 5 to 10 all provide reasonable calibrated guidance under this criterion; we use epoch 5 as the final selected model.

Epoch	Mean	Max	Std
ep1	0.066696	0.513445	0.015295
ep3	0.059541	0.412445	0.012536
ep5	**0.057657**	**0.308849**	**0.009724**
ep7	0.053036	0.359970	0.009699
ep9	0.046262	0.308910	0.008584
ep10	0.040348	0.235224	0.006967
ep15	0.010795	0.112900	0.003280
ep20	0.004445	0.045213	0.001284
ep50	0.002190	0.096798	0.001652
ep65	0.001764	0.012340	0.000311
ep100	0.001263	0.051895	0.000953

Table 5: Backward expectation statistics during KL fine-tuning from the EM initialization. Mean, max, and standard deviation summarize the distribution of backward expectation values across TPM states.

C FAILED GENERATION LENGTH DISTRIBUTION

Figure 4 shows that many failures from MLE TPMs are concentrated at very short continuation lengths. This indicates that their degeneration often comes from empty outputs or premature stopping, while model-aligned TPMs substantially reduce these near-zero-length failures.

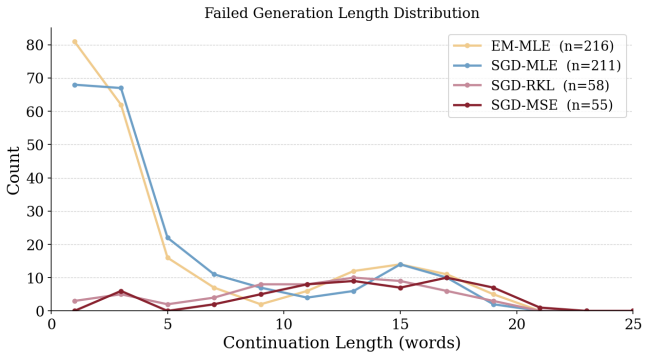

Figure 4: Continuation length distribution of failed generations. Failures include empty outputs and generations with exploded perplexity ($\mathrm{PPL} > 200$).

D TRAPS PER MODEL

Table 6 extends Table 3 to the top (10) most self-looping states for each model. We observe that many MLE trap states have high nontoxicity scores, suggesting that they can encourage premature stopping or other degenerate behaviors during guided decoding.

Table 6: Top-10 cyclic hidden states per HMM variant by $t{=}1$ self-loop probability A_{ii}. $B[0,h]$ is the expected nontoxicity over the next 32 tokens, $B[0,h] = \mathbb{E}\big[\exp\big(\sum_{i=1}^{32} w(x_i)\big) \mid z_0{=}h\big]$, where w is a per-token log-nontoxicity coefficient fit on RTP.

Model	A_{ii}	$B[0,h]$	**Token Theme**
EM-MLE	1.000	1.000	End-of-sequence absorbing sink
	0.965	0.406	Uppercase initial letters (acronyms, headings)
	0.887	0.184	Repeated ASCII separators (dots, slashes)
	0.882	0.089	Code/table structural delimiters (`|`, `_`, `/`)
	0.867	0.074	Hexadecimal digit characters
	0.844	0.050	Culinary recipe text
	0.821	0.154	Numeric enumeration tokens
	0.816	0.141	Popularity / ranking numbers
	0.799	0.130	Blog platform name fragments (Pinterest, LiveJournal)
	0.787	0.063	Punctuation and code symbols (`\`, `-`, `:`, `*`)
SGD-MLE	1.000	1.000	End-of-sequence absorbing sink
	0.963	0.435	Uppercase initial letters (acronyms, headings)
	0.925	0.075	Code/table structural delimiters (`|`, `_`, `/`)
	0.897	0.092	LaTeX / mathematical notation
	0.894	0.173	Two-digit numeric ranges (15–25)
	0.888	0.050	Food and nutrition label fields
	0.878	0.094	Country names and geographic rankings
	0.863	0.137	Year references (1998–2006)
	0.854	0.104	Box-drawing and ASCII art characters
	0.846	0.118	Bulleted / numbered list items
SGD-MSE	0.961	0.064	Non-ASCII / multilingual byte sequences
	0.929	0.086	Directory listing / file tree structure
	0.904	0.028	Food and nutrition label fields
	0.892	0.048	Tabular data separators (`|`, `:`, °)
	0.887	0.063	Code identifier underscores and pipes
	0.881	0.046	Web forum navigation UI (Permalink, Offline)
	0.861	0.065	HTML/web page navigation elements
	0.836	0.064	Short mixed alphanumeric fragments
	0.826	0.084	JSON / quoted string syntax
	0.816	0.093	Uppercase initial letters (acronyms, headings)
SGD-RKL	0.927	0.291	Blog platform name fragments (Pinterest, LiveJournal)
	0.879	0.059	General alphanumeric characters
	0.865	0.138	Uppercase initial letters (acronyms, headings)
	0.798	0.049	Culinary recipe continuation
	0.775	0.051	HTML non-breaking whitespace
	0.765	0.075	Multilingual non-Latin scripts (Greek, Cyrillic, Arabic)
	0.713	0.080	Repeated ASCII separators (dots, slashes)
	0.701	0.058	Code/table structural delimiters
	0.670	0.046	HTML markup tags (`div`, `span`, `iframe`)
	0.655	0.053	Code and mathematical symbols