Beyond Model Collapse: Scaling Up with Synthesized Data Requires Reinforcement

Yunzhen Feng [*1,2] **Elvis Dohmatob** [*1] **Pu Yang** [*3] **Francois Charton** [1] **Julia Kempe** [1,2,4]

Abstract

Synthesized data from generative models is increasingly considered as an alternative to human-annotated data for fine-tuning Large Language Models. This raises concerns about model collapse: a drop in performance of models fine-tuned on generated data. Considering that it is easier for both humans and machines to tell between good and bad examples than to generate high-quality samples, we investigate the use of feedback on synthesized data to prevent model collapse. We derive theoretical conditions under which a Gaussian mixture classification model can achieve asymptotically optimal performance when trained on feedback-augmented synthesized data, and provide supporting simulations for finite regimes. We illustrate our theoretical predictions on news summarization with large language models. We show that training from feedback-augmented synthesized data, either by pruning incorrect predictions or by selecting the best of several guesses, can prevent model collapse, validating popular approaches like RLHF.

1. Introduction

As generative models for language (Touvron et al., 2023; Achiam et al., 2023), images (Ramesh et al., 2021; Rombach et al., 2022), and video (OpenAI, 2024) achieve human-level performance, a significant fraction of the training data for future models will be generated by previous models. ChatGPT alone generates 0.1% of the tokens currently produced by humans (Altman, 2024). There is an increasing use of AI-synthesized data in diverse domains such as coding (Haluptzok et al., 2022) and mathematics (Trinh et al., 2024) and strong language models are tauted as a possible replacement for expensive human annotators.

[*]Equal contribution [1]Meta FAIR [2]Center for Data Science, New York University [3]School of Mathematical Sciences, Peking University [4]Courant Institute, New York University. Correspondence to: Yunzhen Feng <yf2231@nyu.edu>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024.

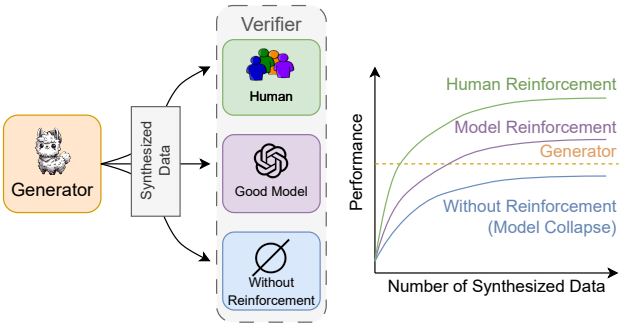

Figure 1. We propose using a verifier to select generated synthesized data. Human and model reinforcement can enhance performance and prevent model collapse, as opposed to the degradation observed without reinforcement.

This gradual replacement of human-written corpora by machine-generated tokens gives rise to a number of concerns, notably the risk of "model collapse" (Shumailov et al., 2023), where iterated training on synthesized data brings a drop in model performance, and, ultimately, "dumber models". This phenomenon was observed empirically (Hataya et al., 2023; Martínez et al., 2023a;b; Bohacek and Farid, 2023; Briesch et al., 2023; Guo et al., 2023) and described theoretically (Alemohammad et al., 2023; Bertrand et al., 2023; Dohmatob et al., 2024a). Its main consequence is the breaking of known scaling laws (Dohmatob et al., 2024b): as data becomes more synthetic, larger training sets do not enhance performance.

Meanwhile, we are witnessing the massive use of Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022) and its variants, which leverages human feedback and annotation to train models well past their performance on scraped internet data. As the performance of language models improves, their use as feedback generators, replacing human annotators, is increasingly considered. This leads us to ask the timely question:

> Can feedback, from humans or machines, improve synthesized data, to the point that it can be used to train new models without fear of collapse?

To study this question, we first provide analytical (affirmative) results in a theoretical setting, where we consider Gaussian mixtures (and generalizations) in the high-dimensional limit with linear models as classifiers. We allow a possibly

noisy verifier (e.g. human or oracle) to select (or prune) generated data. We demonstrate that as the number of synthesized data points approaches infinity, the model trained on selected data can achieve optimal results, on par with training on the original data. Specifically, we identify a sharp phase transition: from zero accuracy due to errors in the synthesized data and verifier, to optimal accuracy. We conduct simulations on synthesized data to explore how the generator and verifier affect performance and scaling rates in finite regimes. Also here, our results show that oracle supervision consistently yields near-optimal results compared to using original labels. Since discerning high-quality data through human supervision is simpler and more cost-effective than direct human labeling, this provides strong evidence for the efficacy of human-in-the-loop supervision.

We next examine realistic settings to illustrate our theory on news summarization with large language models (Llama2). In this setting, reliance solely on generated data results in poorer performance compared to using the original dataset, even with increased data volume, indicating model collapse. Conversely, with oracle supervision, we achieve an improved synthesized dataset that surpasses the original, with performance improving as more data is added.

We summarize our contribution as follows:

- We provide theoretical analysis to characterize when data selection leads to optimal performance in the high-dimensional limit with unlimited synthesized data. Simulations on synthesized data extend this to finite-data regimes to show that oracle (human) selection can match training with original labels (Section 2).
- We validate these observations with news summarization using LLaMA-2. Model collapse is observed, oracle selection prevents it and improves the selected dataset beyond the synthesized generator (Section 3).

Crucially, note that all that is needed to re-attain model performance akin to training on clean original data is the ability to *distinguish* high-quality from low quality labels; arguably, a task much simpler than annotating the labels. Thus, to go beyond model collapse and continue scaling up with synthesized data, *reinforcement is all you need!*

Related Work. For a more extensive reference list, see Appendix A on data selection and a taxonomy of benefits of synthesized data.

Model Collapse. With the advancement of generative models, synthesized data generated by these models has become increasingly prevalent online, mixing irreversibly into our training corpora. Recent studies have highlighted the potential for dramatic deterioration in downstream models, a phenomenon known as *"model collapse"* (Shumailov et al., 2023). Empirical studies have demonstrated this issue in various settings (Hataya et al., 2023; Martínez et al., 2023a;b; Bohacek and Farid, 2023; Briesch et al., 2023). Synthesized datasets have been shown to reduce diversity (Padmakumar and He, 2024; Guo et al., 2023) and cause distributional distortions (LeBrun et al., 2021). Theoretical analyses also examine the effects of iterative training on self-generated data (Alemohammad et al., 2023; Bertrand et al., 2023; Dohmatob et al., 2024a; Seddik et al., 2024). Notably, (Dohmatob et al., 2024b) warns that model collapse signifies a break in the scaling law, where increasing synthesized data volume does not enhance performance effectively. (Gillman et al., 2024) propose to use correction function with expert knowledge to modify the synthetic data to prevent model collapse. In this work, we aim to apply selection techniques to large synthesized datasets to surpass the quality of the original data that trained the generator.

Synthesized Data with Selection. Empirical studies have demonstrated that applying selection techniques to synthesized data can significantly enhance performance, particularly in the domains of code and mathematics where good verifiers for correctness exist. For instance, (Haluptzok et al., 2022) generate synthesized code data and filter out incorrect instances. (Ulmer et al., 2024) leverage conversational metrics to filter synthetic dialogue data. (Trinh et al., 2024) leverage a symbolic deduction engine as a verifier to sample correct solutions for Olympiade geometry problems. Oracle reinforcement and abundant synthesized input lead to near-optimal performance. When a verifier does not exist, (Li et al., 2022) use a high-quality dataset to train a verifier to select data for self-labeling. Additionally, some studies achieve data selection by carefully choosing prompts with high quality and good diversity, employing heuristic verifiers: instruction tuning (Wang et al., 2023), code generation (Wei et al., 2023), and image synthesis (Hemmat et al., 2023; Azizi et al., 2023).

2. Theoretical Insights

We theoretically characterize under what conditions data selection with reinforcement can lead to improvements, for a family of high-dimensional data distributions. Note that we model the reinforcement process as a *pruning strategy* over synthesized data. Crucially, we will not necessarily assume that the pruning strategy has access to the ground truth; rather, we will formulate our theory in sufficient generality to allow for "intermediate" pruners, which can be viewed as reinforcement from a different (or even the same) model. A full exposition of our general theory is provided in Appendix F; for ease of exposition, we specialize here to Gaussian Mixtures, with more details on those in Appendix E.

Data Distribution. We consider distributions P over $\mathbb{R}^d \times \{0, 1\}$. For binary *Gaussian Mixtures*, features are given by $x \mid y \sim N(\mu_y, \Sigma)$, where $\mu_y = (2y - 1)\mu$, for some $\mu \in \mathbb{R}^d$ and Σ is a positive-definite matrix with $\mathbb{E}\,\|x\|^2 =$

$\|\mu\|_2^2 + \operatorname{tr}\Sigma = 1$. For further ease of exposition we will only consider balanced distributions $\mathbb{P}(y = 1) = \mathbb{P}(y = 0) = 1/2$, for $(x, y) \sim P$.

Synthesized Data. Let $D_N = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a dataset of N iid pairs from the true distribution P and let $D'_N = \{(x_1, y'_1), \ldots, (x_N, y'_N)\}$ be the synthesized data generated from the same distribution, but where label y'_i (instead of y_i) has been generated by an AI model.

Downstream Model and Pruning. We model our data selection (with or without feedback) via a *pruning strategy* $q = (q_1, \ldots, q_N)$ where q_i is a bit which indicates whether the ith training example from D'_N has survived pruning. For the downstream models we consider the family:

$$\mathbb{P}(y = 1 \mid x, w) = \hat{y} := \sigma(x^\top w) \in (0, 1),\ \sigma(z) := \frac{1}{1 + e^{-z}}$$

parametrized by a vector of weights $w \in \mathbb{R}^d$ and sigmoid non-linearity σ. Let $\widehat{w}_N$ be obtained via logistic regression fitted on D'_N with ridge regularization parameter $\lambda > 0$. Thus, $\hat{w}$ minimizes the following objective function

$$L(w) := \frac{1}{N} \sum_{i=1}^{N} q_i \ell(\sigma(x_i^\top w), y'_i) + \frac{\lambda}{2}\|w\|^2,$$

where ℓ is the binary cross-entropy. The corresponding downstream classifier is $\widehat{f}_N = f_{\widehat{w}_N}$, where the notation f_w refers to the linear classifier induced by a weights vector $w \in \mathbb{R}^d$, i.e $f_w(x) = (\operatorname{sign}(x^\top w) + 1)/2$.

Test Accuracy. The test accuracy of the downstream model $\widehat{f}_N$ is defined by

$$acc(\widehat{f}_N) := \mathbb{P}(\widehat{f}_N(x) = f_{Bayes}(x)),$$

for a random test point $(x, y) \sim P$, where $f_{Bayes}(z) := \mathbb{E}[y|x = z]$ is the Bayes-optimal classifier. Note that $acc(f_{Bayes}) = 100\%$. The quantity $acc(\widehat{f}_N)$ will be the main object of our analysis, and we will be interested in how it depends on errors in the generator P and the choice of pruning strategy q, in the infinite-sample limit $N \to \infty$.

"RLHF" Pruning Strategy. We consider a wide class of parametrized pruning strategies q, which we term *RLHF-Pruning*, that satisfy the following reasonable property: *The bits $q_1, \ldots, q_N \in \{0, 1\}$ are independent.* We shall denote by $p \in [0, 1)$, the probability that the label y'_i of a synthesized example (x_i, y'_i) is different from the true label y_i. A *symmetric* (ϕ, ψ)-RLHF pruning strategy is parametrized by (ϕ, ψ) defined as: $\phi = \mathbb{P}(q_i = 1 \mid y'_i = y_i)$ and $\psi = \mathbb{P}(q_i = 1 \mid y'_i \neq y_i)$.

Supervised Pruning: $q_i = 1[y_i(x_i^\top w_{prune}) > 0]$, (1)

for some weights $w_{prune} \in \mathbb{R}^d$ is a special case (see Appendix F on how to obtain (ϕ, ψ) in this case). This pruning strategy filters out all examples on which there is disagreement on the assigned label.

Oracle Pruning. The case $(\phi, \psi) = (1, 0)$. We only keep indices corresponding to examples in the dataset which have correct label (all corrupted labels are discarded).

Insights from Infinite-Sample Regime The following is our main theoretical result (see Theorem F.3 for full statement). It characterizes test accuracy $acc(\widehat{f}_N)$ of the downstream model on pruned data as a function of p (the label disagreement) and the parameters (ϕ, ψ) of the pruner, in the theoretical limit of infinite training data ($N \to \infty$).

Theorem 2.1 (Simplified version of Theorem F.3)**.** *Define the breakdown point $p_\star \in (0, 1)$ by $p_\star := 1/(1 + \psi/\phi)$. In the limit $N \to \infty$ it holds a.s that:*

(i) If $p < p_\star$ then $acc(\widehat{f}_N) = 100\%$.

(ii) If $p > p_\star$ then $acc(\widehat{f}_N) = 0\%$.

Thus, there is a phase-transition around the corruption level $p_\star := 1/(1 + \psi/\phi)$: as p is increased past level $p_\star$, the downstream model $\widehat{f}_N$ abruptly switches from being perfectly accurate, to perfectly inaccurate! The proof computes empirical test accuracy in terms sums of Bernoulli variables corresponding to constellations of flipped labels, which follow a binomial distribution, bounding the gap to the population accuracy, and using concentration of measure type techniques. Note that the sharp transition is due to the infinite-sample regime, where we can avoid finite-sample corrections and the 100% accuracy achievable in the Theorem is idealized, and is expected to only hold in infinite sample regime (with large but fixed input dimension).

Some Consequences: Supervised Pruning. Here, parameters (ϕ, ψ) only depend on the angles $\theta_{gen}, \theta_{prune}, \theta \in [0, \pi]$ given by

$$\theta_{gen} := \angle(w_{gen}, \mu),\ \theta_{prune} := \angle(w_{prune}, \mu),$$
$$\theta := \angle(w_{prune}, w_{gen}).$$

This is because ϕ and ψ now correspond to *orthant probabilities* for a trivariate normal distribution, with correlation coefficients are given by these angles (see also Figure 5).

Although the generator and verifier are coupled together, there are some intuitions that help us decouple them: (1) a better generator always improves performance, (2) when the verifier is poor, such as in cases of no pruning or random pruning, we have a low breakdown point and require a good generator, and (3) when the verifier is sufficiently good, close to an oracle, the breakdown point is high, and any non-degenerate generator is sufficient, for example when $\psi/\phi = 0$, $\widehat{f}_N$ achieves 100% accuracy for any $p < 1$.

Simulations on Synthesized Data. Here we show simulations in finite regimes. We initially sample N_0 data from the distribution P_{orig} as the original dataset, D_{orig}, which is used to train a linear model $\hat{w}$ using ordinary least squares. Subsequently, we use $w_{gen} = \hat{w}$ to generate N_1 synthesized data points with sigmoid, constituting the dataset D_{gen}. The data is selected with various w_{prune} in Equation (1) from w_* to w_θ where θ is the angle between w_θ and w_*. n' is the number of data points selected.

In Figure 3, we run several simulations with different N_0. A larger N_0 corresponds to a better generator trained with more data. The synthesized data is selected using verifiers ranging from oracle-level accuracy to various levels of errors with $\theta_{prune} = \frac{\pi}{12}$ and $\frac{\pi}{6}$. A larger θ_{prune} corresponds to a worse verifier. We have the following observations:

Oracle Supervision Matches Training with Oracle Labels. The oracle achieves the best performance, matching training with clean data across all settings and attaining Bayes optimal accuracy, as predicted by theory.

Weak supervision. Weak supervision results in poorer performance, reflecting the decaying threshold points outlined in theory. When the generator is sufficiently accurate, using weak supervision may harm performance due to the selection of incorrect data points.

3. LLMs for News Summarization

Our experiments on arithmetic problems are given in Appendix C. Here, we proceed to empirical evaluations using Llama-2-7B (Touvron et al., 2023) and Llama-3-8B (Meta, 2024). Our experiments utilize the English summarization subset of the XLSUM dataset (Hasan et al., 2021), which includes 307,000 training samples and 11,500 test samples. Each sample in this dataset pairs a news article with a professionally annotated summary.

For our experiments, Llama-2 is fine-tuned on 12.5% of the training data. This fine-tuned model serves as the generator for creating summaries across the entire training set, forming our synthesized dataset. All the finetuning is with full parameter tuning with one epoch. Throughout all phases of evaluation and generation, we employ greedy decoding to ensure quality generation. The model's performance is assessed using the Rouge-1 metric (Lin, 2004).

In line with our theory, we consider three settings: (1) **Selection with Oracle**: We calculate the Rouge score between the generated summary and the ground truth summary, keeping the data with the highest scores; (2) **Selection with Weak Supervision**: We leverage a fine-tuned Llama-3 model with higher performance than the generator and keep the data with the lowest perplexity; (3) **Self-Selection**: We use the generator to keep the data with the lowest perplexity. We report the result with selection rate 12.5% in Figure 2.

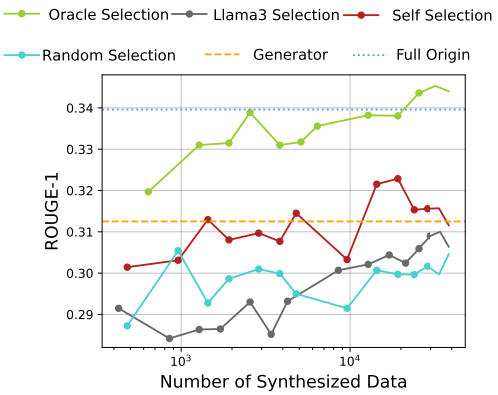

Figure 2. Results of news summarization experiments: models trained on 12.5% of the data. Besides three selection curves, we also include random selection, the Rouge score of the generator model ('generator'), and the Rouge score of a model trained with 100% data with original labels ('Full Origin').

Model Collapse. In Figure 2 **Left**, using the same amount of synthesized data results in worse performance compared to using the original data, indicating model collapse (comparing 'Random Selection' with 'Generator'). Only with more data, the Random selection lines improve and nearly match the performance of the generator.

Selection by Oracle. Employing an oracle for selection yields the best results. Oracle selection even surpasses the model trained with 100% of the training set and original labels, with only 1/8 of the data and training compute.

A Verifier Model with Higher Performance is Not Always Better. Self-selection surprisingly leads to better performance than the generator. We hypothesize that it tends to select easy-to-learn samples. In contrast, Llama-3 results in performance similar to random selection but worse than self-selection. This outcome aligns with theoretical expectations, where the effectiveness of a weak supervisor depends on the angle between all three vectors, as discussed in Section E.4. Although the model has higher performance, it shows little correlation with the generator (θ is larger), which implies that ψ/ϕ might not be better.

4. Conclusion

In this paper, we consider how to prevent model collapse through data selection. We propose to leverage feedback from a verifier to reinforce the synthesized data. We emphasize that when training new models with synthesized data, it is crucial to focus not only on the quality of the generator but also on having a high-quality verifier to select the data. Our work is of significant theoretical and practical importance in the era of large models with increasing use of synthesized data.

Acknowledgements

YF and JK acknowledge support through NSF NRT training grant award 1922658. YF and PY would like to thank Yanzhu Guo, Di He, Zhenyu He for discussions and suggestions. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go mad. *arXiv preprint arxiv:2307.01850*, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Sam Altman. openai now generates about 100 billion words per day. all people on earth generate about 100 trillion words per day. `https://x.com/sama/status/1756089361609981993?lang=en`, 2024.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. On the stability of iterative retraining of generative models on their own data. *arXiv preprint arxiv:2310.00429*, 2023.

Matyas Bohacek and Hany Farid. Nepotistically trained generative-ai models collapse, 2023.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. Large language models suffer from their own output: An analysis of the self-consuming training loop, 2023.

Max F Burg, Florian Wenzel, Dominik Zietlow, Max Horn, Osama Makansi, Francesco Locatello, and Chris Russell. Image retrieval outperforms diffusion models on data augmentation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

François Charton. Linear algebra with transformers. *arXiv preprint 2112.01898*, 2022.

Rudrajit Das and Sujay Sanghavi. Understanding self-distillation in the presence of label noise. In *International Conference on Machine Learning*, pages 7102–7140. PMLR, 2023.

Elvis Dohmatob, Yunzhen Feng, and Julia Kempe. Model collapse demystified: The case of regression. *arXiv preprint arXiv:2402.07712*, 2024a.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv preprint arXiv:2402.07043*, 2024b.

Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. Distillation $\sim$ early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network. *arXiv preprint arXiv:1910.01255*, 2019.

Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in Neural Information Processing Systems*, 36, 2023.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018.

Nate Gillman, Michael Freeman, Daksh Aggarwal, HSU Chia-Hong, Calvin Luo, Yonglong Tian, and Chen Sun. Self-correcting self-consuming loops for generative model training. In *Forty-first International Conference on Machine Learning*, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. The curious decline of linguistic diversity: Training language models on synthetic text, 2023.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. Language models can teach themselves to program better. In *The Eleventh International Conference on Learning Representations*, 2022.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-acl.413.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. Will large-scale generative models corrupt future datasets? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20555–20565, October 2023.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. IS SYNTHETIC DATA FROM GENERATIVE MODELS READY FOR IMAGE RECOGNITION? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nUmCcZ5RKF.

Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *arXiv preprint arXiv:2310.00158*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Advances in neural information processing systems*, 21, 2008.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

Germain Kolossov, Andrea Montanari, and Pulkit Tandon. Towards a statistical theory of data selection under weak supervision. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HhfcNgQn6p.

Benjamin LeBrun, Alessandro Sordoni, and Timothy J O'Donnell. Evaluating distributional distortion in neural language modeling. In *International Conference on Learning Representations*, 2021.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

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

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Combining generative artificial intelligence (ai) and the internet: Heading towards evolution or degradation? *arXiv preprint arxiv: 2303.01255*, 2023a.

Gonzalo Martínez, Lauren Watson, Pedro Reviriego, José Alberto Hernández, Marc Juarez, and Rik Sarkar. Towards understanding the interplay of generative artificial intelligence and the internet. *arXiv preprint arxiv: 2306.06130*, 2023b.

David McAllester. Simplified pac-bayesian margin bounds. In *Learning Theory and Kernel Machines*. Springer Berlin Heidelberg, 2003.

Meta. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/, 2024.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33: 3351–3361, 2020.

OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Vishakh Padmakumar and He He. Does writing with language models reduce content diversity? In *International Conference on Learning Representations (ICLR)*, 2024.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR, 18–24 Jul 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. How bad is training on synthetic data? a statistical analysis of language model collapse. *arXiv preprint arXiv:2404.05090*, 2024.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arxiv:2305.17493*, 2023.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, and Yi Zhang. Bootstrapping llm-based task-oriented dialogue agents via self-talk. *arXiv preprint arXiv:2401.05033*, 2024.

Soobin Um, Suhyeon Lee, and Jong Chul Ye. Don't play favorites: Minority guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3NmO9lY4Jn.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Source code is all you need. *arXiv preprint arXiv:2312.02120*, 2023.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. RLCD: Reinforcement learning from contrastive distillation for LM alignment. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v3XXtxWKi6.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024.

Limitations

One limitation of this study is that we only considered data selection as a means of data curation. Besides data selection, data curation also includes methods such as data augmentation, data regeneration, and weighting. The exploration of general data curation methods to avoid model collapse is left for future work. Our experiments also did not consider the impact of prompt engineering on the generator. This can significantly enhance the generation quality, and according to theoretical predictions, it would be beneficial for synthesized data.

Broader Impact

The study explores the use of synthesized data from generative models as a cost-effective alternative to human-labeled data, aiming to address concerns about model collapse where performance degrades despite increased data volumes. By investigating data selection through reinforcement, the study theoretically and empirically demonstrates that human or oracle supervision can enhance the quality of synthesized datasets, preventing model collapse and potentially surpassing the original dataset's performance. This approach holds significant social impact by proposing a scalable, efficient method to leverage AI-generated data across various domains, reducing dependency on extensive human labeling efforts and mitigating the risks associated with iterative training on synthesized data. The reinforcement-guided data selection can foster more robust and reliable AI models, ultimately contributing to advancements in diverse fields such as language processing, image recognition, and beyond. However, this method also introduces potential negative social impacts. If the verifier, crucial for distinguishing high-quality from low-quality synthesized data, is compromised, it could lead to the propagation of erroneous or malicious data throughout the model. Such vulnerabilities could be exploited through poisoning or targeted attacks, resulting in significant biases, misinformation, and harmful outcomes in AI-driven decisions, particularly in sensitive sectors like healthcare, finance, and law enforcement. Moreover, the reliance on synthesized data could amplify existing biases, reduce cultural and cognitive diversity, and erode public trust in AI technologies, increasing societal vulnerabilities to cyber threats and compromising the perceived reliability and fairness of AI systems.

A. More Works on Synthesized Data

A.1. Taxonomy for Synthesized Data

Contrary to the phenomenon of model collapse, synthesized data has been shown to improve performance in numerous empirical studies. We now provide a taxonomy outlining when and how synthesized data is beneficial. Specifically, we identify four key components: *prompt engineering* ●, *knowledge from advanced models* ▲, *distributional shift and targeted curation* ■, and *external verifiers* ✦. Most empirical studies can be categorized based on one or more of these components. We use ●▲■and ✦to denote the components each reference leverages.

Code and Math. (Haluptzok et al., 2022) ✦generate synthesized data for codes and use a verifier to filter and show that the model can "self-improve" with its own synthesized data. (Gunasekar et al., 2023) ●■filter high-quality data from the web and prompt GPT-3.5 with a specially curated prompt set covering both quality and diversity. (Wei et al., 2023) ●leverage a diverse and large set of open-source code snippets to curate code instructions as prompts with good coverage and high quality. (Zheng et al., 2024; Trinh et al., 2024) ✦leverage a symbolic deduction engine as a verifier to test the correctness of each branch for solving Olympic geometry problems.

Alignment. During standard fine-tunings, synthesized data is often generated by a stronger model like GPT-4 (Peng et al., 2023) ▲. (Wang et al., 2023) ●✦use a good set of prompts and inputs with a heuristic verifier to filter out low-quality ones and maintain high diversity. (Bai et al., 2022) ●■use the model itself to critique whether its own generation is harmful, given already harmful prompts with gold standards from humans. For alignment with reinforcement learning, (Ouyang et al., 2022) ●✦use humans as verifiers to compare synthesized data generated by the current model with a good set of prompts. Some papers propose reinforcement learning with AI feedback (RLAIF) (Lee et al., 2023) ●■that leverages another LLM as the verifier to match human verification. The verifier is a stronger model, instruct-tuned Palm2 L, while the network being trained is the Palm2 XS. However, (Yang et al., 2024) ●later found that using better prompts (self-improve) that direct harmful or harmless responses can surpass RLAIF. (Yuan et al., 2024) ●achieve surprising results with iterative fine-tuning and generating good prompts with in-context learning.

Knowledge distillation. Most papers in the knowledge distillation area involve using a better model to distill for general performance or specific tasks, with ●, ▲, and ■involved from case to case. One example is the tiny story cases (Eldan and Li, 2023) ●▲, where GPT-4 is prompted to generate stories for four-year-olds that are used to train GPT-Neo with good performance.

Image Domain. (Kirillov et al., 2023) and (Li et al., 2022) ■use a distributional shift from high-quality to low-quality data to label and curate a vast amount of unlabeled data. Specifically, (Li et al., 2022) also trains a verifier to filters high-quality data. (Um et al., 2024) ▲■specifically curate minority groups with a diffusion model to enhance performance. (He et al., 2023; Dunlap et al., 2023) ▲■generate synthesized data that aids in classification tasks by tailoring the synthesized data to match the model's learning objectives. (Azizi et al., 2023; Hemmat et al., 2023) ▲■employ guided curation (with supervision) to curate data from diffusion models. (Burg et al., 2023) find that while synthesized data from a diffusion model helps improve downstream tasks, such as classification, using the pre-training data of the diffusion model alone gives even stronger performance.

A.2. Knowledge Distillation with Soft Labels

Related to synthesized data, there is a long history of using synthesized labels in image classifications. In the domains of self-distillation and knowledge distillation (Hinton et al., 2015; Furlanello et al., 2018), data with soft labels generated from the teacher model can significantly improve the performance of the student model. These soft labels convey additional insights—referred to as 'dark knowledge'—that have been theoretically linked to specific advantageous adaptations. These include implicit biases that mitigate overfitting (Mobahi et al., 2020), mimicry of early stopping (Dong et al., 2019) for improved optimization under label noise (Das and Sanghavi, 2023), and adjustments to accommodate specific data structures (Allen-Zhu and Li, 2022). We only consider synthesized data with fixed labels as in the current practice of LLMs and diffusion models.

A.3. Data Selection

Comprehensive surveys on data selection for language models can be found in (Albalak et al., 2024), along with theoretical studies on selection in high-dimensional settings (Sorscher et al., 2022; Kolossov et al., 2024). Specifically, (Kolossov et al.,

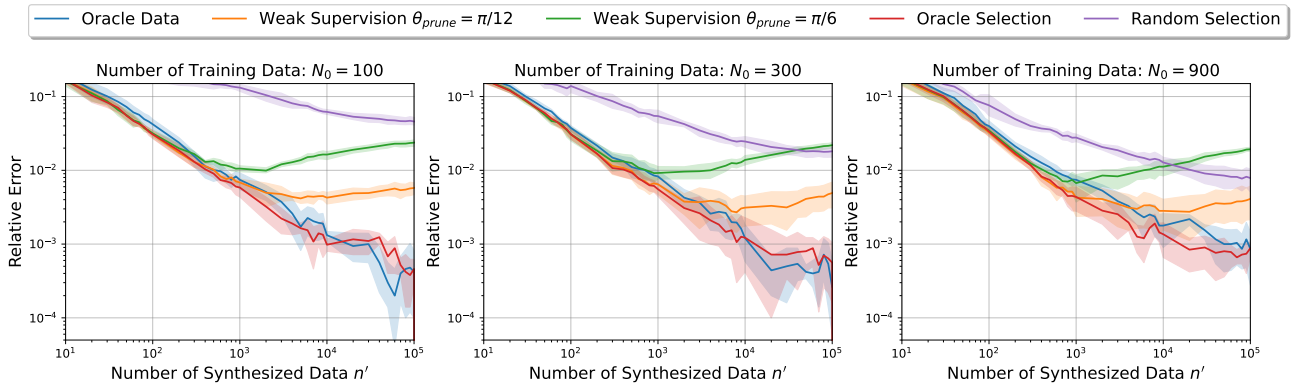

Figure 3. Simulations on the scaling with respect to the number of selected data, n'. $\tau = 0.15$, $N_1 = 10^6$. The Bayes optimal classifier achieves approximately 94% accuracy on this distribution. The y-axis denotes the relative error, i.e., the accuracy relative to the optimal accuracy.

2024) also explore the use of surrogate models for producing labels during selection, followed by curation of the original labels. In our study, selection is applied to synthesized data where original labels are not available, resulting in distinct phenomena compared to these approaches on original data.

B. Simulations on Synthesized Data

The theoretical results are based on the best-case scenario of having unlimited access to synthesized data and operating in the high-dimensional limit. In this framework, the generator and verifier's impact on performance is reflected in binary outcomes: 100% or 0%. Here, we present simulation results in finite regimes to demonstrate the practical implications of the theory, specifically how the generator and verifier affect performance and scaling rates with respect to the number of synthesized data points.

B.1. Setting

Following the theoretical setting, we consider the same Gaussian mixture and linear models for the generator and selector. Let w_* be a fixed unit vector in $\mathbb{R}^d$. The distribution P_{orig} is

$$x|y \sim N(y\tau w_*, I_d/d), \quad \text{for } y \in [-1, +1].$$

Here, τ is a positive scalar that controls the overlap.

Synthesized Data Generation We initially sample N_0 data from the distribution P_{orig} as the original dataset, D_{orig}, which is used to train a linear model $\hat{w}$ using ordinary least squares. Subsequently, we use $w_{gen} = \hat{w}$ to generate N_1 synthesized data points with sigmoid, constituting the dataset D_{gen}.

Reinforcement The data is selected with various w_{prune} in Equation (1) from w_* to w_θ where θ is the angle between w_θ and w_*. n' is the number of data points selected.

C. Predicting the Eigenvalues

We leverage the code base provided by (Charton, 2022) at `https://github.com/facebookresearch/LAWT` under the license CC BY-NC 4.0.

Input and Tokenization. We use the example of solving arithmetic tasks with a transformer, which offers an interpretable setting to understand the generation quality since we have a clear metric for error and an attainable ground truth. Specifically, we leverage the problems described in Charton (2022). Transformers (Vaswani et al., 2017) are trained to predict the five eigenvalues of 5×5 symmetric real matrices from inputs consisting of the 25 real entries. Model inputs are sequences of 25 real entries, rounded to three significant digits, and tokenized as triplets of signs(+ or −), mantissas (from `0` to `999`) and

power of ten exponents (from `E-100` to `E100`). For instance, the 2×2 matrix,

$$\begin{pmatrix} 2.3 & 0.6035 \\ 0.6035 & -3.141 \end{pmatrix}$$

will be encoded as the sequence of 12 tokens: `+ 23 E-1 + 604 E-3 + 604 E-3 - 314 E-2`. Model outputs are vectors of 5 real eigenvalues, rounded to three significant digits, and tokenized as before, as triplets of sign, mantissa and exponent (the `P1000` encoding from (Charton, 2022)). All training, test, and synthesized data are generated by sampling matrices with independent entries from $\mathcal{U}[-10, 10]$. A prediction is considered correct if the relative error in the L^1 norm is below a certain tolerance τ.

Model and Optimization. We train sequence-to-sequence transformers (Vaswani et al., 2017), with 4 layers in the encoder, and one in the decoder, 512 dimensions and 8 attention heads, to minimize a cross-entropy loss, using the Adam optimizer (Kingma and Ba, 2014), with a fixed learning rate of $5 \cdot 10^{-5}$, after an initial linear warm-up phase over the first 10,000 optimization steps. The synthesized data generator is trained on a limited sample of about 200,000 examples for 65 epochs before overfitting. For the results in Table 3, the model is trained with 12M samples for 400 epochs before overfitting.

Evalution. Model accuracies are measured on a held-out test set of examples not seen at training. Model predictions are evaluated by decoding the output sequence as a vector of 5 real numbers $(p_1, p_2, p_3, p_4, p_5)$, and assessing that a prediction $\mathbf{p}(p_1, p_2, p_3, p_4, p_5)$ of eigenvalues $\mathbf{v}(v_1, v_2, v_3, v_4, v_5)$ is correct if the relative error in L^1 norms is below some tolerance τ, i.e. if

$$\sum_{i=1}^{5} |v_i - p_i| < \tau \sum_{i=1}^{5} |v_i|.$$

We use tolerances τ of $5, 2, 1$ and 0.5%.

In both the theoretical results and the simulations, we examine a classification case where oracle supervision can select "100% correct" synthesized data. Beyond these settings, the synthesized data can have a continuous spectrum regarding its distance to the ground truth and the selected data may only be correct to some extent. In this sense, we assess the impact of reinforcement methods and the generator in two experiments: (1) training a transformer to predict eigenvalues of a matrix and (2) fine-tuning Llama-2-7B on a news summarization task.

C.1. Understanding the Quality of Synthesized Data

Selection is Crucial. In Table 1, we report the accuracy (on a test set) of generated predictions using greedy decoding and beam search of various sizes. On the right side, only the best beam solution—the most confident solution by the model—is evaluated. Increasing the number of beams does not lead to an increase in accuracy, indicating that self-selection on the prediction does not result in improved predictions. However, on the left side, we evaluate all top-k candidates in the beam k with respect to the ground truth, and the best one is counted towards the accuracy. When going from greedy (beam 1) to beam 50, the accuracy improves from 66.9% to 90.4% with $\tau = 2\%$. Therefore, we conclude that while the model demonstrates the potential to generate improved solutions, it lacks the inherent capability to autonomously select superior predictions. To curate better synthesized data, external supervision is crucial.

Table 1. **The Generator's Accuracies for Different Beam Sizes.** Left: all solutions in beam are evaluated and the best is calculated, selection with oracle. Right: only the beam solution with the smallest perplexity is evaluated, same as self-selection.

	Verify all beams			Verify the best beam		
Tolerance τ	2%	1%	0.5%	2%	1%	0.5%
Beam 50	90.4	60.4	22.9	65.9	19.2	2.4
Beam 35	89.2	56.9	19.8	66.0	19.2	2.4
Beam 25	88.0	53.2	16.8	66.1	19.3	2.4
Beam 10	83.7	43.1	10.5	66.2	19.5	2.5
Beam 5	79.3	34.9	7.1	66.5	19.7	2.4
Greedy	66.9	20.2	2.4	66.9	20.2	2.4

Table 2. **Performance of Models Trained on Various Synthesized Data.** The models are evaluated using greedy decoding. "Synthesized Generator" refers to the assessed performance of the generator as an indicator on generation quality.

		Tolerance τ 2%	1%	0.5%
Data Selection 2%		72.1	20.2	2.3
Label Selection	Beam 50	**84.0**	**33.4**	**4.9**
	Beam 25	79.9	28.7	4.1
	Beam 10	73.9	22.7	2.9
	Beam 5	69.1	19.0	2.3
Greedy w/o selection		60.5	14.5	1.7
Synthesized Generator		66.9	20.2	2.4

Table 3. **Performance of Models Trained with More Data and a Stronger Verifier.** Synthesized data curated with oracle pruning at 1% or 2% tolerance. Train from scratch and evaluate using greedy decoding.

Method	Data Size	Tolerance τ 2%	1%	0.5%
Data Selection 2%	1M	72.1	20.2	2.3
Data Selection 2%	12M	80.1	26.3	3.4
Data Selection 1%	12M	95.1	50.1	8.6

C.2. Transformer for Math

We now move to generating synthesized data with this problem. We randomly collect more prompts (matrices) and use the generator to label them. We introduce a **verifier** that serves as the oracle supervision, measuring the distance between model predictions and the correct solutions. The data is selected with this verifier via two methods:

- **Data Selection**: A random set of matrices is created, and the generator computes the eigenvalues using greedy decoding. Only data with the correct predictions (within a tolerance of $\tau = 2\%$, according to the verifier) are retained.
- **Label Selection**: A random set of matrices is created, and the generator predicts k possible solutions using beam search (in the top-k generation pool). The verifier selects the best prediction, which is then used for the training data. We experiment with beam sizes of 5, 10, 25, 35, and 50.

Overall, seven synthesized datasets, each containing one million examples, are created: one using Data Selection, five using Label Selection with various beam sizes, and one without any selection. In the Data Selection setting, approximately two-thirds of the data are retained with a tolerance of $\tau = 2\%$. Using these datasets, the transformer is trained from scratch and evaluated with greedy decoding. The accuracy of the trained models is reported in Table 2, showing the best run across five seeds. These results are compared with the 'Synthesized Generator' row from Table 1 to assess the performance of the generator and as an indicator of generation quality. We observe the following:

Model collapse is observed. Comparing 'Greedy without selection' and 'Synthesized Generator', training with its own synthesized data leads to a degradation in performance, even with five times more synthesized data than the amount used to train the generator. Model collapse happens.

Supervision goes beyond model collapse. When we leverage reinforcement, both data selection and label selection show considerable improvement compared to using the generated data without selection. According to theory, the effective ψ/ϕ is much lower. All the selection results surpass the synthesized generator, indicating that we can improve upon the original data with oracle reinforcement and mitigate model collapse. Additionally, increasing the number of beams consistently enhances performance, as the quality of the selected synthesized data continues to improve.

How Far Can We Go with Synthesized Data and Verifier? We examine data selection with 30 times more data and with a stricter verifier tolerance of 1% to investigate the best performance achievable. As shown in Table 3, using more data and a stronger selection improves performance. A stronger selection corresponds to better oracle-based verifier, aligning with the theory and simulation.

C.3. Finetuning Models with Synthesized Data

In all previous experiments, the data generated from the generator (using beam or reject sampling) were used to train a new model. In this section, we consider using data generated from the generator to finetune models pre-trained on a small sample of ground truth data. We consider four cases:

- Fine-tuning the generator (Model A).
- Fine-tuning a model pre-trained to the same accuracy as the generator (62%, Model B).
- Fine-tuning a model pre-trained to higher accuracy (93%, Model C).
- Fine-tuning a model pre-trained to low accuracy (4%, Model D).

Table 4 compares accuracy of the four fine-tuning cases to that of a model trained from scratch. Fine-tuning only achiueves better performance when the pre-teained model achieved higher accuracy than model A. In all other cases, fine-tuning brings

Table 4. **Performance of models fine-tuned on 1M examples generated by the generator.** $\tau = 2\%$

	Model A (66%)	Model B (62%)	Model C (93%)	Model D (4%)	From scratch
Rejection	61.8	72.9	82.1	66.3	72.1
Beam 50	74.1	82.6	87.3	78.3	84.0
Beam 35	72.7	81.3	86.8	76.8	80.4
Beam 25	71.3	79.8	84.4	73.3	79.9
Beam 10	67.5	75.1	83.5	68.0	73.9
Beam 5	64.9	70.8	80.1	65.6	69.1
Beam 1	61.6	62.1	75.6	55.8	60.5

no improvement. Note that fine-tuning model A on its own generated data achieves the worst result, a clear case of model collapse.

C.4. Computational Resources

We leverage a V100 GPU with 32GB of memory for all experiments involving linear algebra. The training time ranges from 1 to 5 days, depending on the data size and the number of epochs. For results in Table 3, it takes 5 days to train on 12 million data points for 400 epochs.

D. News Summarization

We leverage the XLSUM dataset (Hasan et al., 2021) at `https://huggingface.co/datasets/csebuetnlp/xlsum` under the license CC-BY-NC-SA 4.0.

Data preprocessing. For each data in both training and test dataset, it consists of a news report and a summarization, denoted as (*news*, *summarization*). We write each data in the following form:

> Article: *news*. A summary of the article: *summarization*.

Implementation details. We leverage the official implementation in Huggingface [1] for training, under the license Apache 2.0. Specifically, for training the generator, we start our training with the pre-trained llama-2, and set the learning rate to 5e-5, the learning rate scheduler as 'cosine', the number of epochs to 1, the total batch size to 32, the block size to 1024 and the others to the default value. For generating the synthesized data, we use greedy strategy to generate a summarization for each news in the training set. For training based on the selected synthesized data, we also start our training with the pre-trained llama-2, and set the learning rate to 2e-5, the learning rate scheduler as 'constant' and the others to the same. For evaluation, we first use greedy strategy to generate a summarization for each news in the test set, and then calculate the Rouge-1 score between the generated summarization and the corresponding ground truth, and finally report the average of the Rouge-1 scores of all test data. When calculating the perplexity, we only calculate the perplexity for the generated summary.

Computational Resources. All experiments were conducted using a dedicated computational cluster equipped with 4 NVIDIA A800 GPUs, each with 80 GB of memory. Our training and inference processes are performed on the cluster.

Estimated Time. Training the whole dataset for an epoch takes about 6 hours. Generating the whole dataset takes about 1 day. During evaluation, we need to first generate and calculate the rouge score, which takes around 40 minutes for one checkpoint.

We include the full figures with three levels of selection in Figure 4, 12.5%, 25%, and 50%.

[1] `https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_clm.py`

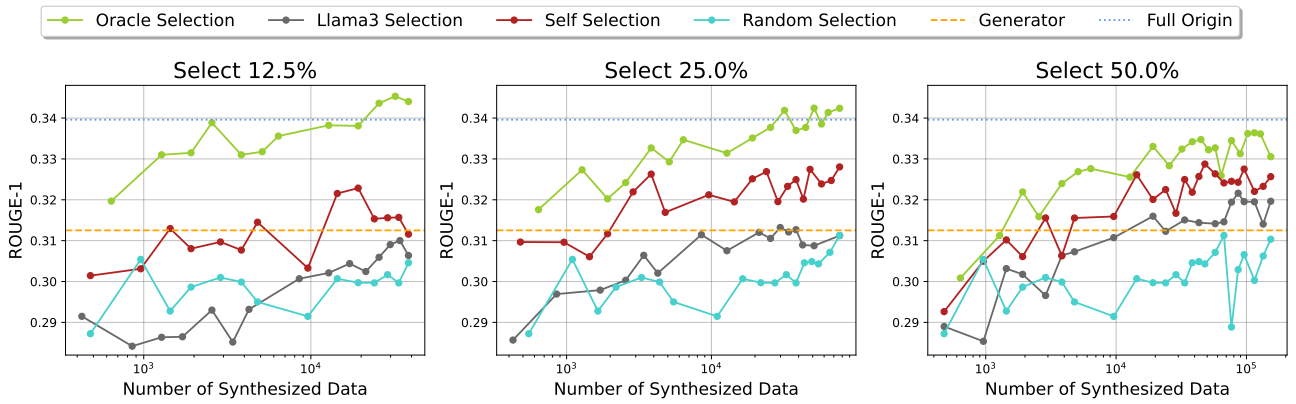

Figure 4. Results of news summarization experiments: the left figure represents models trained on 12.5% of the data, the middle on 25% of the data, and the right on 50% of the data. Each figure includes four curves illustrating different training scenarios: (1) selection with oracle reinforcement, (2) selection with Llama-3 as a weak reinforcement, (3) self-selection by the generator, and (4) random selection. Additionally, two horizontal lines are included for comparison: one representing the Rouge score of the generator model and the other representing the Rouge score of a model trained with 100% data with original labels, serving as the optimal line.

E. Theoretical Insights with Gaussian Mixtures

In Section E we have presented a special case of our general theory, which we describe here in more generality and detail. First, we outline the theory in a simplified setting.

E.1. Setting

Data Distribution. We will consider distributions P over $\mathbb{R}^d \times \{0,1\}$ with certain high dimensional concentration properties of a general form (Condition F.1). A special case are binary *Gaussian Mixtures*: features have conditional distribution given by $x \mid y \sim N(\mu_y, \Sigma)$, where $\mu_y = (2y-1)\mu$, for some $\mu \in \mathbb{R}^d$ and Σ is a positive-definite matrix with $\mathbb{E}\,\|x\|^2 = \|\mu\|_2^2 + \operatorname{tr}\Sigma = 1$. For further ease of exposition we will only consider balanced distributions

$$\mathbb{P}(y=1) = \mathbb{P}(y=0) = 1/2, \text{ for } (x,y) \sim P.$$

Synthesized Data. Let $D_N = \{(x_1,y_1),\ldots,(x_N,y_N)\}$ be a dataset of N iid pairs from the true distribution P and let $D'_N = \{(x_1,y'_1),\ldots,(x_N,y'_N)\}$ be the synthesized data generated from the same distribution, but where label y'_i (instead of y_i) has been generated by an AI model.

Downstream Model and Pruning. We will model our data selection (whether with or without feedback) via a *pruning strategy* $q = (q_1,\ldots,q_N)$ where q_i is a bit which indicates whether the ith training example from D'_N has survived pruning. For the downstream models we consider the family:

$$\mathbb{P}(y=1 \mid x,w) = \hat{y} := \sigma(x^\top w) \in (0,1),\ \sigma(z) := \frac{1}{1+e^{-z}}$$

parametrized by a vector of weights $w \in \mathbb{R}^d$ and sigmoid non-linearity σ. Let $\widehat{w}_N$ be obtained via logistic regression fitted on D'_N with ridge regularization parameter $\lambda > 0$. Thus, $\hat{w}$ is the unique minimizer of the following objective function

$$L(w) := \frac{1}{N}\sum_{i=1}^{N} q_i \ell(\sigma(x_i^\top w), y'_i) + \frac{\lambda}{2}\|w\|^2, \tag{2}$$

where ℓ is the binary cross-entropy. The corresponding downstream classifier is $\widehat{f}_N = f_{\widehat{w}_N}$, where the notation f_w refers to the linear classifier induced by a weights vector $w \in \mathbb{R}^d$, i.e $f_w(x) = (\operatorname{sign}(x^\top w)+1)/2$.

Test Accuracy. The test accuracy of the downstream model $\widehat{f}_N$ is defined by

$$acc(\widehat{f}_N) := \mathbb{P}(\widehat{f}_N(x) = f_{Bayes}(x)), \text{ for a random test point } (x,y) \sim P,$$

where $f_{Bayes}(z) := \mathbb{E}[y|x = z]$ is the Bayes-optimal classifier. In particular, note that $acc(f_{Bayes}) = 100\%$ by construction. The quantity $acc(\widehat{f}_N)$ will be the main object of our analysis, and we will be interested in how it depends on errors in the generator P and the choice of pruning strategy q, in the infinite-sample limit $N \to \infty$.

E.2. Pruning Strategy

We consider a wide class of parametrized pruning strategies q, which we term *RLHF-Pruning* (see Appendix F). They satisfy the following reasonable property:

Assumption E.1 (Independent Selection)**.** *The bits $q_1, \ldots, q_N \in \{0, 1\}$ are independent. Thus, in particular, whether any training example $e_i := (x_i, y_i') \in D_N'$ survives pruning or not is independent of what happens to the other examples $e_{j\neq i}$.*

We shall denote by $p \in [0, 1)$, the probability that the label y_i' of a synthesized example (x_i, y_i') is different from the true label y_i, i.e

$$p := \mathbb{P}(y_i' \neq y_i). \tag{3}$$

Note that p does not dependent on the example index i, due to the iid assumption.

Our RLHF-pruning family (refer to Appendix F for details) is described by four parameters $(\phi_0, \phi_1, \psi_{01}, \psi_{10})$, defined as follows

$$\phi_k = \mathbb{P}(q_i = 1 \mid y_i' = k, y_i = k), \quad \psi_{k\ell} = \mathbb{P}(q_i = 1 \mid y_i' = \ell, y_i = k). \tag{4}$$

For simplicity of exposition, we will focus on *symmetric* pruning strategies, $\phi_1 = \phi_0 = \phi$ and $\psi_{01} = \psi_{10} = \psi$. Assumption E.1 implies that for any class labels $k, \ell \in \{0, 1\}$, the random variables $(z_{ik\ell})_{i\in[N]}$ defined by $z_{ik\ell} = 1[y_i = k, y_i' = \ell, q_i = 1]$ are iid with Bernoulli($p_{k\ell}$) distribution, with

$$p_{kk} = (1 - p)\phi_k/2, \text{ and } p_{k\ell} = p\psi_k/2 \text{ if } k \neq \ell. \tag{5}$$

In this section we focus on a special case of *supervised* pruning strategies q of the form

$$\textbf{Supervised Pruning:} \quad q_i = 1[y_i(x_i^\top w_{prune}) > 0], \tag{6}$$

for some weights $w_{prune} \in \mathbb{R}^d$. This pruning strategy filters out all examples on which there is disagreement on the assigned label. In Appendix F we show how we can map this to (ϕ, ψ)-pruning.

Let us provide two more notable examples of (symmetric) pruning strategies.

No Pruning. The case $(\phi, \psi) = (1, 1)$ corresponds to no pruning, i.e using the entire training dataset.

Oracle Pruning. The case $(\phi, \psi) = (1, 0)$. The pruning strategy only keeps indices corresponding to examples in the dataset which have correct label (all corrupted labels are discarded).

E.3. Performance of Models Trained with Pruning: Insights from Infinite-Sample Regime

The following is our main theoretical result (see Theorem F.3 for full statement). It characterizes test accuracy $acc(\widehat{f}_N)$ of the downstream model on pruned data as a function of p (the label disagreement) and the parameters (ϕ, ψ) of the pruner, in the theoretical limit of infinite training data ($N \to \infty$).

Theorem E.2 (Simplified version of Theorem F.3)**.** *Let Assumption E.1 be in order. Fix p, ϕ, ψ and define the breakdown point $p_\star \in (0, 1)$ by $p_\star := 1/(1 + \psi/\phi)$. For the family of data distributions obeying Condition F.1 (including the Gaussian mixture), for a downstream model $\widehat{f}_N$ trained on data from a generator with error rate p, pruned with an RLHF-type strategy with parameters (ϕ, ψ), in the limit $N \to \infty$ it holds a.s that:*

(i) If $p < p_\star$ then $acc(\widehat{f}_N) = 100\%$.

(ii) If $p > p_\star$ then $acc(\widehat{f}_N) = 0\%$. The pruner is overwhelmed by so many inaccuracies in the synthesized data, and the downstream model learns the exact opposite of the true class labels.

Thus, there is a sharp phase-transition around the corruption level $p_\star := 1/(1+\psi/\phi)$: as p is increased past level $p_\star$, the downstream model $\widehat{f}_N$ abruptly switches from being perfectly accurate, to perfectly inaccurate! The proof (see Appendix F.7 for a sketch) explicitly computes empirical test accuracy in terms of $N_{k\ell} := \sum_{i=1}^{N} z_{ik\ell}$, which follow a binomial distribution, bounding the gap to the population accuracy, and using concentration of measure type techniques. Note that the sharp transition is due to the infinite-sample regime, where we can avoid finite-sample corrections.

See Figure 5 (and Figure 6 in Appendix F) for an empirical illustration of the theorem.

Remark E.3. *Note that the* 100% *accuracy achievable in Theorem E.2 is idealized, and is expected to only hold in infinite sample regime (with a possibly large but fixed input dimension).*

E.4. Some Consequences of Theorem F.3

We now present some illustrious applications of Theorem F.3. These examples are empirically confirmed in Figure 5 (see also Figure 6 in Appendix F).

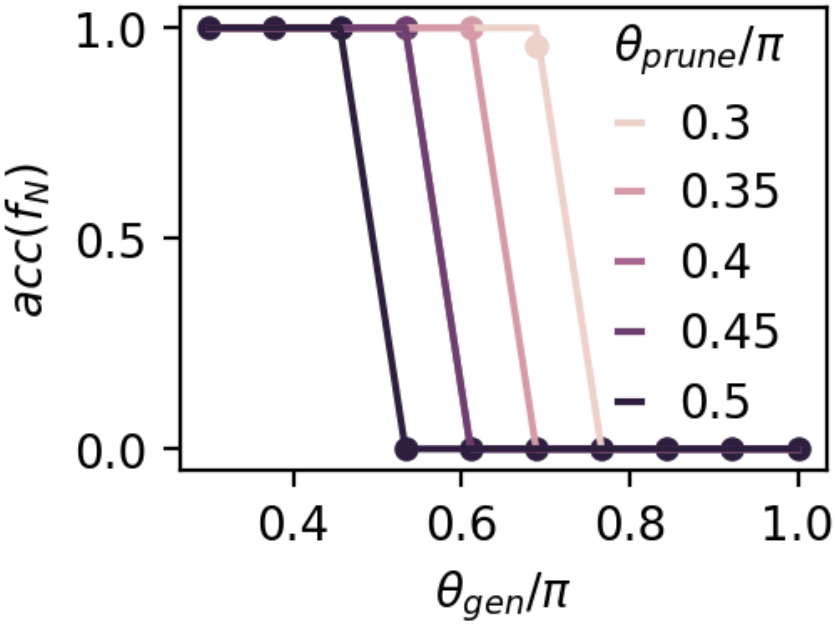

Figure 5. **Empirical Confirmation of Theorem E.2.** Comparing the breakdown points of different generators and pruners of different strengths. Synthesized data is generated from a linear model w_{gen} with classification error rate $p = \theta_{gen}/\pi \in [0,1]$. Refer to Equations (7), and recall that the triplet of angles $(\theta_{gen}, \theta_{prune}, \theta)$ maps to parameters (ϕ, ψ) for an RLHF-type pruner. The data is pruned with another linear model w_{prune} which has classification error θ_{prune}/π. Broken lines correspond to the prediction of Theorem E.2, while solid points correspond to experiments. Notice the sharp phase transitions where the model suddenly switches from perfect accuracy to worse-than-chance, a phenomenon predicated by the theorem.

No Pruning. Here, we have $\psi/\phi = 1$ and so the downstream model achieves 100% accuracy for all values of corruption parameter p up to the breakdown point $p_\star = 1/2$ predicted by Theorem F.3 .

Oracle Pruning. For this scenario, $\psi/\phi = 0$ and so Theorem F.3 predicts that the downstream model $\widehat{f}_N$ achieves 100% accuracy for all values of corruption parameter p up to the breakdown point $p_\star = 1$. This is perhaps not so surprising in hindsight. The point is that even for moderately large values of ψ/ϕ, the breakdown point $p_\star$ can still be quite close to 1.

Supervised Pruning. Consider isotropic Gaussian mixture data with means $\pm\mu$, and a pruning strategy as in Eq. (6). The parameters (ϕ, ψ) only depend on the angles $\theta_{gen}, \theta_{prune}, \theta \in [0, \pi]$ given by

$$\begin{aligned} \theta_{gen} &:= \angle(w_{gen}, \mu),\ \theta_{prune} := \angle(w_{prune}, \mu), \\ \theta &:= \angle(w_{prune}, w_{gen}). \end{aligned} \tag{7}$$

This is because, the $p_{k\ell}$'s defined in (5) now correspond to *orthant probabilities* for a trivariate normal distribution, with correlation coefficients are given by these angles (see also Figure 5).

Decoupling the Generator and Verifier. Although the generator and verifier are coupled together in supervised pruning, there are some intuitions that help us decouple them: (1) a better generator always improves performance, (2) when the verifier is poor, such as in cases of no pruning or random pruning, we have a low breakdown point and require a good generator to achieve good performance, and (3) when the verifier is sufficiently good, close to an oracle, the breakdown point is high, and any non-degenerate generator is sufficient.

F. A General Theory of Pruning with Reinforcement

We now provide the most general setting in which our theory holds. While some of our exposition here overlaps with Section E, we prefer to leave it as a complete text that provides a stand-alone overview.

F.1. Data Distribution

Consider a probability distribution P over $\mathbb{R}^d \times \{0,1\}$ with the following high-dimensional property

Condition F.1. *Given $N \leq N(d)$ iid samples $(x_1, y_1), \ldots, (x_N, y_N)$ from P with $N \leq N(d)$, the following hold estimates w.p $1 - o(1)$ uniformly on all $i, j \in [N]$, in the limit $d \to \infty$*

$$\|x_i\|^2 \simeq 1,$$
$$x_i^\top x_j \simeq \begin{cases} a, & \text{if } y_i = y_j, \\ b, & \text{if } y_i \neq y_j \end{cases}$$

where $b < a < 1$ are constants. For simplicity of presentation of our results, We will further assume that $b = -a$ or $b = 0$.

The above structural condition is inspired by an assumption in (Das and Sanghavi, 2023).

For simplicity of exposition, we will only consider balanced distributions, meaning that

$$\mathbb{P}(y = 1) = \mathbb{P}(y = 0) = 1/2, \text{ for } (x, y) \sim P.$$

Gaussian Mixture Example. As a first example, in the case of Gaussian mixtures where the features have conditional distribution given by

$$x \mid y \sim N(\mu_y, \Sigma), \tag{8}$$
$$\tag{9}$$

where $\mu_y = (2y - 1)\mu$, for some $\mu \in \mathbb{R}^d$ and positive-definite matrix Σ with $\mathbb{E}\,\|x\|^2 = \|\mu\|^2 + \operatorname{tr}\Sigma = 1$, we may take

$$a = \|\mu\|^2, \quad b = -a. \tag{10}$$

Condition F.1 then holds thanks to concentration, with $N(d) = e^{\Theta(d)}$.

F.2. Training Data, Data Pruning, and Downstream Model

Let $D_N = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a dataset of N iid pairs from the true distribution P and let $D'_N = \{(x_1, y'_1), \ldots, (x_N, y'_N)\}$ a version of the dataset (also iid) with labels y'_i instead of y_i. For example, this could be labels generated by an AI trying to reproduce real-world data. D'_N is the data on which the downstream model is trained.

We will consider a family of models given by

$$\mathbb{P}(y = 1 \mid x, w) = \hat{y} := \sigma(x^\top w) \in (0, 1),$$

parametrized by a vector of weights $w \in \mathbb{R}^d$. Here, σ is the sigmoid function defined by

$$\sigma(z) := \frac{1}{1 + e^{-z}}. \tag{11}$$

For the loss function, we use binary cross-entropy (BCE), defined by

$$\ell(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}). \tag{12}$$

Let $\widehat{w}_N$ be obtained via logistic regression fitted on D'_N with ridge regularization parameter $\lambda > 0$. Thus, $\hat{w}$ is the unique[2] minimizer of the following objective function

$$L(w) := \frac{1}{N} \sum_{i=1}^{N} q_i \ell(\sigma(x_i^\top w), y'_i) + \frac{\lambda}{2} \|w\|^2.$$

Here q_i is a bit which indicates whether the ith training example has survived pruning. The numbers $q = (q_1, \ldots, q_N)$ is called a *pruning strategy*. The corresponding downstream classifier is $\widehat{f}_N = f_{\widehat{w}_N}$, where the notation f_w refers to the linear classifier induced by a weights vector $w \in \mathbb{R}^d$, i.e

$$f_w(x) := \begin{cases} 1, & \text{if } x^\top w > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The test accuracy of the downstream model $\widehat{f}_N$ is defined by

$$acc(\widehat{f}_N) := \mathbb{P}(\widehat{f}_N(x) = f_{Bayes}(x)), \text{ for a random test point } (x, y) \sim P,$$

where $f_{Bayes}(z) := \mathbb{E}[y|x = z]$ is the Bayes-optimal classifier. In particular, note that $acc(f_{Bayes}) = 100\%$ by construction.

This quantity will be the main object of our analysis, and we will be interested in how it depends on the corruption level p and the choice of pruning strategy q, in the infinite-sample limit $N \to \infty$.

For later reference, we also define an empirical version, namely the accuracy of $\widehat{f}_N$ evaluated on the clean dataset D_N, namely

$$\widehat{acc}(\widehat{f}_N) := \frac{1}{|M|} |\{i \in M \mid \widehat{f}_N(x_i) = y_i\}|, \tag{14}$$

where $M := \{i \in [N] \mid q_i = 1\}$ collects the indices of training samples which survive pruning by q.

F.3. A Class of Parametrized Pruning Strategies

Given hyper-parameters $\phi_0, \phi_1, \psi_{01}, \psi_{10} \in [0, 1]$, we consider a broad class of parametrized pruning strategies with the following property. For any class labels $k, \ell \in \{0, 1\}$, the random variables $(z_{ik\ell})_{i \in [N]}$ defined by $z_{ik\ell} = 1[y_i = k, y'_i = \ell, q_i = 1]$ are iid with Bernoulli distribution $Bern(p_{k\ell})$, where

$$\begin{aligned} p_{k\ell} &= \mathbb{P}(q_i = 1, y'_i = \ell, y_i = k) \\ &= \mathbb{P}(q_i = 1 \mid y'_i = \ell, y_i = k) \mathbb{P}(y'_i = \ell \mid y_i = k) \mathbb{P}(y_i = k) \\ &= \begin{cases} \phi_k (1 - p)/2, & \text{if } k = \ell, \\ \psi_{k\ell} p/2, & \text{else.} \end{cases} \end{aligned} \tag{15}$$

and the numbers p, ϕ_k and $\psi_{k\ell}$ are defined by

$$p := \mathbb{P}(y'_i \neq y_i), \quad \phi_k = \mathbb{P}(q_i = 1 \mid y'_i = k, y_i = k), \quad \psi_{k\ell} = \mathbb{P}(q_i = 1 \mid y'_i = \ell, y_i = k). \tag{16}$$

Consequently, if $N_{k\ell}$ is the number of training examples that have true label k, fake label ℓ, and survive pruning, then

$$N_{k\ell} := \sum_{i=1}^{N} z_{ik\ell} \tag{17}$$

[2]Unicity is due to strong convexity of objective function.

which is has binomial distribution $Bin(N, p_{k\ell})$. As mentioned in the main text, for simplicity of exposition we considered the following simplifying assumption

$$\phi_1 = \phi_0 = \phi, \quad \psi_{01} = \psi_{10} = \psi. \tag{18}$$

Such a pruning strategy will be referred to as an RLHF-type strategy with parameter (ϕ, ψ). As usual, RLHF stands for Reinforcement Learning with Human Feedback. It can be likened to the sense in which the term is classically used, where one assumes access to a strong oracle (e.g a human) who can tell apart bugguous predictions, but can also make mistakes.

Remark F.2. *Note that the parametrization (ϕ, ψ) and (p_{00}, p_{11}) describe the same RLHF-type policy via the following bijective transformation.*

$$p_{00} = p_{11} = (1-p)\phi/2, \quad p_{01} = p_{10} = p\psi/2. \tag{19}$$

F.4. Examples

Let us present some notable examples of pruning RLHF-type pruning strategies.

No Pruning. The case $(\phi, \psi) = (1, 1)$ corresponds to no pruning, i.e the entire training dataset is used.

Pure RLHF. The case $(\phi, \psi) = (1, 0)$. The pruning strategy only keeps indices corresponding to examples in the dataset which have correct label (all corrupted labels discarded).

Supervised ((Margin-Based) Pruning. Let $w_{prune} \in \mathbb{R}^d$, and consider the pruning strategy defined by

$$q_i = 1[y_i(x_i^\top w_{prune}) > 0].$$

This pruning strategy simplify filters out all examples on which it disagrees on the assigned label.

F.5. Performance Bounds for Models Trained with RLHF Pruning

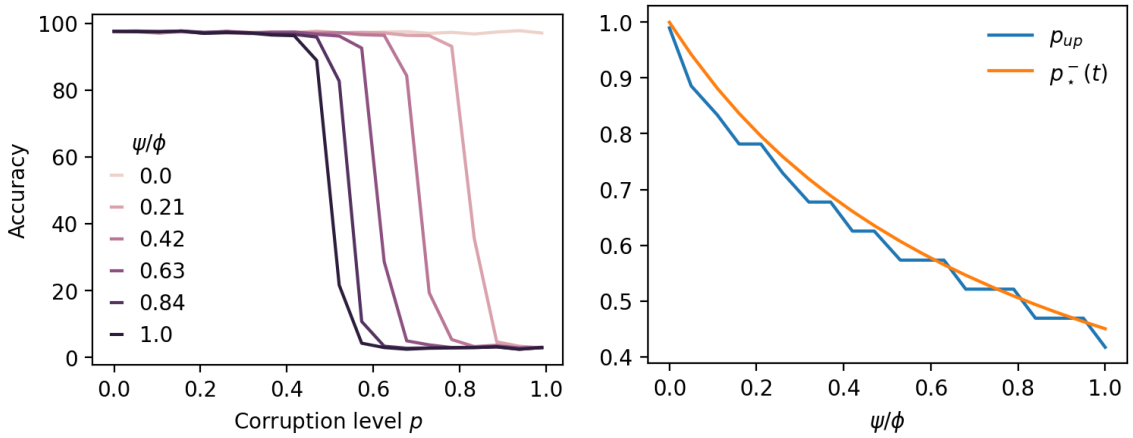

Figure 6. **Empirical Confirmation of Theorem F.3.** Comparing the breakdown points of different models. Here, the task is classifying a Gaussian mixture, with infinite training samples from datasets generated from a model with classification error rate p (x-axis). Notice the sharp phrase-transitions where the model suddenly switches from perfect accuracy to worse-than-chance, a phenomenon predicted by Theorem F.3. **Left.** Performance of RLHF-type pruning strategies with different values of the hyper-parameters (ϕ, ψ). Recall that the case $\psi/\phi = 1$ corresponds to no pruning, while $\psi/\phi = 0$ corresponds to pure RLHF. **Right.** Comparing p_{up}, approximated with $\sup\{p \mid acc(\widehat{f}_N) \geq 90\%\}$ (computed empirically), against the analytic estimate $p_\star^-(t)$ given in Theorem F.3 (for $t = 0.1$). Again, the results are in excellent agreement with the predictions of the theorem.

The following is one of our main results (proved in Appendix G).

Theorem F.3. *Suppose Condition F.1 is in order. Fix $\phi, \psi, t \in (0,1)$ and define $p_\star^\pm(t) \in (0,1)$ by*

$$p_\star^-(t) := \frac{1-t}{1-t+(1+t)\psi/\phi}, \quad p_\star^+ := \frac{1+t}{1+t+(1-t)\psi/\phi} \tag{20}$$

If $p < p_\star^-(t)$, then the limit $N \to \infty$ it holds w.p $1 - o(1)$ that the $acc(\widehat{f}_N) = 100\%$ for a downstream model $\widehat{f}_N$ trained on data from a generator with error rate p pruned with an RLHF-type strategy with parameters (ϕ, ψ).

On the other hand, if $p > p_\star^+$, then in the limit $N \to \infty$ it holds w.p $1 - o(1)$ that the $acc(\widehat{f}_N) = 0\%$ for a downstream model $\widehat{f}_N$.

Thus, there is a sharp phase-transition around the corruption level $p_\star := 1/(1+\psi/\phi)$: as p is increased past level $p_\star$, the downstream model $\widehat{f}_N$ abruptly switches from being perfectly accurate, to perfectly inaccurate!

See Figure 6 for an empirical illustration of the theorem.

The thresholds $p_\star^\pm(t)$ appearing in the above theorem are proxies for the so-called *breakdown points* $p_{up} \geq p_{down}$ defined by

$$p_{up} = \inf\left\{p \in [0,1] \,\middle|\, acc(\widehat{f}_N) \overset{a.s}{\to} 0\% \text{ in the limit } N \to \infty\right\}, \tag{21}$$

$$p_{down} = \sup\left\{p \in [0,1] \,\middle|\, acc(\widehat{f}_N) \overset{a.s}{\to} 100\% \text{ in the limit } N \to \infty\right\}. \tag{22}$$

Theorem F.3 implies $p_{down} \geq p_\star^-(t)$ and $p_{up} \leq p_\star^+(t)$ for all $t \in (0,1)$. Consequently,

Corollary F.4. *Under the hypotheses of Theorem F.3, it holds that $p_{up} = p_{down}$.*

F.6. Some Consequences of Theorem F.3

We now present some illustrious applications of Theorem F.3. These examples are empirically confirmed in Figure 6.

No Pruning. Here, we have $\psi/\phi = 1$ and so the downstream model achieves 100% accuracy for all values of corruption parameter p up to the proxy breakdown point predicted by Theorem F.3 is then $p_\star^- = 1/2 - t/2$.

Pure RLHF. For this scenario, $\psi/\phi = 0$ and so Theorem F.3 predicts that the downstream model $\widehat{f}_N$ achieves 100% accuracy for all values of corruption parameter p up to the breakdown point $p_\star^- = 1$. This is perhaps not so surprising in hindsight. The point is that even for moderately large values of ψ/ϕ, the proxy breakdown point $p_\star^-$ given in (20) can still be quite close to 1.

Self-supervised (Margin-Based) Pruning. Consider Gaussian mixture data with means $\pm\mu$, and consider a margin-based pruning strategy in Equation (6). It is clear that ϕ and ψ only depend on all the 3 angles between the set of vectors $\{w_*, w_{gen}, w_{prune}\}$, with $w_* = \mu$.

F.7. Sketch of Proof of Theorem F.3

The proof is based on the following representation (refer to Proposition G.2) of the accuracy of the downstream classifier $\widehat{f}_N$ evaluated on the the clean training dataset D_N, namely

$$\widehat{acc}(\widehat{f}_N) = \frac{N_{11}1_{\overline{A}<1/2} + N_{00}1_{\overline{D}<1/2} + N_{10}1_{\overline{B}>1/2} + N_{01}1_{\overline{C}>1/2}}{N_{11}+N_{00}+N_{10}+N_{01}}, \tag{23}$$

for some random some random variables $\overline{A}, \overline{B}, \overline{C}, \overline{D} \in (0,1)$ which depend on the $N_{k\ell}$'s defined in (17).

Remark F.5. *We only compute the accuracy $\widehat{acc}(\widehat{f}_N)$ of the downstream model $\widehat{f}_N$ evaluated on the clean training dataset D_N. By classical results in learning theory (McAllester, 2003; Shalev-Shwartz and Ben-David, 2014; Kakade et al., 2008), we know that the gap to the population version (test accuracy) $acc(\widehat{f}_N)$ shrinks to zero at rate $O(1/\sqrt{N})$, and so since the claim in Theorem F.3 is made only in the limit $N \to \infty$, we are good.*

Next, in Proposition G.3 and Proposition G.4, necessary and sufficient conditions are established to ensure $\overline{A}, \overline{D} < 1/2$ and $\overline{B}, \overline{C} > 1/2$, and therefore $\widehat{acc}(\widehat{f}_N) = 100\%$. These conditions are given explicitly in terms of the $N_{k\ell}$'s. Finally, in Proposition G.5, concentration of measure is used to control the $N_{k\ell}$'s, and present the aforementioned conditions in terms of the $p_{k\ell}$'s defined in (15), and therefore in terms of p, ϕ, and ψ alone, giving condition (20).

G. Proof of Theorem F.3

Our analysis is based on non-trivial extensions of arguments by Das and Sanghavi (2023). Viz,

- We allow for a pruning mechanism (aforementioned work does study pruning, just self-distillation), and
- We use a careful asymptotic analysis to avoid solving certain complicated fixed-point equations defining the weights vector $\widehat{w}_N$ of the downstream model $\widehat{f}_N$.

G.1. Preliminary Computations

For later use, given a pruning strategy q, define the following objects

$$I_k := \{j \in [N] \mid y_j = k\}, \tag{24}$$
$$I'_\ell := \{j \in [n] \mid y'_j = \ell\}, \tag{25}$$
$$M := \{i \in [N] \mid q_i = 1\}, \tag{26}$$
$$N_{k\ell} := \sum_{i \in I_k \cap I'_\ell} q_i = |I_k \cap I'_\ell \cap M|, \tag{27}$$
$$R := 1 - a > 0. \tag{28}$$

Thus, $N_{k\ell}$ is the number of training examples that have true label k, fake label ℓ, and survive pruning. The following result will be crucial in the sequel.

Proposition G.1. *We have the representation $\widehat{w} = \sum_{i \in M} \alpha_i x_i$, where*

$$\alpha_i = \begin{cases} A, & \text{if } i \in I_1 \cap I'_1 \cap M, \\ -B, & \text{if } i \in I_1 \cap I'_0 \cap M, \\ C, & \text{if } i \in I_0 \cap I'_1 \cap M, \\ -D, & \text{if } i \in I_0 \cap I'_0 \cap M, \end{cases} \tag{29}$$

and $A, B, C, D \geq 0$ solve the following system of equations

$$\begin{aligned} \gamma A &= \sigma(-(aN_{11}A - aN_{10}B + bN_{01}C - bN_{00}D) - RA), \\ \gamma B &= \sigma(aN_{11}A - aN_{10}B + bN_{01}C - bN_{00}D - RB), \\ \gamma C &= \sigma(-(bN_{11}A - bN_{10}B + aN_{01}C - aN_{00}D) - RC), \\ \gamma D &= \sigma(bN_{11}A - bN_{10}B + aN_{01}C - aN_{00}D - RD). \end{aligned} \tag{30}$$

Proof. The following result is inspired by (Das and Sanghavi, 2023) and the proof is similar. Observe that KKT conditions $\nabla L(w) = 0$ give $\sum_{i=1}^N q_i(\hat{y}_i - y'_i)x_i + \gamma w = 0$, i.e

$$w = \sum_{i=1}^N q_i \alpha_i x_i, \text{ with } \alpha_i := \frac{y'_i - \hat{y}_i}{\gamma}, \ \hat{y}_i := \sigma(v_i), \ v_i = x_i^\top w. \tag{31}$$

One then computes

$$\begin{aligned} v_i = x_i^\top w = \sum_{j=1}^N q_i \alpha_i x_i^\top x_j &= q_i \alpha_i + \begin{cases} a(s - q_i\alpha_i) + bt, & \text{if } i \in I_1, \\ a(t - q_i\alpha_i) + bs, & \text{if } i \in I_0, \end{cases} \\ &= \begin{cases} as + bt + Rq_i\alpha_i, & \text{if } i \in I_1, \\ bs + at + Rq_i\alpha_i, & \text{if } i \in I_0, \end{cases} \end{aligned} \tag{32}$$

where $s \geq 0$ and $t \geq 0$ are given by

$$s := \sum_{j \in I_1} q_j \alpha_j, \quad t := \sum_{i \in I_0} q_j \alpha_j. \tag{33}$$

We deduce that for any $i \in M$,

$$\gamma \alpha_i = y'_i - \sigma(v_i) = \begin{cases} 1 - \sigma(as + bt + Rq_i\alpha_i), & \text{if } i \in I_1 \cap I'_1, \\ -\sigma(as + bt + Rq_i\alpha_i), & \text{if } i \in I_1 \cap I'_0, \\ 1 - \sigma(bs + at + Rq_i\alpha_i), & \text{if } i \in I_0 \cap I'_1, \\ -\sigma(bs + at + Rq_i\alpha_i), & \text{if } i \in I_0 \cap I'_0. \end{cases} \tag{34}$$

Due to monotonicity of σ, we deduce the existence of $A, B, C, D \geq 0$ such that

$$\alpha_i = \begin{cases} A, & \text{if } i \in I_1 \cap I'_1 \cap M, \\ -B, & \text{if } i \in I_1 \cap I'_0 \cap M, \\ C, & \text{if } i \in I_0 \cap I'_1 \cap M, \\ -D, & \text{if } i \in I_0 \cap I'_0 \cap M. \end{cases} \tag{35}$$

$$\hat{y}_i = y'_i - \gamma \alpha_i = \begin{cases} 1 - \gamma A, & \text{if } i \in I_1 \cap I'_1 \cap M, \\ \gamma B, & \text{if } i \in I_1 \cap I'_0 \cap M, \\ 1 - \gamma C, & \text{if } i \in I_0 \cap I'_1 \cap M, \\ \gamma D, & \text{if } i \in I_0 \cap I'_0 \cap M. \end{cases} \tag{36}$$

Furthermore, these scalars must verify

$$\begin{aligned} &\gamma A = 1 - \sigma(as + bt + RA) = \sigma(-(as + bt) - RA), \\ &\gamma B = \sigma(as + bt - RB), \\ &\gamma C = 1 - \sigma(bs + at + RC) = \sigma(-(bs + at) - RC), \\ &\gamma D = \sigma(bs + at - RD). \end{aligned} \tag{37}$$

Finally, observe that,

$$s = N_{11}A - N_{10}B, \quad t = N_{01}C - N_{00}D, \tag{38}$$

from which we get

$$\begin{aligned} as + bt &= a(N_{11}A - N_{10}B) + b(N_{01}C - N_{00}D) \\ &= aN_{11}A - aN_{10}B + bN_{01}C - bN_{00}D, \\ bs + at &= b(N_{11}A - N_{10}B) + a(N_{01}C - N_{00}D) \\ &= bN_{11}A - bN_{10}B + aN_{01}C - aN_{00}D. \end{aligned}$$

Plugging this into (37) gives (30). □

G.2. Analytic Formula for Accuracy Evaluated Clean Training Data

One computes the accuracy $\widehat{acc}(\widehat{f}_N)$ of the downstream model evaluated on the clean training dataset D_N as

$$\widehat{acc}(\widehat{f}_N) = \frac{1}{|M|} \left(|\{i \in M \mid y_i = 1 \wedge \hat{y}_i > 1/2 \text{ OR } y_i = 0 \wedge \hat{y}_i < 1/2\}| \right).$$

We can rewrite this as follows

$$\begin{aligned}
|M| \cdot \widehat{acc}(\widehat{f}_N) &= |\{i \in M \mid y_i = 1 \wedge \hat{y}_i > 1/2 \text{ OR } y_i = 0 \wedge \hat{y}_i < 1/2\}| \\
&= |\{i \in I_1 \cap M \mid \hat{y}_i > 1/2\}| + |\{i \in I_0 \cap M \mid \hat{y}_i < 1/2\}| \\
&= \sum_{i \in I_1 \cap M} 1_{\hat{y}_i > 1/2} + \sum_{i \in I_0 \cap M} 1_{\hat{y}_i < 1/2} \\
&= |I_1 \cap I_1' \cap M| 1_{\gamma A < 1/2} + |I_1 \cap I_0' \cap M| 1_{\gamma B > 1/2} \\
&\quad + |I_0 \cap I_1' \cap M| 1_{\gamma C > 1/2} + |I_0 \cap I_0' \cap M| 1_{\gamma D < 1/2} \\
&= N_{11} 1_{\gamma A < 1/2} + N_{00} 1_{\gamma D < 1/2} + N_{10} 1_{\gamma B > 1/2} + N_{01} A_{\gamma C > 1/2}.
\end{aligned} \tag{39}$$

On the other hand, it is clear that the size of the mask is $|M| = \sum_{k,\ell} N_{k\ell}$. Putting things together gives the following result which shall be crucial in the sequel.

Proposition G.2. *For any $\phi, \psi \in [0,1]$, there is a solution (A, B, C, D) of the system of equations* (30) *such that*

$$\widehat{acc}(\widehat{f}_N) = \frac{N_{11} 1_{\overline{A} < 1/2} + N_{00} 1_{\overline{D} < 1/2} + N_{10} 1_{\overline{B} > 1/2} + N_{01} 1_{\overline{C} > 1/2}}{N_{11} + N_{00} + N_{10} + N_{01}}, \tag{40}$$

where $\overline{A} := \gamma A$, $\overline{B} = \gamma B$, $\overline{C} = \gamma C$, and $\overline{D} = \gamma D$ as usual.

Thus, to attain 100% accuracy, it suffices to have $\overline{A}, \overline{D} < 1/2$ and $\overline{B}, \overline{C} > 1/2$. The proof of Theorem F.3 will be all about establishing sufficient conditions which ensure these inequalities.

G.3. Sufficient Conditions for Perfect Accuracy

Note that since $\gamma = N\lambda$ with $\lambda > 0$ fixed and $N \to \infty$, we have $\gamma \to \infty$ and system of equations (30) simplify to[3]

$$\begin{aligned}
\overline{B} &= \sigma((aN_{11}\overline{A} - aN_{10}\overline{B} + bN_{01}\overline{C} - bN_{00}\overline{D})/\gamma), \\
\overline{A} &= \sigma(-(aN_{11}\overline{A} - aN_{10}\overline{B} + bN_{01}\overline{C} - bN_{00}\overline{D})/\gamma) = 1 - \overline{B}, \\
\overline{D} &= \sigma((bN_{11}\overline{A} - bN_{10}\overline{B} + aN_{01}\overline{C} - aN_{00}\overline{D})/\gamma), \\
\overline{C} &= \sigma(-(bN_{11}\overline{A} - bN_{10}\overline{B} + aN_{01}\overline{C} - aN_{00}\overline{D})/\gamma) = 1 - \overline{D},
\end{aligned} \tag{41}$$

where $\overline{A} := \gamma A$, $\overline{B} = \gamma B$, $\overline{C} = \gamma C$, $\overline{D} = \gamma D$ as usual, and we have used the elementary property that $\sigma(-z) = 1 - \sigma(z)$. Eliminating $\overline{A}$ and $\overline{C}$, the above equations further collapse to

$$\begin{aligned}
\overline{B} &= \sigma((aN_{11}(1-\overline{B}) - aN_{10}\overline{B} + bN_{01}(1-\overline{D}) - bN_{00}\overline{D})/\gamma), \\
&= \sigma((aN_{11} + bN_{01} - a(N_{11}+N_{10})\overline{B} - b(N_{01}+N_{00})\overline{D})/\gamma), \\
\overline{D} &= \sigma((bN_{11}(1-\overline{B}) - bN_{10}\overline{B} + aN_{01}(1-\overline{D}) - aN_{00}\overline{D})/\gamma) \\
&= \sigma((bN_{11} + aN_{01} - b(N_{10}+N_{10})\overline{B} - a(N_{01}+N_{00})\overline{D})/\gamma).
\end{aligned} \tag{42}$$

Two special cases are tractable.

The Symmetric Case: $b = -a$. We have $\overline{D} = 1 - \overline{B}$, and thus the equations become

$$\begin{aligned}
&\overline{D} = \overline{A}, \quad \overline{C} = \overline{B}, \quad \overline{D} = 1 - \overline{B}, \\
&\overline{B} = \sigma((a(N_{11}+N_{00}) - a(N_{11}+N_{10}+N_{01}+N_{00})\overline{B})/\gamma).
\end{aligned} \tag{43}$$

If $\overline{B} \leq 1/2$, then we must have $\overline{B} \geq (N_{11}+N_{00})/(N_{11}+N_{10}+N_{01}+N_{00})$, which is impossible if we impose

$$N_{10} + N_{01} < N_{11} + N_{00}, \tag{44}$$

i.e the number of bad indices which survive is smaller than the number of good indices which survive pruning. Thus, under the previous condition, we must have $\overline{C} = \overline{B} > 1/2$ and $\overline{A} = \overline{D} = 1 - \overline{B} < 1/2$. By symmetry of the preceeding argument we know that the condition is also necessary. We deduce the following result.

[3]These simplifications are made possible by the *Mean Value Theorem*.

Proposition G.3. *Suppose $b = -a$. Then, for any solution (A, B, C, D) of the system of equations* (30)*, the inequalities*

$$\overline{C} = \overline{B} > 1/2, \quad \overline{D} = \overline{A} < 1/2, \tag{45}$$

hold if and only iff $N_{10} + N_{01} < N_{11} + N_{00}$.

Skewed Case: $b = 0$. Here, we have

$$\begin{aligned} \overline{B} &= \sigma(a(N_{11} - (N_{11} + N_{10})\overline{B})/\gamma), \\ \overline{D} &= \sigma(a(N_{01} - (N_{01} + N_{00})\overline{D})/\gamma) \end{aligned} \tag{46}$$

If $\overline{B} \leq 1/2$, then $\overline{B} \geq N_{11}/(N_{11} + N_{10})$, which is impossible if we impose

$$N_{10} < N_{11}, \tag{47}$$

i.e the number of examples with true label 1, which are incorrectly labelled as 0 in the dataset, which survive pruning is less than the number of examples with true label 1, which are correctly labelled and survive pruning. We deduce that $\overline{B} > 1/2$ under the above condition.

Similarly, if $\overline{D} \geq 1/2$, then $\overline{D} \leq N_{01}/(N_{01} + N_{00})$, which is impossible if we impose

$$N_{01} < N_{00}, \tag{48}$$

i.e the number of with true label 0 but incorrectly labelled as 1 in the dataset, which survive pruning is less than the number of examples with true label 1, which are correctly labelled and survive pruning. We obtain the following result.

Proposition G.4. *Suppose $b = 0$. Then, for any solution $(\overline{A}, \overline{B}, \overline{C}, \overline{D})$ of* (30)*, we have*

$$\overline{C}, \overline{B} > 1/2 \text{ iff } N_{10} < N_{11}, \tag{49}$$

$$\overline{D}, \overline{A} < 1/2 \text{ iff } N_{01} < N_{00}. \tag{50}$$

G.4. Concentration

We shall now derive conditions which are sufficient to ensure the hypothesis in Propositions G.3 and G.4, namely $N_{k\ell} < N_{kk}$ for all $k, \ell \in \{0, 1\}$ with $k \neq \ell$. Recall that for any $k, \ell \in \{0, 1\}$, the counter $N_{k\ell}$ is random with binomial distribution $Bin(N, p_{k\ell})$. Now, by basic binomial concentration, we know that if $p, \psi \in [0, 1)$ and $\phi \in (0, 1]$, then for any fixed $t \in (0, 1)$, it holds w.p $1 - o(1)$ that

$$\begin{cases} N_{k\ell} \leq (1 + t)Np_{k\ell}, & \text{if } k \neq \ell, \\ N_{k\ell} \geq (1 - t)Np_{k\ell}, & \text{if } k = \ell. \end{cases} \tag{51}$$

In particular, w.p $1 - o(1)$, it holds that

$$N_{k\ell} \leq (1 + t)Np_{k\ell}, \tag{52}$$

$$N_{kk} \geq (1 - t)Np_{kk}. \tag{53}$$

Comparing the above inequalities, we deduce the following result.

Proposition G.5. *If the following condition holds*

$$\frac{p_{01} + p_{10}}{p_{00} + p_{11}} < \frac{1 - t}{1 + t} = 1 - \epsilon \text{ with } \epsilon := \frac{2t}{1 + t}, \tag{54}$$

then w.p $1 - o(1)$ it holds that

$$N_{10} + N_{01} < N_{11} + N_{00}. \tag{55}$$

G.5. Proof of Theorem F.3

Follows directly from putting together Propositions G.2, G.3, G.4, and G.5, and then solving the inequality

$$\frac{1 - t}{1 + t} \leq \frac{p_{01} + p_{10}}{p_{00} + p_{11}} = \frac{2p\psi}{2(1 - p)\phi} = \frac{p\psi}{(1 - p)\phi}$$

for p. □