Associative Memories with Heavy-Tailed Data

Vivien Cabannes
FAIR, Meta

Elvis Dohmatob
FAIR, Meta

Alberto Bietti
Flatiron Institute

Abstract

Learning arguably involves the discovery and memorization of abstract rules. But how associative memories appear in transformer architectures optimized with gradient descent algorithms? We derive precise scaling laws for a simple input-output associative memory model with respect to parameter size, and discuss the statistical efficiency of different estimators, including optimization-based algorithms. We provide extensive numerical experiments to validate and interpret theoretical results, including fine-grained visualizations of the stored memory associations.

1 Introduction

Transformer language models have reached a new scale and social impact since the introduction of ChatGPT. On the modeling side, some transformer layers have been found to behave as key-value memories [10, 20]. More generally outer-product associative memory matrices arise naturally from training dynamics on intermediate weights [4]. Beyond simple associative recall, the combination of multiple associative rules at different layers may lead to certain circuits with rich "reasoning" behaviors based on context [7, 4, 21]. For example, an intermediate layer input token may encode for the topic "linux", leading to an output token that will trigger a specific behavior in the transformer's following layers when processing the token "terminal". This motivates our study, which focuses on a simple model that aims to be representative of LLMs in two ways. First, we focus on heavy-tailed data distributions over discrete tokens, a natural assumption for text data [23]. Second, we consider associative memory models that store input-output pairs through outer-products of finite-dimensional embeddings, and can be seen as a proxy of the intermediate layers of transformers.

To summarize our contributions: *(i)* we provide precise statistical rates for outer-product memories with random embeddings, and compare different memory storage schemes in the context of Zipf-distributed data; *(ii)* we compare theoretical schemes to the weights learned by various optimization algorithms used in practice, and illustrate the role of different design choices with numerical experiments.

2 Settings and Statistical Study

Data. Let $p \in \Delta_{[N]\times[M]}$ be a joint distribution on inputs $x \in [N]$ and outputs $y \in [M]$, where $p(x) \propto x^{-\alpha}$ follows a Zipf law with $\alpha > 1$, and $p(y|x) = \delta_{f_*(x)}(y)$ is a deterministic function captured by $y = f_*(x)$. Let $\mathcal{D}_T = (x_t, y_t)_{t\in[T]} \sim p^{\otimes T}$ be T known independent samples generated by p. Here, x and y models intermediate transformer layer input and output tokens. We assume no noise in the input/output relationship for simplicity, and long-tails on the input tokens.

Model. We focus on input/output mapping $f : [N] \to [M]$ parameterized by a matrix $W \in \mathbb{R}^{d\times d}$,

$$f_W(x) = \arg\max_{y\in[M]} u_y^\top W e_x, \qquad e : [N] \to \mathbb{R}^d; \quad u : [M] \to \mathbb{R}^d, \tag{1}$$

where $e_x = e(x)$ and $u_y = u(y)$ are token embeddings, which were obtained as the realization of random variables $E \sim \mathbb{N}(0, I_d)$ and $U \sim \mathrm{Uniform}(\mathcal{S}^{d-1})$. In our deterministic setting, we shall

Associative Memory & Hopfield Networks in 2023. NeurIPS 2023 workshop.

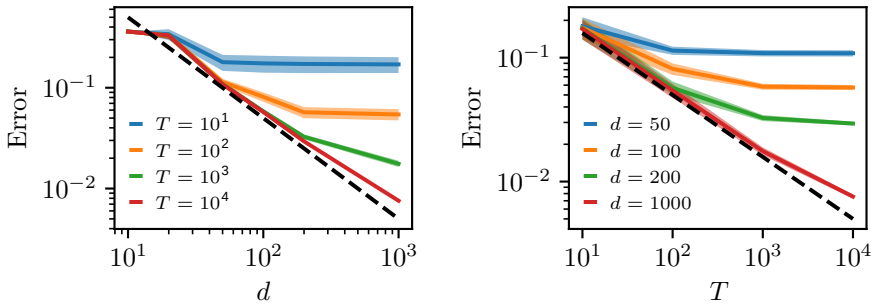

Figure 1: Scaling laws with respect embedding dimension d (left), respectively the number of data seen T (right), for various numbers of dataset size T, respectively embedding dimension d. This plots validates empirically the theory developed in the paper that proves scaling laws in $E(f_q)$ $d^{\alpha+1} + T^{1+1/\alpha}$ (dashed lines) under our setting with $= 2$, (1), (3), and the association scheme (5) with $= 0$ and $P = d$=8. The experiments averaged over 100 runs, standard deviations are shown with solid color.

consider the simpler model where only the pairs $(x, f_*(x))$ are stored in W, i.e.,

$$W_q = \sum_{x \in [N]} q(x) u_{f_*(x)} e_x^\top, \qquad q \in \Delta_{[N]}. \tag{2}$$

The quality of a mapping f is quantified through the generalization error

$$\mathcal{E}(f) = \mathbb{E}_{(X,Y)\sim p}[\mathbf{1}_{f(X)\neq Y}], \qquad f : [N] \to [M]. \tag{3}$$

Arguably, the model (1) lays out a simple model to study memorization, which could easily be extended to model more intricate memorization and training behaviors inside a transformer language model. Indeed, memories of the form (2) were found to accurately model the behavior of weight matrices in multi-layer transformers trained by gradient methods on certain tasks [4]. Hence, we expect our study to shed light on more complex mechanisms in transformers, which may involve additional aspects such as attention layers, feed-forward layers, and noisy superpositions of embeddings representing multiple tokens from an input sequence.

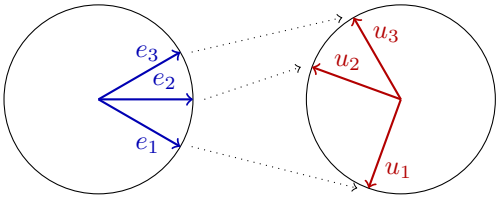

Figure 2: Error due to finite memory capacity: the stacking of associative memories in a matrix W may exhibit a pattern $W = \sum_x u_{f_*(x)} e_x^\top$ where three inputs mapped to three different outputs interact in such a way that $u_2^\top W e_1 = e_2^\top e_1 + u_2^\top u_3 e_3^\top e_1$ $1 + u_1^\top u_3 e_3^\top e_1 = u_1^\top W e_1$, so that $f_W(x = 1) = 2$ ~~$\neq$~~ $1 = f(x = 1)$. In other terms, memory interference may lead to wrong prediction, illustrating the finite capacity of the model f_W (1) to store all data associations.

Association scheme. Because the embeddings are random, they can interfere in ways that limit the memory capacity of the associative memory model (1), see Figure 2. In order to privilege the storage of frequent associations, one may weight memories according to their empirical frequencies, leading to the scheme, for $\rho \geq 0$ and

$$\hat{q}_\rho(x) = \hat{p}(x)^\rho, \qquad \hat{p}(x) = \frac{1}{T} \sum_{t \in [T]} \mathbf{1}_{X_t = x}, \qquad \hat{q}_\rho \overset{\text{(in law when } T\to\infty)}{\longrightarrow} q_\rho = p^\rho. \tag{4}$$

A better option consists in explicitly limiting the storage of our model with a simple thresholding algorithm

$$\hat{q}_{\rho,[P]}(x) = \hat{p}(x)^\rho \mathbf{1}_{x \in \mathrm{top}_P((x_t)_{t\in[T]})}, \qquad \hat{q}_{\rho,[P]} \to q_{\rho,[P]}(x) = p(x)^\rho \mathbf{1}_{x\in[P]}, \tag{5}$$

where $\mathrm{top}_P((x_t))$ denotes the set made of the P most frequent inputs in the data (x_t). The following theorems, proven in Appendix, give a crisp picture of the scaling laws to expect with both the number of data T and the model capacity d.

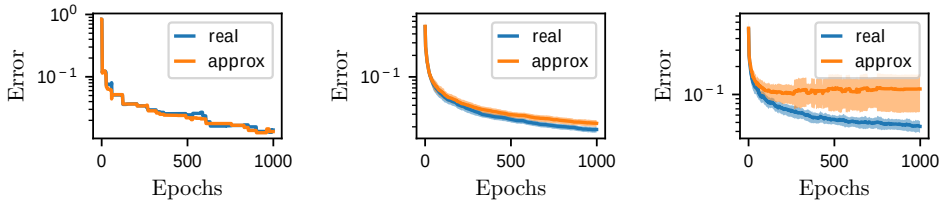

Figure 3: Comparison between the error found by optimizing W (1) with SGD on the cross-entropy loss, and its approximation with $q(x)$ (2) and the approximate update rule (11). We consider $N = 100$, $M = 5$, $f(x) = x \bmod M$, $= 2$, and batch size equals one. Left: One run with $d = N = 100$ with $= 10$. Middle: Average over 100 runs with $d = N = 100$ with $= 1$. Right: Average when $d = N = 10 = 10$ with $= 1$, which implies that our approximation is not valid anymore. The same results can be obtained for bigger batch sizes as shown in Figure 13.

Theorem 1 (Without thresholding)**.** *The performance of $f_\rho := f_{W_{q_\rho}}$ (4) is, up to poly-logarithm factors and constants that depends on both ρ, α and M,*

$$\mathbb{E}_{e,u}\mathcal{E}(f_\rho) \leq \left(\frac{d}{\varphi(N)}\right)^{-(\alpha-1)/2\rho\alpha} + T^{-1+1/\alpha}, \text{ where } \varphi(N) = \begin{cases} 1 & \text{if } 2\rho\alpha > 1 \\ \log(N) & \text{if } 2\rho\alpha = 1 \\ N^{1-2\rho\alpha} & \text{if } 2\rho\alpha < 1 \end{cases}. \quad (6)$$

In particular, when $\rho = 1$, $\mathbb{E}_{e,u}\mathcal{E}(f_0)$ scales in $d^{-(\alpha-1)/2\alpha}$. In the limit where $\rho = 0$, $\mathbb{E}_{e,u}\mathcal{E}(f_0)$ can be understood as $(d/N)^{-\infty}$ which will go to zero if and only if d is bigger than N.

Theorem 2 (With thresholding)**.** *With $N = +\infty$, the optimal performance of $f_\rho := f_{W_{q_{\rho,[P]}}}$ (4) is found for $P \simeq d^{1/(2\alpha\rho+1)}$. It is, up to poly-logarithm factors and constants,*

$$\mathbb{E}_{e,u}\mathcal{E}(f_\rho) \leq d^{-(\alpha-1)/(2\rho\alpha+1)} + T^{-1+1/\alpha}. \quad (7)$$

Theorem 3 (Minimax performance)**.** *When $N = +\infty$, tor any weighting scheme q, there exists a conditional distribution $p(y|x)$ such that the error made for the distribution p is lower bounded by, up to multiplicative constants,*

$$\mathbb{E}_{e,u}\mathcal{E}(f_q) \geq d^{-\alpha+1} + T^{-1+1/\alpha}. \quad (8)$$

Moreover, this performance is reached (up to logarithms factor) by the thresholding algorithm (5) *with $P \simeq d/\log(d)$ and $\rho = 0$.*

We make two insightful observations on the previous theorems. First, the optimal scaling (8) recovers the law of Hutter [13] with respect to T, and the one of Michaud et al. [21] with respect to d. This is intuitive, since Hutter [13] assumes memorizing exactly all previously seen data, while each memory could be seen as specifying a "quantum of knowledge" as modeled in Michaud et al. [21], with $d^{-\alpha+1}$ corresponding to the risk (3) of only storing the most frequent d tokens. We illustrate those optimal scalings on Figure 1. Second, the scheme q_ρ with $\rho = 0$ (5) follows two regimes, an overflow regime where d . $\varphi(N) = N$ and in essence the memory W_{q_0} is too full to recover any signal in it, and a infinite memory regime where d & N and all associations can be stored without much interference, allowing to reach a tiny generalization error. The phase transition between the two regimes is illustrated on Figure 8.

3 Optimization-based memorization

This section studies memory schemes privileged by optimization based algorithms, digging into the training dynamics behind memorization. In terms of relevance, we argue that our model (1) is a proxy for the inner layers of a transformer that memorize patterns before matching them against new data at inference time. As such, we want to understand how different key elements in the training of a transformer influence storage in our memory model.

Theoretical approximation of gradient updates. We consider the cross entropy loss as a surrogate objective to minimize, and study the form of gradient updates on batches of data. Formally, the matrix

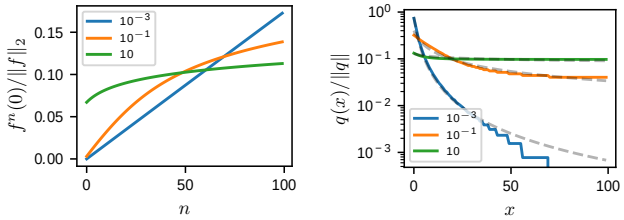

Figure 4: Theoretical approximation of the association scheme found with stochastic gradient descent with batch one and fixed learning rates. Left: Plot of $f^n(0)$ as a function of n where f is the effect of one gradient update on $q(x)$ (11). Right: Plot of the resulting $q_\gamma(x)$ when $n_x \propto p(x) / (x+3)^\alpha$ with $\alpha = 2$ and $n_N = 1$. In dashed, we represent q_ρ (4) for $\rho = 0.05$, $\rho = 0.35$ and $\rho = 1$. Those curves map well q_γ for $\gamma = 10$, $\gamma = 10^1$ and $\gamma = 10^3$ respectively.

$W \in \mathbb{R}^{d\times d}$ in (1) is optimized to minimize the loss

$$\mathcal{L}(W) = \mathbb{E}_{(X,Y)\sim p}[\ell(x,y;W)], \qquad \ell(x,y;W) = -u_y^\top W e_x + \log\Big(\sum_{z\in[M]} \exp(u_z^\top W e_x)\Big). \tag{9}$$

As detailed in Appendix B.5, when $p_W(z|x)$ does not change much for all $z \neq f_*(x)$, since u_z were sampled at random in $\mathcal{S}^d$, we expect the following approximation to hold

$$\nabla_W \ell(x, f_*(x); W) \approx -(1 - p_W(f_*(x)|x)) u_y e_x^\top. \tag{10}$$

This is notably the case for $W = 0$, random W, or if W only stores pairs $(x, f_*(x))$ with $d \gg N$. As a consequence, T steps of SGD with batch size one lead to an association scheme of the form (2) with

$$q_\gamma(x) \approx f^{Tp(x)}(0) = \underbrace{f \circ f \circ \cdots \circ f}_{Tp(x) \text{ times}}(0), \qquad \text{where} \qquad f : x \mapsto x + \frac{\gamma}{1 + M^{-1}\exp(x)}. \tag{11}$$

This equation tells us what form to expect for q for optimization schemes with different hyperparameters. This approximation is shown in Figure 4, and is validated empirically in Figure 3. Figure 5 visualizes the effect of gradient updates on W through relevant projections.

Effect of key hyperparameters. When $d > N$, the updates approximation (11) and the resulting q_γ show how a *large learning rate* γ is beneficial for our problem. Interestingly, as shown on Figure 5 (left), the same behavior holds in the presence of limited capacity, i.e., $d < N$, although interferences between embeddings break our approximation (10). Moreover, Figure 7 illustrates the usefulness of *scheduling*: using a large learning rate enables us to store associations while there is still memory space, while reducing it later in training avoids overwriting previous storage unless an association is highly frequent. Finally, processing an input x in a batch will reweight it by its frequency $p(x)$, while processing it by itself will update W similarly to setting $q_\gamma(x) = 1$ if x has not been already seen, as shown on Figure 4, predicting the better performance of *small batches*. In support of this line of reasoning, Figure 5 (middle) illustrates the benefits of splitting the descent with many steps, small batch size and large step size, even when $d < N$.

Practical considerations. In order to optimize our simple model the fastest, we have seen the usefulness of large step size and small batch size. However, for large transformers such design choices are impractical. First, large step sizes may lead to instability in realistic models [11]. Second, in order to reduce training time and improve hardware efficiency, one should process large batches [26]. Interestingly, we found that *Adam*, by renormalizing the variance of gradient updates (see Figures 15 and 16), helps to optimize our model without large learning rates. Finally, it should be noted that minimizing the cross-entropy loss implies setting $p_W(y|x) = 1$, which will lead to W diverging to infinity and unstable loss gradients. In order to ensure numerical stability, it is natural to rescale the vector $We_x \in \mathbb{R}^d$, especially since what matters for the final prediction f_W is only its direction. This is precisely what *layer-norm* is doing, which, in practice, shows even better results than Adam alone Figure 5 (right).

Learning the embeddings. Taking a step back, Lemma 3 implies that our model with d^2 parameters, the matrix $W \in \mathbb{R}^{d\times d}$ (2), only memorize about $d/\log(d)$ associations $(e_x, u_y) \in (\mathbb{R}^d)^2$ of size $2d$.

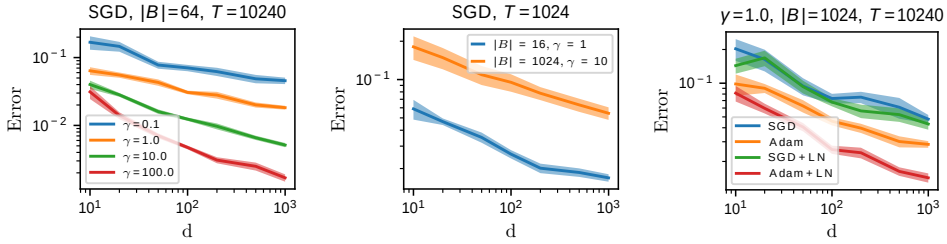

Figure 5: Effect of step size, batch size, layer-norm and Adam (with $_1 = _2 = 0$, which corresponds to SignGD). All the experiments are conducted with $N = 100$, $M = 5$, $= 2$, $f(x) = x \bmod M$, averaged over ten runs. We initialized parameters and rescale learning rates to ensure maximal feature updates, as explained in Appendix C.1. To avoid confounders, we scale on the middle plot for the variance of the gradient updates to be independent of the batch size.

Figure 6: Gradient descent dynamics from perspective of the matrix $(u_y^\top W_t e_x)_{y,x}$ 2 $\mathbb{R}^{M \ N}$ with $N = 10$, $M = 5$, $= 1{:}5$, $f(x) = x \bmod 5$, and $d = 5 < N$. A lighter color in the square $(y; x)$ means a higher value of $u_y^\top W e_x$. The optimal W corresponds to two diagonal strips of yellow boxes (see Figure 15). The matrix W_t is updated with stochastic gradient descent with batch size equal to one. From time to time, stochastic gradient descent will hit an association that is not properly stored in memory yet (the red boxes). It will consequently update the weight matrix W_t ! W_{t+1} (side by side pairs) to store it (44). Left pair: update with a big learning rate $= 10$, whose risk is to erase previous memories (the light colored boxes), similarly to q_0 (4). Right pair: update with a small learning rate $= 10^{\ 1}$, which will not store rare memory, similarly to q_ρ (4) with large .

Intriguingly, it is known that an exponential number of quasi-orthogonal elements can be put in $\mathbb{R}^d$, an event that actually holds with high probability when embeddings are random, showcasing intrinsic limitations of our "linear" model (1). Eventually, one could consider higher moments of $e_{x'}^\top e_x$ which has been the basis for modern Hopfield networks [15, 24]. However, it is unclear if this augments the number of memories stored for a given number of parameters (since it requires keeping track of each of the P vectors $e_x \in \mathbb{R}^d$). Interestingly, we note that when embeddings are learned, it is actually possible to store as many memories as desired, which can be seen from the fact that

$$W = I, \forall\, y \in [M]\, u_y \in \mathcal{S}^d, e_x = u_{f_*(x)} \qquad \Rightarrow \qquad f_*(x) = \arg\max_y u_y^\top W e_x. \tag{12}$$

Optimizing token embeddings is probably an important elements to increase memorization capacity in transformers, although enforcing $e_x = u_{f_*(x)}$ is unrealistic when embeddings are shared over different heads, and the input/output relation to learning differs among heads.

Additional experiments and visualization of the memory matrices W are provided in Appendix C.

4 Conclusion

In this work, we quantify the effect of different memorization schemes when the data follows a Zipf's law. We leverage these theoretical results to study how different optimization algorithms commonly used for transformers may lead to more efficient memorization. In particular, we showcase the efficacy of small batches and large learning rates, and, under the design constraints resulting from efficient hardware utilization and training stability, the usefulness of Adam and layer normalization.

References

[1] Shun-Ichi Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, 1972.

[2] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

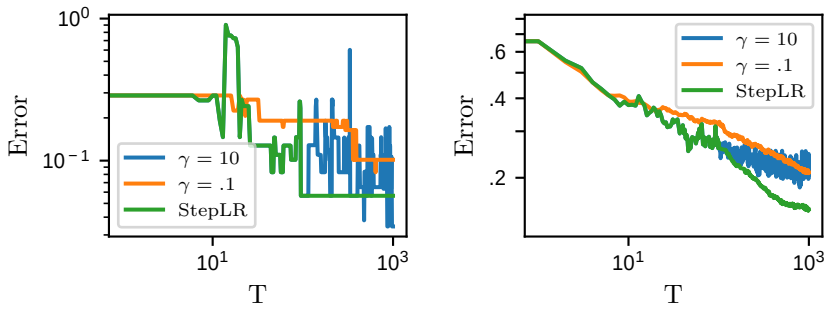

Figure 7: Learning curve of the generalization error E (3) with respect to the number of data processed by stochastic gradient descent in the setting of Figure 6. Left: comparison on a single run. A big step size allows to store more memory at the risk of overwriting past association, which explains the higher variance of the blue curve but its overall better performance. A small step size will avoid loss spikes due to memory overwriting, but will take more time to store rare associations, leading to worse performance. By decreasing the learning rates along training, e.g., with the "StepLR" scheduler [22], one can get the best of both world, i.e., store memories fast at the beginning of training when storage capacity is underused, while being more cautious at the end of training when there is no more "free" memory space. Right: Similar plot with $N = 30$ averaged over one hundred runs.

[3] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *ICML*, 2018.

[4] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. In *NeurIPS*, 2023.

[5] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 1991.

[6] Lukasz Debowski. A simplistic model of neural scaling laws: Multiperiodic santa fe processes. *arXiv preprint arXiv:2302.09049*, 2023.

[7] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. Technical report, Anthropic, 2021.

[8] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *STOC*, 2020.

[9] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *NeurIPS*, 2020.

[10] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.

[11] Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Edward Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instabilities of deep learning models. In *International Conference on Learning Representations*, 2021.

[12] John Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 1982.

[13] Marcus Hutter. Learning curve theory. *arXiv preprint arXiv:2102.04074*, 2021.

[14] Teuvo Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, 1972.

[15] Dmitry Krotov and John Hopfield. Dense associative memory for pattern recognition. In *NeurIPS*, 2016.

[16] Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *ICLR*, 2023.

[17] William Little. The existence of persistent states in the brain. *Mathematical Biosciences*, 1974.

[18] Christopher Longuet-Higgins, David. Willshaw, and Peter Buneman. Theories of associative recall. *Quarterly Reviews of Biophysics*, 1970.

[19] Alexander Maloney, Daniel Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

[20] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

[21] Eric Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *arXiv preprint arXiv:2303.13506*, 2023.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[23] Steven Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin and Review*, 2014.

[24] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *ICLR*, 2021.

[25] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *ICML*, 2021.

[26] Samuel Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *ICLR*, 2018.

[27] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artifical Intelligence*, 1990.

[28] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *NeurIPS*, 2022.

[29] Karl Steinbuch. Die Lernmatrix. *Kybernetik*, 1961.

[30] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

[31] Juan Valle-Lisboa, Andrés Pomi, and Eduardo Mizraji. Multiplicative processing in the modeling of cognitive activities in large neural networks. *Biophysical Reviews*, 2023.

[32] David Willshaw, Peter Buneman, and Christopher Longuet-Higgins. Non-holographic associative memory. *Nature*, 1969.

[33] Yuhuai Wu, Markus Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *ICLR*, 2022.

[34] Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023.

[35] Greg Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In *NeurIPS*, 2021.

[36] Jingzhao Zhang, Sairaneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? In *NeurIPS*, 2020.

Table 1: Some insightful provable scaling laws with respect to the memory capacity d, and the number of data T, for two schemes that store associations as (2) and random embeddings.

Model	Error scaling	Comment
$q(x) = p(x)$	$d^{-(\alpha-1)/2\alpha} + T^{-1+1/\alpha}$	Found with large batches in one step
$q(x) = \mathbf{1}_{x\leq d}$	$d^{-\alpha+1} + T^{-1+1/\alpha}$	Optimal scaling with random embeddings

A Additional discussion

Related work. Associative memory models have a long history in the literature on neural computation [29, 32, 18, 14, 1, 17, 12, 27, 25, 31], though the statistical insights we provide based on specific data distributions are new, to the best of our knowledge. Memorization behaviors have drawn a lot of attention recently, and are believed to be an important notion to understand the learning happening in deep neural network [e.g., 30, 8, 9, 10, 33]. Building on memorization and heavy-tailed discrete data, our model bears similarities to the ones of Hutter [13], Michaud et al. [21] or Debowski [6], although we focus on practical models with finite capacity. The discrete nature of tokens contrasts with other recent works on scaling laws that have focused on continuous Gaussian inputs [e.g., 2, 19, 28].

Why do we make errors? With a simple deterministic model, one may wonder how can we not learn perfectly the mapping f_*. There are two sources of error. One is due to not having enough data to see all the potential association $(x, f_*(x))$, and has already been studied by Hutter [13]. The other one is due to the limited memory capacity of our model, which we illustrate in Figure 2.

Model with exponential storage capacity. As a consequence of Lemma 2, the following model

$$f_1(x) = \arg\max_y u_y^\top \sum_{x'\in[P]} u_{f_*(x')}\sigma(e_{x'}^\top e_x - \eta), \tag{13}$$

where $\sigma(x) = x_+$ is the ReLU function, can fit $P = \exp(\eta^2 d/4)$ elements in memory, leading to a scaling in $\mathcal{E}(f_1) \asymp \exp(-(\alpha-1)\eta^2 d/4)$ when $p(x)$ follows a α-Zipf law. This result follows directly from two facts. When input embeddings are chosen at random, the probability that they are not η-quasi orthogonal is bounded by $P^2\exp(-d\eta^2/2)$. When input embeddings are η-quasi orthogonal, $f_1(x) = f_*(x)$ for any $x \in [P]$.

B Proofs

B.1 Useful Lemmas

Lemma 1 (Finite data, infinite memory)**.** *Consider a infinite memory model $\hat{f}$, which at time T predicts correctly all x that where seen in the past training, i.e., $x \in \{X_t\}_{t\in[T]}$, where the (X_t, Y_t) where drawn independently at random from a distribution $p \in \Delta_{[N]\times[M]}$. Under the data model the generalization error reads, with respect to the random dataset $\mathcal{D}_T = (X_t, Y_t)_{t\in[T]}$,*

$$\mathbb{E}_{\mathcal{D}_T}[\mathcal{E}(\hat{f})] \asymp T^{-1+1/\alpha}. \tag{14}$$

Here, the notation $a \asymp b$ means that there exists two constants c_1 and c_2 such that $c_1 b \leq a \leq c_2 b$.

Proof. Let us consider the infinite memory model, where an LLM can store in memory all previously seen associations (x, y). At each time t, a random positive integer x is drawn from some fixed probability distribution. At time T, the LLM would have seen $x_1, \dots, x_T$ and the associated $f_*(x_t)$, where each x_t is a random positive integer drawn independently from p. As such, the LLM would have learned a map $\hat{f}$, that only miscorrects the inputs x which are different from all the x_t for $t \in [T]$. The generalization error reads, with respect to the random dataset $\mathcal{D}_T = (X_t, Y_t)_{t\in[T]}$,

$$\mathbb{E}_{\mathcal{D}_T}[\hat{f}] = \mathbb{P}_{X,\mathcal{D}_T}(X \notin \{X_t\}_{t\in[T]}) = \sum_{x\in[N]} p(x)P_{\mathcal{D}_t}(x \notin \{X_t\}_{t\in[T]}) = \sum_{x\in[N]} p(x)(1-p(x))^T.$$

Using that $(1-a)^T = \exp(T\log(1-a))$ and $2\log(2)a \leq \log(1+a) \leq a$ for any $a \geq -1/2$, we get

$$\sum_{x\in[N]} \mathbf{1}_{p(x)\leq 1/2} \cdot p(x)\exp(-2\log(2)p(x)T) \leq \sum_{x=2}^{N} p(x)\exp(-2\log(2)p(x)T)$$
$$\leq \mathbb{E}_{\mathcal{D}_T}[\hat{f}] \leq \sum_{x\in[N]} p(x)\exp(-p(x)T).$$

Relating this series to the corresponding integral, we have

$$\begin{aligned}
&\int_{x\in[1,N]} p(x)\exp(-2\log(2)p(x)T)\,\mathrm{d}x - 1/T \\
&\quad\leq \int_{x\in[2,p^{-1}(1/T)]} p(x-1)\exp(-2\log(2)p(x-1)T)\,\mathrm{d}x \\
&\qquad + \int_{x\in[p^{-1}(1/T),N]} p(x)\exp(-2\log(2)p(x)T)\,\mathrm{d}x \\
&\quad\leq \sum_{x=2}^{N} p(x)\exp(-2\log(2)p(x)T) \leq \mathbb{E}_{\mathcal{D}_T}[\hat{f}] \leq \sum_{x\in[N]} p(x)\exp(-p(x)T) \\
&\qquad \leq \int_{x\in[1,N]} p(x)\exp(-2\log(2)p(x)T)\,\mathrm{d}x + 1/T
\end{aligned}$$

Letting N goes to infinity, we get the scaling

$$\mathbb{E}_{\mathcal{D}_T}[\hat{f}] \asymp \int_1^\infty p(x)e^{-Tp(x)}\,\mathrm{d}x \pm 1/T. \tag{15}$$

Assuming that $p(x) = Cf(x)$ for some constant C, and a smooth strongly decreasing function $f : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\lim_{x\to 0} f(x) = +\infty$, one may consider the change of variable $u = f(x)$, i.e., $x = f^{-1}(u)$. If so,

$$\mathrm{d}x = \mathrm{d}(f^{-1})'(u) = \frac{\mathrm{d}u}{f'\circ f^{-1}(u)}.$$

Hence it holds that

$$\mathbb{E}_{\mathcal{D}_T}[\hat{f}] \asymp \int_1^\infty \frac{-u}{f'\circ f^{-1}(u)} e^{-uT}\,\mathrm{d}u. \tag{16}$$

This relates to the Laplace transform of the function inside the integrand. In particular, one can work out that when $p(x) \propto C_\alpha x^{-\alpha}$, $f^{-1}(u) = u^{-1/\beta}$ from which one can deduce that

$$\int_1^\infty x^{-\alpha}\exp(-Tx^{-\alpha})\,\mathrm{d}x = \frac{\alpha}{\Gamma(\frac{\alpha-1}{\alpha})} T^{-\frac{\alpha-1}{\alpha}},$$

which recovers a result of Hutter [13]. □

Definition 1 (Quasi-orthogonality)**.** *The family $(u_z)_{z\in[P]}$ with $u_z \in \mathbb{R}^d$ is η-quasi orthogonal if*

$$\forall\,\{z,z'\} \subset [P], \qquad |\langle u_z, u_{z'}\rangle| \leq \eta, \qquad \textit{and} \qquad \|u_z\| = 1. \tag{17}$$

Lemma 2. *For any $d \in \mathbb{N}$ and $P \geq 3$, there exists an embedding $u : [P] \to \mathbb{R}^d$ such that the family $(u_z)_{z\in[P]}$ is $\eta = 2\sqrt{d^{-1}\log(P)}$-quasi orthogonal.*

Proof. The proof of Lemma 2 concerning quasi orthogonal embeddings can be done through a reasoning on random embeddings. Let (X_i) be P independent identically distributed random variables. We are interested in the event where the normalized (X_i) are η-quasi orthogonal.

$$\begin{aligned}
\mathbb{P}(\cap_{\{i,j\}\subset[P]}\{|\langle X_i, X_j\rangle| \leq \eta\|X_i\|\|X_j\|\}) &= 1 - \mathbb{P}(\cup_{\{i,j\}\subset[P]}\{|\langle X_i, X_j\rangle| \geq \eta\|X_i\|\|X_j\|\}) \\
&\geq 1 - \frac{P(P-1)}{2}\mathbb{P}(|\langle X_1, X_2\rangle| \geq \eta\|X_1\|\|X_2\|).
\end{aligned}$$

If this event can happen, it means that there exists such η-quasi orthogonal samples. As a consequence, we are looking to maximize η such that

$$\mathbb{P}(|\langle X_1, X_2\rangle| \geq \eta \|X_1\| \|X_2\|) < \frac{2}{P(P-1)}. \tag{18}$$

Let us consider (X_i) to be distributed accordingly to a rotation-invariant probability. By symmetry, we have, with f_1 denoting the first vector of the canonical basis in $\mathbb{R}^d$,

$$\mathbb{P}(|\langle X_1, X_2\rangle| \geq \eta \|X_1\| \|X_2\|) = \mathbb{P}(|\langle X, f_1\rangle| \geq \eta \|X\|) = \mathbb{P}(|\langle \frac{X}{\|X\|}, f_1\rangle| \geq \eta) \tag{19}$$

By symmetry, the vector $X/\|X\|$ is uniform on the sphere. Using that $\mathbb{P}(|\langle X, f_1\rangle| > \eta) = 2\mathbb{P}(\langle X, f_1\rangle > \eta)$ and

$$\begin{aligned}\mathbb{P}(|\langle X, f_1\rangle| \geq \eta) &= \frac{2}{\mathrm{Vol}(\mathcal{S}^{d-1})} \int_{x \in \mathcal{S}^{d-1}} \mathbf{1}_{x_1 \geq \eta} \, \mathrm{d}x = \frac{2}{\mathrm{Vol}(\mathcal{S}^{d-1})} \int_{x_1=\eta}^{2} \mathrm{Vol}(\sqrt{1-x_1^2} \cdot \mathcal{S}^{d-2}) \, \mathrm{d}x_1 \\ &= \frac{2\,\mathrm{Vol}(\mathcal{S}^{d-2})}{\mathrm{Vol}(\mathcal{S}^{d-1})} \int_{t=\eta}^{1} (1-t^2)^{\frac{d-1}{2}} \, \mathrm{d}t = \frac{2\Gamma(\frac{d}{2}+1)}{\sqrt{\pi}\Gamma(\frac{d}{2}+\frac{1}{2})} \int_{t=\eta}^{1} (1-t^2)^{\frac{d-1}{2}} \, \mathrm{d}t.\end{aligned}$$

To upper bound this probability, we proceed with

$$\begin{aligned}\mathbb{P}(|\langle X, f_1\rangle| \geq \eta) &= \frac{2\Gamma(\frac{d}{2}+\frac{1}{2})}{\sqrt{\pi}\Gamma(\frac{d}{2}+\frac{1}{2})} \int_{t=\eta}^{1} (1-t^2)^{\frac{d-1}{2}} \, \mathrm{d}t \leq \frac{2(\frac{d}{2}+1)^{1/2}}{\sqrt{\pi}} \int_{t=\eta}^{1} \frac{t}{\eta} (1-t^2)^{\frac{d-1}{2}} \, \mathrm{d}t \\ &= \frac{2(\frac{d}{2}+1)^{1/2}}{\sqrt{\pi}} \frac{1}{\eta(d+1)} (1-\eta^2)^{\frac{d+1}{2}} \leq \frac{\sqrt{2}}{\sqrt{\pi}\sqrt{\eta^2 d}} \exp(-\frac{\eta^2 d}{2}).\end{aligned}$$

The last inequality follows from the fact that

$$\frac{(d+2)}{(d+1)^2} = \frac{d+1+1}{d+1} \frac{1}{d+1} = \frac{1+\frac{1}{d+1}}{1+\frac{1}{d}} \frac{1}{d} \leq d^{-1},$$

and that for any $x \in (-1, 1)$, the concavity of the logarithm mean that $\log(1+x) \leq x$ hence that

$$(1+x)^n = \exp(n \log(1+x)) \leq \exp(nx).$$

This leads to the following series of implications

$$\begin{aligned}\exists\,(X_i)\ \eta\text{-quasi orthogonal} \quad &\Leftarrow \quad \frac{1}{\sqrt{\pi}} \big(\frac{\eta^2 d}{2}\big)^{-1/2} \exp(-\frac{\eta^2 d}{2}) \geq \frac{2}{P^2} \\ &\Leftrightarrow \quad \big(\frac{\eta^2 d}{2}\big)^{1/2} \exp(\frac{\eta^2 d}{2}) \geq \frac{P^2}{2\sqrt{\pi}} \\ &\Leftarrow \quad \frac{\eta^2 d}{2} \geq 1 \quad \text{and} \quad \exp(\frac{\eta^2 d}{2}) > \frac{P^2}{2\sqrt{\pi}} \\ &\Leftarrow \quad \frac{\eta^2 d}{2} \geq 2\log(P) - \log(2\sqrt{\pi}) \geq 1 \\ &\Leftarrow \quad \frac{\eta^2 d}{4} \geq \log(P) \geq \frac{1+\log(2\sqrt{\pi})}{2}.\end{aligned}$$

Finally, we have proven the existence of a η-quasi orthogonal family for

$$\eta \geq \sqrt{4\log(P) d^{-1}}, \qquad \text{as long as} \qquad P \geq 3. \tag{20}$$

□

Lemma 3 (Infinite data, finite memory)**.** *Let $M \geq 4$ and $d > 8\log(M)$. For any memory weight scheme $q : [N] \to \mathbb{R}$, when the embeddings e_x are independent random variables $e_x \sim \mathcal{N}(0, I)$, and the unembeddings are taken uniformly at random on the sphere,*

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \leq \inf_{\gamma} 2d^{-\gamma} + p\left(\left\{x \in [N] \,|\, dq(x)^2 \leq 16 c_\gamma \left(Q_\infty + \frac{8c_\gamma \|q\|_2^2}{d}\right)\right\}\right), \tag{21}$$

where $Q_\infty := \max_y \sum_{x; f_*(x)=y} q(x)^2$, $c_\gamma = \log(M) + \gamma\log(d)$, and $p(\mathcal{X}) = \sum_{x \in \mathcal{X}} p(x)$ denotes the probability of x to belong to $\mathcal{X} \subset [N]$. In terms of lower bound,

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \geq \frac{1}{20} p(\{x \in [N] \,|\, 3(d+1)q(x)^2 \leq Q_\infty\}). \tag{22}$$

Proof. The error made by f_W relates to the ordering between the signals $u_{f_*(x)} W e_x^\top$ and the noises $\max_{y \neq f_*(x)} u_y W^\top e_x$.

Let $f_q = f_{W_q}$. We have the following sequence of equivalence, assuming uniqueness of the argument of the maximum for simplicity,

$$\begin{aligned} f_q(x_0) \neq f_*(x_0) \qquad &\Leftrightarrow \qquad \arg\max_{y \in [M]} \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top u_y \neq f_*(x_0) \\ &\Leftrightarrow \qquad \max_{y \in [M]} \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top u_y > \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top u_{f_*(x_0)} \\ &\Leftrightarrow \qquad \max_{y \in [M]} \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top (u_y - u_{f_*(x_0)}) > 0. \end{aligned}$$

As a consequence,

$$\begin{aligned} \mathcal{E}(f_q) &= \sum_{x_0 \in [N]} p(x_0) \mathbf{1}_{f_{q(x_0)} \neq f_*(x_0)} \\ &= \sum_{x_o \in [N]} p(x_0) \mathbf{1}_{\max_y \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top (u_y - u_{f_*(x_0)}) > 0}. \end{aligned} \tag{23}$$

In other terms, we have proven the following characterization, which holds for any q, even if derived from a finite number of data,

$$\mathcal{E}(f_q) = p(\{x \in [N] \,|\, \max_y \sum_{x' \in [N]} q(x') e_{x'}^\top e_x \langle u_{f_*(x')}, u_y - u_{f_*(x)} \rangle > 0\}). \tag{23}$$

Let us introduce randomness in the model. If each $e_x \sim \mathcal{N}(0, I)$ is actually an independent random Gaussian vector in $\mathbb{R}^d$, we continue our derivation with

$$\begin{aligned} \mathbb{E}_e[\mathcal{E}(f_q)] &= \sum_{x_o \in [N]} p(x_0) \mathbb{E}_{e_{x_0}}[\mathbb{P}_{(e_x)_{x \neq x_0}}(f_q(x_0) \neq f_*(x_0) \,|\, e_{x_0})] \\ &= \sum_{x_o \in [N]} p(x_0) \mathbb{E}_{e_{x_0}}[\mathbb{P}_{(e_x)_{x \neq x_0}}(\max_y \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top (u_y - u_{f_*(x_0)}) > 0 \,|\, e_{x_0})] \\ &= \sum_{x_o \in [N]} p(x_0) \mathbb{E}_{e_{x_0}}[\mathbb{P}_{(e_x)_{x \neq x_0}}(\max_y Z_y > 0 \,|\, e_{x_0})]. \end{aligned}$$

Here, we have introduced the random variables Z_y for $y \neq f_*(x_0)$, inheriting their randomness from $(e|e_{x_0})$, and defined by

$$Z_y = \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top (u_y - u_{f_*(x_0)}). \tag{24}$$

Those are projections of Gaussian variables, hence are Gaussian. Using the fact that $\mathbb{E}[e_x] = 0$, their mean is

$$\mu_y := \mathbb{E}[Z_y] = q(x_0) \|e_{x_0}\|^2 u_{f_*(x_0)}^\top (u_y - u_{f_*(x_0)}). \tag{25}$$

Those variables are correlated. Using the characterization of the mean, we deduce that their covariance reads

$$\begin{aligned} \Sigma_{y_1, y_2} &:= \mathbb{E}[(Z_{y_1} - \mathbb{E}[Z_{y_1}])(Z_{y_2} - \mathbb{E}[Z_{y_2}])] \\ &= \sum_{x, x' \neq x_0} q(x) q(x') \mathbb{E}[e_x^\top e_{x_0} e_{x'}^\top e_{x_0}] u_{f_*(x)}^\top (u_{y_1} - u_{f_*(x_0)}) u_{f_*(x')}^\top (u_{y_2} - u_{f_*(x_0)}) \\ &= (u_{y_1} - u_{f_*(x_0)}) (\sum_{x \neq x_0} q(x)^2 e_{x_0}^\top \mathbb{E}[e_x e_x^\top] e_{x_0} u_{f_*(x)} u_{f_*(x)}^\top) (u_{y_2} - u_{f_*(x_0)}). \\ &= (u_{y_1} - u_{f_*(x_0)}) (\sum_{x \neq x_0} q(x)^2 \|e_{x_0}\|^2 u_{f_*(x)} u_{f_*(x)}^\top) (u_{y_2} - u_{f_*(x_0)}). \end{aligned}$$

Finally, we obtain the following covariance

$$\Sigma_{y,y'} = \|e_{x_0}\|^2 (u_y - u_{f_*(x_0)})^\top (\sum_{x \neq x_0} q(x)^2 u_{f_*(x)} u_{f_*(x)}^\top)(u_{y'} - u_{f_*(x_0)}). \tag{26}$$

We are left with the computation of the probability that the maximum of the n correlated, non-centered, exchangeable, Gaussian variables (Z_y) is bigger than zero.

Generic upper bound. Since we do not care about the scaling with respect to M, we proceed with

$$\max_{y \in [M]} \mathbb{P}(Z_y \leq 0) \leq \mathbb{P}(\max Z_y \leq 0) \leq \sum_{y \in [M]} \mathbb{P}(Z_y \leq 0) \leq M \max_{y \in [M]} \mathbb{P}(Z_y \leq 0), \tag{27}$$

which leads to

$$\begin{aligned}
&\mathbb{P}_{(e_x)_{x \neq x_0}}(\max_y \sum_{x \in [N]} q(x) e_x^\top e_{x_0} u_{f_*(x)}^\top (u_y - u_{f_*(x_0)}) > 0 | e(x_0)) \\
&\leq \sum_{y \neq f_*(x_0)} \exp(-\mathbf{1}_{\mu_y < 0} \frac{\mu_y^2}{2\Sigma_{y,y}}) \\
&= \sum_{y \neq f_*(x_0)} \exp(-\mathbf{1}_{\langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle < 0} \frac{\|e_{x_0}\|^2}{2} \cdot \frac{q(x_0)^2 \langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle^2}{\sum_{x \neq x_0} q(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x_0)} \rangle^2}).
\end{aligned}$$

Finally, recognizing a χ^2-variable with d degrees of freedom, for any $a > 0$,

$$\mathbb{E}[\exp(-a\|e_{x_0}\|^2)] = (1+2a)^{-d/2} = \exp(-\frac{d}{2}\log(1+2a)).$$

This leads to the final bound, with $\chi_{u,x} = \min_{y \in [M]} \mathbf{1}_{\langle u_{f_*(x)}, u_y - u_{f_*(x)} \rangle \leq 0}$.

$$\mathbb{E}_e[\mathcal{E}(f_q)] \leq \sum_{x \in [N]} p(x) \min\{1, \sum_{y \neq f_*(x)} \big(1 + \frac{q(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x)} \rangle^2}{\sum_{x' \neq x} q(x')^2 \langle u_{f_*(x')}, u_y - u_{f_*(x)} \rangle^2}\big)^{-\frac{d}{2} \cdot \chi_{u,x}}\}. \tag{28}$$

This holds for any unembedding u and associative weight scheme q. In the following, we will assume that the unembedding u are such that $\chi_{u,x} = 1$, which is notably the case when the u_y are normalized (i.e., $u_y \in \mathcal{S}^{d-1}$).

Matching lower bound. Going back to (27), one can get a matching lower bound.

$$\begin{aligned}
\mathbb{E}_e[\mathcal{E}(f_q)] &\geq \sum_{x \in [N]} p(x) \mathbb{E}_{e_x}[\max_{y \neq f_*(x)} \mathbb{P}(Z_y \leq 0 | e_x)] \\
&\geq \sum_{x \in [N]} p(x) \max_{y \neq f_*(x)} \mathbb{E}_{e_x}[\mathbb{P}(Z_y \leq 0 | e_x)] \\
&= \frac{1}{2} \sum_{x \in [N]} p(x)(1 - \max_{y \neq f_*(x)} \mathbb{E}_{e_x}[\mathrm{erf}(\frac{\mu_y}{\sqrt{2\Sigma_{y,y}}})]).
\end{aligned}$$

To conclude, we need an inequality of anti-concentration for Gaussian variables. In essence, we should distinguish two type of inputs $x \in [N]$:

- the ones where $\mu_y / \Sigma_{y,y}$ will be large enough to store the association $u_{f_*(x)} e_x^\top$, which will lead to an error decreasing exponentially fast;
- the ones where the same ratio is too small and that we should count in the lower bound.

Following this split, one can go for the simple "survival" lower bound

$$\begin{aligned}
\mathbb{E}_e[\mathcal{E}(f_q)] &\geq \sup_{t>0} \frac{1-\mathrm{erf(t)}}{2} \sum_{x_0 \in [N]} p(x_0) \max_{y \neq f_*(x_0)} \mathbb{E}_{e_{x_0}}[\mathbf{1}_{\mu_y^2 \leq 2\Sigma_{y,y} t^2}] \\
&= \sup_{t>0} \frac{1-\mathrm{erf(t)}}{2} \sum_{x_0 \in [N]} p(x_0) \max_{y \neq f_*(x_0)} \cdots \\
&\qquad \mathbb{P}_{e_{x_0}}(\|e_{x_0}\|^2 q(x_0)^2 \langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle^2 \leq 2t^2 \sum_{x \neq x_0} q(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x_0)} \rangle^2). \\
&\geq \sup_{t,s>0} \frac{1-\mathrm{erf(t)}}{2} \sum_{x_0 \in [N]} p(x_0) \mathbb{P}_{e_{x_0}}(\|e_{x_0}\|^2 \leq s) \max_{y \neq f_*(x_0)} \cdots \\
&\qquad \mathbf{1}_{sq(x_0)^2 \langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle^2 \leq 2t^2 \sum_{x \neq x_0} q(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x_0)} \rangle^2}.
\end{aligned}$$

Without optimizing for constants, taking $t = 1/\sqrt{2}$ and $s = d$, we get the simple "survival bound" that there exists a constant c such that

$$\mathbb{E}_e[\mathcal{E}(f_q)] \geq c \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x)} \rangle^2 \leq \sum_{x' \neq x} q(x')^2 \langle u_{f_*(x')}, u_y - u_{f_*(x)} \rangle^2}. \tag{29}$$

The constant can be computed explicitly as

$$c = \frac{1-\mathrm{erf}(1/\sqrt{2})}{2} \cdot \mathbb{P}(\|e_{x_0}\|^2 \leq d) > 0.158 \cdot 1/2 = 0.079,$$

where we have used that $\|e_{x_0}\|^2$ is a χ^2-variable with mean d hence smaller median, which implies that $\mathbb{P}(\|e_{x_0}\|^2 < d) > 1/2$.

Quasi-orthogonal output embeddings. Let us consider $u : [M] \to \mathbb{R}^d$ such that $(u_y)_{y \in [M]}$ is η-quasi orthogonal.

Upper bound. Going back to (28), we can work out a lower bound with

$$\begin{aligned}
&\frac{q(x_0)^2 \langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle^2}{\sum_{x \neq x_0} q(x)^2 \langle u_{f_*(x)}, u_y - u_{f_*(x_0)} \rangle^2} \\
&\geq \frac{q(x_0)^2 (1-\eta)^2}{\sum_{x \neq x_0} q(x)^2 (\mathbf{1}_{f_*(x)=y}(1+\eta)^2 + \mathbf{1}_{f_*(x)=f_*(x_0)}(1-\eta)^2 + \mathbf{1}_{f_*(x) \notin \{y, f_*(x_0)\}} 4\eta^2)} \\
&\geq \frac{q(x_0)^2 (1-\eta)^2}{4 \sum_{x \neq x_0} q(x)^2 (\mathbf{1}_{f_*(x)=y} + \mathbf{1}_{f_*(x)=f_*(x_0)} + \mathbf{1}_{f_*(x) \notin \{y, f_*(x_0)\}} \eta^2)} \\
&= \frac{1}{4} \frac{q(x_0)^2 (1-\eta)^2}{\sum_x q(x)^2 ((1-\eta^2)\mathbf{1}_{f_*(x) \in \{y, f_*(x_0)\}} + \eta^2) - q(x_0)^2} \\
&= \frac{1}{4} \frac{q(x_0)^2 (1-\eta)^2}{\eta^2 \|q\|^2 + (1-\eta^2) \sum_{x; f_*(x) \in \{y, f_*(x_0)\}} q(x)^2 - q(x_0)^2} \\
&= \frac{1}{4} \frac{q(x_0)^2 (1-\eta)^2}{\eta^2 \|q\|^2 + (1-\eta^2)(Q_y + Q_{f_*(x)}) - q(x_0)^2}.
\end{aligned}$$

Here, we have used that for the numerator

$$\langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)} \rangle^2 = (\langle u_{f_*(x_0)}, u_y \rangle - 1)^2 \geq (1-\eta)^2,$$

and the same for the term in the denominator (since their ratio cancels out), as well as

$$\langle u_y, u_y - u_{f_*(x_0)} \rangle^2 \leq (1+\eta)^2, \qquad \langle u_{f_*(x)}, u_y - u_{f_*(x_0)} \rangle^2 \leq (2\eta)^2.$$

Moreover, we have introduced

$$Q_y = \sum_{x'; f(x')=y} q(x')^2. \tag{30}$$

Using the fact that $(1+x)^d = \exp(d\log(1+x)) \leq \exp(dx)$, an upper bound directly follows from those derivations,

$$\mathbb{E}_e[\mathcal{E}(f_q)] \leq \sum_{x_0 \in [N]} p(x_0) \min\{1, M \exp\big(-\frac{d(1-\eta)^2}{2} \frac{q(x_0)^2}{4\eta^2\|q\|_2^2 + 2Q_\infty}\big)\}, \tag{31}$$

where

$$Q_\infty = \max_{y \in [M]} Q_y = \max_{y \in [M]} \sum_{x; f_*(x)=y} q(x)^2. \tag{32}$$

Matching lower bound. Similarly, one can work out a lower bound with

$$\begin{aligned}\frac{q(x_0)^2\langle u_{f_*(x_0)}, u_y - u_{f_*(x_0)}\rangle^2}{\sum_{x\neq x_0} q(x)^2\langle u_{f_*(x)}, u_y - u_{f_*(x_0)}\rangle^2} &\leq \frac{q(x_0)^2(1+\eta)^2}{\sum_{x\neq x_0} q(x)^2(\mathbf{1}_{f_*(x)=y}(1-\eta)^2 + \mathbf{1}_{f_*(x)=f_*(x_0)}(1+\eta)^2} \\ &\leq \frac{q(x_0)^2}{\frac{1-\eta}{1+\eta}Q_y + Q_{f_*(x)} - q(x_0)^2}.\end{aligned}$$

Combining this with (29), we get the lower bound, with $c = .079$,

$$\mathbb{E}_e[\mathcal{E}(f_q)] \geq c \sum_{x\in[N]} p(x)\mathbf{1}_{(d+1)q(x)^2 \leq \frac{1-\eta}{1+\eta}Q_\infty}. \tag{33}$$

Remark that in the previous lower bound, we have dropped the previous factor $\eta^2\|q\|^2$ that appears in the upper bound. We expect this term to actually be present in a tighter error characterization. In essence, we expect the embeddings to fill the full space $\mathcal{S}^{d-1}$ so that most of the difference $\langle u_{f_*(x)}, u_y - u_{f_*(x_0)}\rangle^2$ behave as η^2 most of the time. However, quantifying this precisely is beyond the scope of this paper.

Random output embeddings. In the case where the output embeddings are random, we can distinguish two cases. The cases where the embedding is η-quasi orthogonal, where one can retake the previous derivations, and the case where they are not, which will have a small probability if η is large enough.

Consider u to be random embedding taking uniformly on the unit sphere. Let us introduce the event

$$E_\eta = \{u \text{ is } \eta\text{-quasi orthogonal}\}.$$

We have seen in the proof of Lemma 2 that

$$1 - \mathbb{P}(E_\eta) \leq \frac{M^2}{2\sqrt{\pi}}\sqrt{\frac{2}{\eta^2 d}}\exp(-\frac{\eta^2 d}{2}). \tag{34}$$

For any random variable Z that is bounded by one, we have the bounds

$$\mathbb{P}(E)\mathbb{E}[Z|E] \leq \mathbb{E}[Z] = (1-\mathbb{P}(E))\mathbb{E}[Z|\neg E] + \mathbb{P}(E)\mathbb{E}[Z|E] \leq (1-\mathbb{P}(E)) + \mathbb{E}[Z|E]. \tag{35}$$

The upper bound of Lemma 3 directly follows from plugging (31) and (34) into this last equation

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \leq \frac{M^2}{2\sqrt{\pi}}\sqrt{\frac{2}{\eta^2 d}}\exp(-\frac{\eta^2 d}{2}) + \sum_{x\in[N]} p(x_0) \sum_{y\neq f_*(x_0)} \big(1 + \frac{(1-\eta)^2}{4}\frac{q(x_0)^2}{\|q\|_2^2}\big)^{-\frac{d}{2}}. \tag{36}$$

Since this is true for any η one can consider the supremum in the upper bound.

In term of lower bound, retaking (33),

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \geq \sup_{\eta\geq 0} c(1 - \frac{M^2}{2\sqrt{\pi}}\sqrt{\frac{2}{\eta^2 d}}\exp(-\frac{\eta^2 d}{2})) \sum_{x\in[N]} p(x)\mathbf{1}_{(d+1)q(x)^2 \leq 2\frac{1-\eta}{1+\eta}Q_\infty}. \tag{37}$$

In particular, when $d > 8\log(M)$ one can consider $\eta < 1/2$ such that $\eta^2 d > 4\log(M)$, which leads to $(\eta-1)/(\eta+1) > 1/3$, and, if $M \geq 4$

$$1 - \frac{M^2}{2\sqrt{\pi}}\sqrt{\frac{2}{\eta^2 d}}\exp(-\frac{\eta^2 d}{2}) \geq 1 - \frac{1}{2\sqrt{\pi}}\frac{1}{\sqrt{2\log(M)}} > 2/3.$$

All together we have proven that, as long as $M \geq 4$ and $d \geq 8\log(M)$ with $c_1 > .079 \cdot 2/3 > .052$ and $c_2 > 1/3$,

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \geq c_1 \sum_{x\in[N]} p(x)\mathbf{1}_{(d+1)q(x)^2 \leq c_2 Q_\infty}. \tag{38}$$

Writing upper bounds as survival bounds. Until now, we have written the upper bounds as the sum of exponential (31) and the lower bounds as a sum of missed associations (38), which we called "survival" bound. In order to best read how tight our characterization is, one can rewrite the upper bounds as survival bounds. In particular, as we did in the lower bound, we will dissociate x corresponding to a small exponential and the other ones. Using the fact that the $p(x)$ sum to one, we get, when the output embeddings are η-quasi orthogonal,

$$\begin{aligned}
\mathbb{E}_e[\mathcal{E}(f_q)] &\leq \sum_{x_0 \in [N]} p(x_0) \min\{1, M \exp\big(-\frac{d(1-\eta)^2}{2} \frac{q(x_0)^2}{4\eta^2 \|q\|_2^2 + 2Q_\infty}\big)\} \\
&\leq \sum_{x_0 \in [N]} p(x_0) \inf_{t>0} M \exp\big(-\frac{t(1-\eta)^2}{4}\big) + \mathbf{1}_{dq(x_0)^2 \leq t(2\eta^2\|q\|_2^2 + Q_\infty)} \\
&\leq \inf_{t>0} \exp\big(-\frac{t(1-\eta)^2}{4} + \log(M)\big) + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq t(2\eta^2\|q\|_2^2 + Q_\infty)}.
\end{aligned}$$

To simplify the bound, consider the constraints

$$\eta^2 \leq Q_\infty / \|q\|_2^2, \qquad \text{and} \qquad \eta < 1/2, \tag{39}$$

we get, using $t = 16(\log(M) + \gamma \log(d))$ for $\gamma > 0$, we get

$$\begin{aligned}
\mathbb{E}_e[\mathcal{E}(f_q)] &\leq \inf_{t>0} \exp\big(-\frac{t(1-\eta)^2}{4} + \log(M)\big) + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq t(2\eta^2\|q\|_2^2 + Q_\infty)} \\
&\leq \inf_{t>0} \exp\big(\frac{-t + 16\log(M)}{16}\big) + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 3tQ_\infty} \\
&\leq \exp(-\gamma \log(d)) + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 48(\log(M) + \gamma\log(d))Q_\infty}.
\end{aligned}$$

Finally, when the output embedding are η-quasi orthogonal with η satisfying (39), we get

$$\mathbb{E}_e[\mathcal{E}(f_q)] \leq \inf_{\gamma > 0} d^{-\gamma} + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 48(\log(M) + \gamma\log(d))Q_\infty}. \tag{40}$$

When the unembedding are chosen at random, when $d > 8\log(M)$, one can choose $\eta < 1/2$, and (36) is cast as, chosen $d\eta^2 = 4\log(M) + 2\gamma\log(d)$,

$$\begin{aligned}
\mathbb{E}_{e,u}[\mathcal{E}(f_q)] &\leq \inf_{\eta,\gamma} \frac{M^2}{2\sqrt{\pi}} \sqrt{\frac{2}{\eta^2 d}} \exp(-\frac{\eta^2 d}{2}) \\
&\qquad + d^{-\gamma} + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 16(\log(M) + \gamma\log(d))(2\eta^2\|q\|_2^2 + Q_\infty)} \\
&\leq \inf_{\gamma} \frac{d^{-\gamma}}{2\sqrt{\pi}\sqrt{2\log(M) + \gamma\log(d)}} \\
&\qquad + d^{-\gamma} + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 16(\log(M) + \gamma\log(d))(\frac{8\log(M) + 4\gamma\log(d)}{d}\|q\|_2^2 + Q_\infty)} \\
&\leq \inf_{\gamma} 2d^{-\gamma} + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \leq 16(\log(M) + \gamma\log(d))(\frac{8\log(M) + 4\gamma\log(d)}{d}\|q\|_2^2 + Q_\infty)}.
\end{aligned}$$

Finally, we have shown that when the embeddings are taken at random

$$\mathbb{E}_{e,u}[\mathcal{E}(f_q)] \leq \inf_{\gamma} 2d^{-\gamma} + \sum_{x \in [N]} p(x) \mathbf{1}_{dq(x)^2 \geq 16(\log(M) + \gamma\log(d))(\frac{8\log(M) + 4\gamma\log(d)}{d}\|q\|_2^2 + Q_\infty)}. \tag{41}$$

□

B.2 Proof of Theorem 1

Lemma 3 illustrates how the error made by a scheme q at the input x relates to the ratio between the signal $dq(x)$, provided by the associative memory $u_{f_*(x)}e_x^\top$, and the noise Q_∞, which corresponds to the signal provided by the most competitive class for $y \in [M]$. This is true up to a higher term in $\|q\|^2/d$, which corresponds to a class $y = f_*(x)$ competing against itself when the random embeddings $e_{x'}$ for x' such that $f_*(x') = y$ point in the opposite direction of e_x.

When d is large and p is regular, $c_\gamma \|q\|_2^2/d$ will be dominated by Q_∞ and the cut-off of $q(x)^2/Q_\infty$ at $32c_\gamma/d$ will behave similarly to a cut-off at $1/d$ up to logarithmic terms. Moreover, when q is chosen independently of $p(y|x)$,[1] one can expect $Q_\infty \approx p_*\|q\|^2$ where $p_* = \max_{y\in[M]} p(y)$. As a consequence, up to constants and logarithmic term, we get

$$\mathcal{E}(f_q) \overset{(\log)}{\asymp} p(\{x \in [N] \,|\, dq(x)^2 \leq p_*\|q\|^2\}). \tag{42}$$

Infinite data. When $p(x) \simeq x^{-\alpha}$, $q(x) = p(x)^\rho \simeq x^{-\rho\alpha}$, hence,

$$p(\{x \in [N] \,|\, dq(x)^2 \leq p_*\|q\|^2\}) \simeq p(\{x \in [N] \,|\, x \leq (d\|q\|^{-2})^{1/2\rho\alpha}\}) \simeq (d\|q\|^{-2})^{-(\alpha-1)/2\rho\alpha}).$$

We are left with the computation of $\varphi(N) := \|q\|^2 \simeq \int_1^N q(x)^2 \,\mathrm{d}x \simeq \int_1^N x^{-2\rho\alpha} \,\mathrm{d}x$. When $2\rho\alpha > 1$, this integral reads $1 - N^{-2\alpha\rho+1}$ which is bounded by one.

Finite data. Denoting by $\mathcal{D}_T$ the random dataset of T data, for any sequence of set $(E_x)_{x\in[N]}$,

$$\begin{aligned}
\mathbb{E}_{u,e,\mathcal{D}_T}[\mathcal{E}(f_{\hat{q}})] &= \sum p(x)\mathbb{P}_{u,e,\mathcal{D}_T}(f(x) \neq f_*(x)) \\
&= \sum p(x)\mathbb{P}_{u,e,T}(\hat{q} \notin E_x) + \sum p(x)\mathbb{P}_{u,e,T}(f(x) \neq f_*(x) \,|\, \hat{q} \in E_x).
\end{aligned}$$

The second term has been worked out before, using that $Q_\infty \leq \|q\|_2^2$

$$\begin{aligned}
\mathbb{P}_{u,e,T}(f(x) \neq f_*(x) \,|\, \hat{q} \in E_x) &\leq \inf_\gamma 2d^{-\gamma} + \mathbb{P}_T(d\hat{q}(x)^2 \leq 16c_\gamma(\hat{Q}_\infty + \frac{8c_\gamma\|\hat{q}\|_2^2}{d}) \,|\, \hat{q} \in E_x). \\
&\leq \inf_\gamma 2d^{-\gamma} + \mathbb{P}_T(d\hat{q}(x)^2 \leq c'_\gamma\|\hat{q}\|_2^2 \,|\, \hat{q} \in E_x),
\end{aligned}$$

where $c'_\gamma = 16c_\gamma(1 + \frac{8c_\gamma}{d})$.

Let us now consider the specific scheme (4), with $\rho > 0$

$$\hat{q}(x) = (\frac{1}{T}\sum_{t\in[T]} \mathbf{1}_{x=X_t})^\rho, \qquad q(x) = p(x)^\rho.$$

Using the multiplicative Chernoff bound, we get the probability bound (the randomness being due to the data),

$$\mathbb{P}_T(\hat{q}(x) < \frac{q(x)}{2^{1/\rho}}) = \mathbb{P}_T(\hat{p}(x) < \frac{p(x)}{2}) \leq \exp(-Tp(x)/8).$$

As a consequence, reusing the proof of Lemma 1, when p follows a Zipf law,

$$\begin{aligned}
\mathbb{E}[\mathcal{E}(f_{\hat{q}})] &= \sum p(x)\mathbb{P}(f(x) \neq f_*(x)) \\
&= \sum p(x)\exp(-Tp(x)/8) + \sum p(x)\mathbb{P}(f(x) \neq f_*(x) \,|\, \hat{q}(x) > q(x)/2^{1/\rho}) \\
&\asymp T^{-1+1/\alpha} + \sum p(x)\mathbb{P}(f(x) \neq f_*(x) \,|\, \hat{q}(x) > q(x)/2^{1/\rho}).
\end{aligned}$$

We are left with the computation of the second term, denote $c_\rho = 2^{-1/\rho}$, we have

$$\mathbb{E}_{u,e}\mathbb{P}_T(f(x) \neq f_*(x) \,|\, \hat{q}(x) > c_\rho q(x)) \leq \inf_\gamma 2d^{-\gamma} + \mathbb{P}_T(d\hat{q}(x)^2 \leq c'_\gamma\|\hat{q}\|_2^2 \,|\, \hat{q} \geq q(x)/2).$$

[1] To be more precise, one should actually choose $q(x)$ to be class dependent so to cram in memory as many x as possible for each different class $y = f(x)$, ensuring that $y \mapsto \sum_{x;f_*(x)=y} q(x)^2$ is constant with respect to y. For simplicity, we will not discuss this behavior that does not change the big picture beyond our exposition.

By definition of $\hat{q}$, together with Jensen's inequality when $\rho \leq 1/2$

$$\frac{1}{N} = \frac{1}{N} \sum_{x \in [N]} (q(x)^2)^{1/2\rho} \geq (\frac{1}{N} \|q\|_2^2)^{1/2\rho},$$

hence $\|q\|^2 \leq N^{1-2\rho}$. When $\rho > 1/2$, the worst value of $\|q\|$ is when all the mass is concentrated on one $q(x')$, in which case $\|q\|^2 \leq 1$. With the corresponding $\psi(N)$, we get

$$\mathbb{E}_{u,e} \mathbb{P}_T(f(x) \neq f_*(x) \,|\, \hat{q}(x) > c_\rho q(x)) \leq \inf_\gamma 2d^{-\gamma} + \mathbf{1}_{dc_\rho^2 q(x)^2 \leq c'_\gamma \varphi(N)}.$$

Finally, reusing the proof of Theorem 1, and hiding logarithmic factors,

$$\begin{aligned}
\mathbb{E}[\mathcal{E}(f_{\hat{q}})] &= \sum p(x) \mathbb{P}(f(x) \neq f_*(x)) \\
&\leq T^{-1+1/\alpha} + \inf_\gamma 2d^{-\gamma} + p(\{x \,|\, dc_\rho^2 q(x)^2 \leq c'_\gamma \psi(N)\}). \\
&\,.\; T^{-1+1/\alpha} + (\frac{d}{\psi(N)})^{-(\alpha-1)/2\rho\alpha}.
\end{aligned}$$

The case $\rho = 0$, can be easily treated by considering an error if and only if the number of seen elements $|\{x_t \,|\, t \in [T]\}|$ is smaller than d.

B.3 Proof of Theorem 2

The proof follows the structure of the proof of Theorem 1 above.

Infinite data. When $p(x) \simeq x^{-\alpha}$, $q(x) = \mathbf{1}_{x \in [P]} p(x)^\rho \simeq \mathbf{1}_{x \in [P]} x^{-\rho\alpha}$, we get

$$\begin{aligned}
p(\{x \in [N] \,|\, dq(x)^2 \leq p_* \|q\|^2\}) &= p(\{x \in [P] \,|\, dq(x)^2 \leq p_* \|q\|^2\}) + p(\{x > P\}) \\
&\simeq (\frac{d}{\varphi(P)})^{-(\alpha-1)/2\rho\alpha} + P^{-\alpha+1}.
\end{aligned}$$

The optimal threshold P is set by equalizing the two terms, which we compute as

$$\begin{aligned}
&(\frac{d}{\varphi(P)})^{-(\alpha-1)/2\rho\alpha} = P^{-\alpha+1} \\
&\quad \Leftrightarrow \quad \frac{-\alpha+1}{2\rho\alpha} \log(d) - \frac{-\alpha+1}{2\rho\alpha} \log(P) = (-\alpha+1) \log(P) \\
&\quad \Leftrightarrow \quad \log(d) - \log(P) = 2\rho\alpha \log(P) \\
&\quad \Leftrightarrow \quad P = d^{1/(2\rho\alpha+1)}.
\end{aligned}$$

This choice of P leads to a scaling in, with $f_{\rho,[P]} = f_{q_{\rho,[P]}}$,

$$\mathbb{E}_{e,u}[\mathcal{E}(f_{\rho,[P]}) \overset{(\log)}{\asymp} p(\{x \in [N] \,|\, dq(x)^2 \leq p_* \|q\|^2\}) \simeq P^{-(\alpha-1)} = d^{-(\alpha-1)/(2\rho\alpha+1)}.$$

Finite data. Let us now consider the empirical scheme (5), with $P \in \mathbb{N}$ and $\rho \geq 0$

$$\hat{q}(x) = \hat{p}(x)^\rho \mathbf{1}_{x \in \mathrm{top}_P((x_t)_{t \in [T]})}, \qquad q(x) = p(x)^\rho \mathbf{1}_{x \in [P]}.$$

We basically proceed with the same technique but with the event E_x the probability that x belongs to the top P of the empirical frequencies. When dealing with a binomial distribution, one can enumerate all possible outcomes for the empirical frequencies. For a template $a \in \Delta_{[N]}$, we said that a sequence (x_t) is of type a if its empirical frequency is equal to a,

$$\mathcal{T}(a) = \{(x_t) \in [N]^T \,|\quad \forall\, x \in [N], \sum_{t \in [T]} \mathbf{1}_{x_t = x} = Ta(x)\}.$$

Some enumeration arguments that can be found in Cover and Thomas [5, Chapter 11] leads to

$$\mathbb{P}_{\mathcal{D}_T}((x_t) \in \mathcal{T}(a)) = |\mathcal{T}(a)| \exp(-T(H(a) + D_{\mathrm{KL}}(a\|p))) \leq \exp(-T \cdot D_{\mathrm{KL}}(a\|p))).$$

Hence, the probability that x does not belong to the top P of the empirical frequencies of (x_t) is bounded by

$$\mathbb{P}_{\mathcal{D}_T}(x \notin \text{top}_P(x_t) \in \mathcal{T}(a)) \leq \sum_{a \in \mathcal{A}} \exp(-T \cdot D_{\text{KL}}(a \| p))),$$

where $\mathcal{A}$ is the set of all templates a where x is not in the top P of $(a(x'))_{x' \in [N]}$. With T samples over N elements there is at most $(N+1)^T$ different type templates, hence

$$\sum_{a \in \mathcal{A}} \exp(c_a \cdot T) \leq (T+1)^N \sup_{a \in \mathcal{A}} \exp(c_a \cdot T) = \sup_{a \in \mathcal{A}} \exp(c_a \cdot T + N \log(T+1)).$$

As a consequence,

$$\mathbb{P}_{\mathcal{D}_T}(x \notin \text{top}_P(x_t) \in \mathcal{T}(a)) \leq \sup_{a \in \mathcal{A}} \exp(-T \cdot D_{\text{KL}}(a \| p)) + N \log(T+1))$$

We are left with the computation of the "information projection distance" between p and the set of distribution where x does not belong to the top P. In order to get x out of the top P of p one should switch $p(x)$ with $p(P)$, which leads to

$$D_{\text{KL}}(p' \| p) = p(x) \log(p(x)/p(P)) + p(P) \log(p(P)/p(x)) = (p(x) - p(P)) \log(p(x)/p(P))$$

When considering $x < P/2$ and p following a Zipf law we get

$$D_{\text{KL}}(p' \| p) \geq (p(x) - p(2x)) \log(p(P/2)/p(P)) \geq c_\alpha x^{-\alpha}(1 - 2^{-\alpha}) \alpha \log(2) = c'_\alpha p(x)$$

where $c'_\alpha = c_\alpha(1 - 2^{-\alpha})\alpha \log(2)$. As a consequence, for any $P \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_T}[\mathcal{E}(f_{\hat{q}})] \leq c_0 P^{-\alpha+1} + \sum_{x \in [P/2]} p(x) \mathbb{P}(f(x) \neq f_*(x)). \\
&\leq c_0 P^{-\alpha+1} + (T+1)^N \sum_{x \in [P/2]} p(x)(\exp(-T c'_\alpha p(x)) + \mathbb{P}(f(x) \neq f_*(x) \,|\, x \in \text{top}_P((x_t))) \\
&\leq c_0 P^{-\alpha+1} + (T+1)^N \exp(-2^\alpha T c'_\alpha P^{-\alpha}) + \sum_{x \in [P/2]} p(x) \mathbb{P}(f(x) \neq f_*(x) \,|\, x \in \text{top}_P((x_t)).
\end{aligned}$$

When $\rho = 0$, setting $P = \min(c_1 d, T^{-1/\alpha}/\log(T))$ with c_1 chosen so that all x stored in memory leads to $f_*(x) = f(x)$ gives to the right scaling with both T and d: up to logarithmic factors,

$$\mathbb{E}[\mathcal{E}(f_{\hat{q}})] \ . \ d^{-\alpha+1} + T^{-1+1/\alpha} + \exp(-c_3 \log(T)^\alpha + N \log(T+1)).$$

Because $\alpha > 1$, the last term decreases faster than any polynomial power of T, hence ends up being negligible in front of $T^{-1+1/\alpha}$.

For the case $\rho \in (0, 1]$ one can dissociate two events: the event where x belongs to the top $P/2$ empirical frequencies; the event where $\hat{p}(x) > p(x)/2$; and conclude with similar derivations as precedently

$$\begin{aligned}
\mathbb{E}[\mathcal{E}(f_{\hat{q}})] \leq\ & c_0 P^{-\alpha+1} + (T+1)^N \exp(-2^\alpha T c'_\alpha P^{-\alpha}) + c_4 T^{-1+1/\alpha} \\
&+ \sum_{x \in [P/2]} p(x) \mathbb{P}(f(x) \neq f_*(x) \,|\, x \in \text{top}_P((x_t)), \hat{p}(x) > p(x)/2).
\end{aligned}$$

Retaking previous arguments leads to the same scalings as the ones of Theorem 2 with respect to d and a scaling in $T^{-1+1/\alpha}$ with respect to T. This ends the proof of the mixed scaling with both finite data and finite memory capacity.

B.4 Proof of Theorem 3

The lower bound directly follows from (22) together with $Q_\infty = p_* \|q\|^2$ and the fact that q is invariant to rescaling, so the best we can do is fit as much memories P as we can until reaching $3(d+1) = p_* P$ leading to a scaling in $\int_P^\infty p(x) \, \mathrm{d}x = C_\alpha P^{-\alpha+1}/(\alpha+1)$.

B.5 Theoretical Approximation of Gradient Updates

Lemma 4. *The gradient of this loss with respect to W takes the following form,*

$$\nabla_W \ell(x, y; W) = -(1 - p_W(y|x))(u_y - \varepsilon)e_x^\top, \quad \textit{with} \quad \varepsilon = \sum_{z \in [M]} p_W(z|x, z \neq y) u_z. \tag{43}$$

where $p_W(y|x) \propto \exp(u_y^\top W e_x)$ are model predictions for the current W.

Proof. The cross-entropy loss is written as

$$\ell((x, y, W) = -\log(\frac{\exp(u_y^\top W e_x)}{\sum_{z \in [M]} \exp(u_z^\top W e_x)}) = -u_y^\top W e_x + \log(\sum_{z \in [M]} \exp(u_z^\top W e_x)).$$

Hence stochastic gradient descent will update the matrix W by adding terms of the form

$$\begin{aligned}
\partial_W \ell((x, y), W) &= -u_y e_x^\top + \frac{\sum_{z \in [M]} \exp(u_z^\top W e_x) u_z e_x^\top}{\sum_{y \in [M]} \exp(u_y^\top W e_x)} \\
&= -u_y e_x^\top + \sum_{z \in [M]} p_W(z|x) u_z e_x^\top \\
&= -(1 - p_W(y|x)) u_y e_x^\top + \sum_{z \neq y} p_W(z|x) u_z e_x^\top \\
&= -(1 - p_W(y|x))(u_y e_x^\top - \sum_{z \neq y} \frac{p_W(z|x)}{1 - p_W(y|x)} u_z e_x^\top).
\end{aligned}$$

Note that $p_W(z|x)/(1 - p_W(y|x)$ corresponds the the probability of the z conditioned with respect to x under the event that z is not y, formally

$$\frac{p_W(z|x)}{1 - p_W(y|x)} = p(z|x, z \neq y).$$

Finally,

$$\begin{aligned}
\partial_W \ell((x, y), W) &= -(1 - p_W(y|x))(u_y e_x^\top - \sum_{z \neq y} p_W(z|x, z \neq y) u_z e_x^\top) \\
&= -(1 - p_W(y|x))(u_y e_x^\top - \mathbb{E}_{z \sim p_W}[u_z|x, z \neq y] e_x^\top).
\end{aligned}$$

While, it is clear that the model (2) does not describe the solution found by cross entropy, one might hope that the term $\mathbb{E}[u_z] e_x^\top$ will somewhat cancel themselves out and be an order of magnitude smaller than the leading term $u_y e_x^\top$. □

When $p_W(z|x)$ does not change much for all $z \neq f_*(x)$, since u_z were sampled at random in $\mathcal{S}^d$, we expect ε (43) to concentrate around zero with $\|\varepsilon\|^2 \approx 1/M$, hence to be negligible in front of $u_{f_*(x)}$. As a consequence,

$$\nabla_W \ell(x, f_*(x); W) \approx -(1 - p_W(f_*(x)|x)) u_y e_x^\top.$$

This is notably the case for $W = 0$, random W, or if W only stores pairs $(x, f_*(x))$ with $d \gg N$.

For a batch of n data $B = [x_1, \cdots, x_n]$, a gradient update with step size γ_t updates W_t as

$$W_{t+1} = W_t - \gamma_t \sum_{x \in B} \nabla_W \ell(x, f_*(x); W_t). \tag{44}$$

With the approximate updates (10), one steps of SGD with batch size one lead to an update, assuming $\exp(u_z W e_x) \approx 1$ for any $z \neq f_*(x)$,

$$q_{t+1}(x) - q_t(x) = \mathbf{1}_{x_t = x} \gamma \cdot (1 - p_{W_{q_t}}(f_*(x)|x)) \approx \frac{\mathbf{1}_{x_t = x} \gamma}{1 + (M - 1)^{-1} \exp(q_t(x))},$$

After T steps, each x will be seen $Tp(x)$ times on average, which explains (11).

Similarly, very large batch size $b = |B|$ and T/b update steps, each x will appear in each batch about $bp(x)$ times, which leads to the rough approximation

$$q_{\gamma,b}(x) = f^{T/b}(0) = \underbrace{f \circ f \circ \cdots \circ f}_{T/b \text{ times}}(0), \qquad \text{where} \qquad f : x \mapsto x + \frac{\gamma b p(x)}{1 + M^{-1}\exp(x)}. \tag{45}$$

In practice, we can approximate the effect of a batch by counting how many times x was in this batch and setting $bp(x)$ to be the exact count, which will lead to tighter approximation.

C Experimental details

C.1 Maximal Parameters Updates

In order to carefully choose step-sizes that scale well with width d in optimization algorithms, we follow [35] and consider learning rates consistent with maximal feature learning updates. Here we consider the following initializations:

- W is initialized as a Gaussian random matrix with $\mathcal{N}(0, \frac{1}{d})$ entries.
- Input embeddings e_x and output embeddings u_y are initialized as either random on the unit-sphere in d dimensions, or with Gaussian $\mathcal{N}(0, \frac{1}{d})$ entries. In both cases, every embedding has norm ≈ 1.

Updates to W. The updates to the matrix W look as follows:

- SGD with step-size η_W:

$$W' = W + \eta_W \delta W, \quad \delta W = \sum_j \alpha_j u_{y_j} e_{x_j}^\top,$$

  with $\alpha_j = \Theta_d(1)$, and a dimension-independent number of elements in the sum. Choosing $\eta_W = \Theta(1)$ then ensures that for any input embedding e_x, we have $\|W' e_x\| = \Theta(1)$ as desired.
- Adam (idealized here as signSGD) with step-size η:

$$W' = W + \eta_W \operatorname{sign}(\delta W), \quad \operatorname{sign}(\delta W)_{ij} = \frac{\delta W_{ij}}{|\delta W_{ij}|}.$$

  The coordinates of $\operatorname{sign}(\delta W)$ are now $\Theta(1)$ instead of $\Theta(1/d)$, thus the step-size needs to be taken as $\eta_W = \Theta(1/d)$ in order to satisfy $\|W' e_x\| = \Theta(1)$ (see [35, 34] for more details)

Updates to embeddings. The updates to embeddings look as follows:

- SGD updates:

$$u_y' = u_y + \eta_u \delta u_y, \quad \delta u_y = \sum_j \alpha_j W e_{x_j},$$
$$e_x' = e_x + \eta_e \delta e_x, \quad \delta e_x = \sum_j \alpha_j' W^\top u_{y_j},$$

  with $\alpha_j = \Theta(1)$ and a dimension-independent number of js. Since the algorithm ensures $\|W e_{x_j}\| = \Theta(1)$ and $\|W^\top u_{y_j}\| = \Theta(1)$ throughout training, choosing $\eta_u, \eta_e = \Theta(1)$ ensures that these conditions continue to hold after each update.
- Adam/signSGD updates:

$$u_y' = u_y + \eta_u \operatorname{sign}(\delta u_y), \quad (\operatorname{sign}(\delta u_y))_i = \frac{(\delta u_y)_i}{|(\delta u_y)_i|},$$
$$e_x' = e_x + \eta_e \operatorname{sign}(\delta e_x), \quad (\operatorname{sign}(\delta e_x))_i = \frac{(\delta e_x)_i}{|(\delta e_x)_i|}.$$

  Since the updates have coordinates of order $\Theta(1)$, in order to ensure that embeddings remain of norm $\Theta(1)$ after each update, we thus need $\eta_u, \eta_e = \Theta(1/\sqrt{d})$.

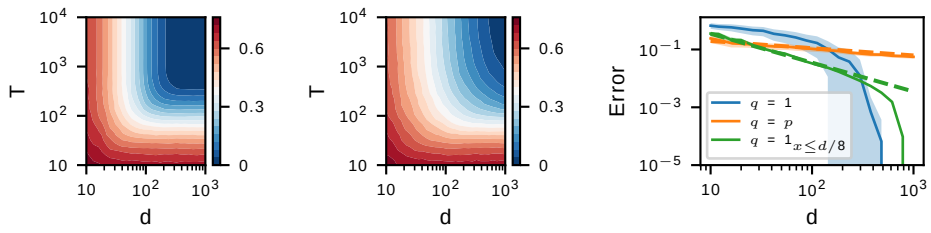

Figure 8: Generalization error (3) as a function of d and T for the model (2) averaged over 100 runs. The data follows a Zipf law with $\alpha = 0.5$, $N = 100$, $M = 5$ and $f(x) = x \bmod M$. Left: error for q_0 (4), either d is too small and there will be memory overflow leading to large error, either it is big enough and with enough data, the error will be null. Middle: error for q_1 (4), for small d and big T, it avoid memory overflow allowing a smaller error then q_0; however for big d it does not allocated enough memory to rare association, leading to a bigger error. Those results can be interpreted mechanistically by looking at the corresponding memory matrices (see Figure 10). Right: Generalization error when $T = +1$, $N = 100$ and $\alpha = 2$: the scheme q_0 leads to a zero-one type of plot where if $d < N$ the error is high, and if $d > N$ the error decreases fast to zero (in blue); the scheme q_1 leads to an error decreasing in $d^{(\alpha-1)/2\alpha} = d^{1/4}$ as predicted by theory (in orange); the scheme $q_{0,P}$ (5) with $P = d/8$, decreases in $d^{(\alpha-1)} = d^{1}$ until reaching the tipping point when $d/8 > N$ (in green).

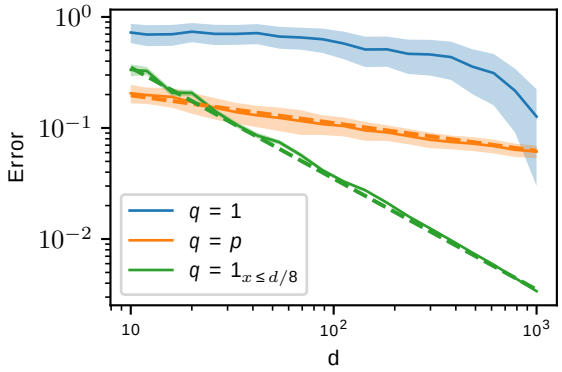

Figure 9: Same figure as the right one of Figure 8 yet with a bigger N, here $N = 1000$. The dashed curves represent $E = .35\ d^{1/4}$ (orange) and $E = 3.5\ d^{1}$ (green). They validate the scaling predicted by theory where we used $N = +1$ to get tight polynomial scalings of E (3) with respect to d.

C.2 Additional figures

Our theory predicted optimal scaling laws in $d^{-1+\alpha}$. However, there are some catches behind the proof:

- The lower bound is true when $N = 100$, otherwise the error can actually reach zero when d becomes larger than a tipping number d_t which compares to N. This fact is illustrated on Figure 8. Increasing N augments the tipping point d_t, rectifying the learning curve as illustrated on Figure 9.
- This was proven for models where $q(x, y) = q(x)$, and where $q(x)$ is not optimized with respect to $f_*(x)$. As such, it is not clear if those lower bounds hold for optimization-based algorithms, although we argue that we do not expect different mechanisms to take place in the proofs. We illustrate this empirically in the left of Figure 11.

Similarly, the unreasonable effect of learning the embeddings would be highly disappointing if those were hard to optimize in practice. The right of Figure 11 illustrates how with a few steps, one can achieve a zero generalization error when learning the embeddings. And Figure 12 shows visually the learned embeddings when $d = 2$.

In order to better understand gradient updates, Figure 14 shows the dynamic of the association memory W updated with SGD and a large step size. To validate the approximation (11), Figure 3 plots the generalization error associated with SGD and its theoretical approximation, while Figure 4 illustrates the idealized association scheme q_γ associated with a step size γ, batch size one and a Zipf law on $x \in [N]$. We equally resort to numerical simulation to study how optimization manages to

Figure 10: Representation of the weight matrix $(u_y^\top W e_x)_{y,x} \in \mathbb{R}^{M \times N}$ for $N = 10$, $M = 5$, $f(x) = x \bmod M$. The data x follows a Zipf-law with $= 1$ and $T = 10^3$. The matrix W is obtained according to (2) together with the scheme (4). Left: $= 0$ (4), $d = 10$, there is not enough memory capacity, and the model does not succeed to store memories, leading to a large generalization error. Middle left: $= 0$ (4), $d = 50$, there is enough memory capacity, we learn the right association $y = x \bmod M$. Middle right: $= 1$ (4), $d = 10$, the weighting q allows to store the most important memories beside having a small memory capacity. Right: $= 1$ (4), $d = 50$, the weighting q is too strong which does not allow to store memory associated with rare association (bottom of the matrix).

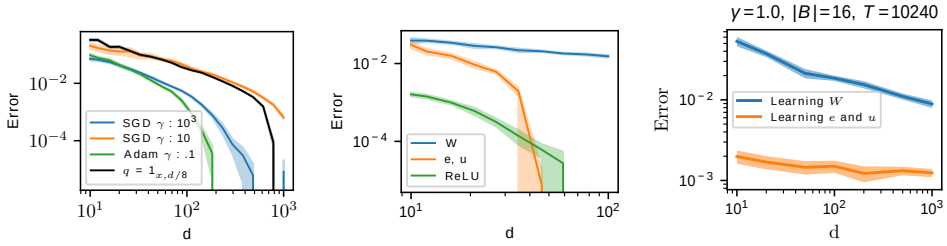

Figure 11: Scalings with respect to d for optimization-based algorithms, in the setting of Figure 8. Left: optimization-based algorithms beat the best algorithm designed by hands with $q(x;y) = q(x)$. Note how the curve seems to have the same optimal exponent $E \quad d^{\alpha+1}$ (the left part of the figure show similar slopes for all curves) yet with smaller constant in front, leading to earlier typing point before reaching zero generalization error due to full storage of all the associations. Middle: Comparison of learning the sole matrix W (blue), or learning the embeddings e and u (orange), together with the possibility to use non-linear model $u_y \mathsf{ReLU}(e_x)$ with e and u learned (green). All curves are obtained after 10^3 updates with batch size 10^3. Right: Comparison with the same setting as Figure 5. Learning the embeddings or going non-linear allows to impressively optimize memory storage, leading to better exponent with respect to d and earlier tipping point for a size number of updates.

rearrange memories when $d < N$. Figure 6 showcases two types of behaviors depending on the size of γ. *(i)* When the learning rate γ is large, associations will be stored easily in memory, but will tend to overwrite previous storage. *(ii)* When the learning rate γ is small, associations need to be seen often to build up in the matrix W (2) which will take more time, but will not erase memory. This provides another intuition explanation for why a bigger step size leads to better results on the left of Figure 5.

Adam. We have seen before how the update of SGD with large batch can be approximated with

$$\gamma_t^{-1}(W_{t+1} - W_{t-1}) = \sum_{x \in B} (1 - p_W(f_*(x)|x)) u_{f_*(x)} e_x^\top \approx \sum_{x \in \mathbb{N}} |B|(1 - p_W(f_*(x)|x)) p(x) u_{f_*(x)} e_x^\top .$$

Those naive updates would lead to a model that resembles (2) with $q = p^\rho$ for $\rho \approx 1$ (4). In concordance with previous research on the matter [36, 16], we found Adam to be helpful in our setup as well, see Figure 5 (right). In first order approximation, Adam is approximated as signSGD [3]. Arguably, this introduces a normalization effect to the gradient, helping to reach the saturation phase of $n \mapsto f^n$ (11) shown on Figure 4, homogenizing the resulting matrix W to behave similarly to $q_1 = 1$, therefore optimizing memory capacity. To underpin this intuition, we compare Adam with plain SGD and SGD with rescaled variance on population data. That is, we consider gradient descent with $\nabla_W \mathcal{L}(W)$ (9). The rescale variance SGD, consists in dividing the gradient by the variance of $\nabla_W \ell(X, f_*(X); W)$ (43) when $X \sim p$. For simplicity, we consider Adam with $\beta_1 = \beta_2 = 0$, in which case, it equates sign SGD, i.e., SGD when considering the sign of each entries of $\nabla_W \mathcal{L}(W)$ in the updates $W_t \to W_{t+1}$. Figures 15 and 16 reinforce our intuition that the usefulness of Adam lies

Figure 12: Experiments with learned embeddings when $= 2$, $N = 100$ and $M = 5$ with $y = f(x) = x \bmod M$ and $d = 2$. Left: level lines of the function $\mathbb{R}^2 \to [5]; u \mapsto \arg\max_{y \in [5]} u_y^\top u$ with u_y the learned unembedding. Middle: scatter plot of the learned input embeddings $e_x \in \mathbb{R}^2$ for $x \in [N]$ colored accordingly to $f(x)$ for e_x. It illustrates how the input embeddings match with the output ones, similarly to (12). Right: learned input embeddings obtained with $M = 10$, and allowing again a zero generalization error. Reaching a zero error with $d = 2$ greatly contrasts with the condition $d \geq N$ needed to get to a zero generalization error when the embeddings are random.

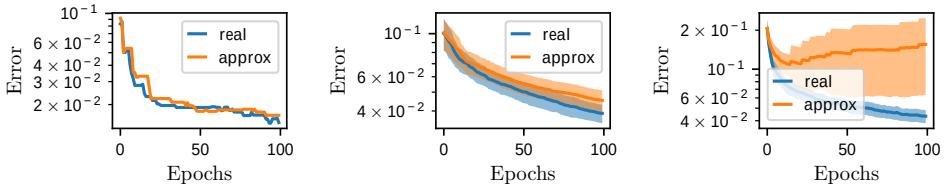

Figure 13: Same as Figure 3 yet with batch size equals one thousands $|B| = 10^3$.

in its ability to rescale gradient update, an effect that could equally be obtained by tuning the learning rate.

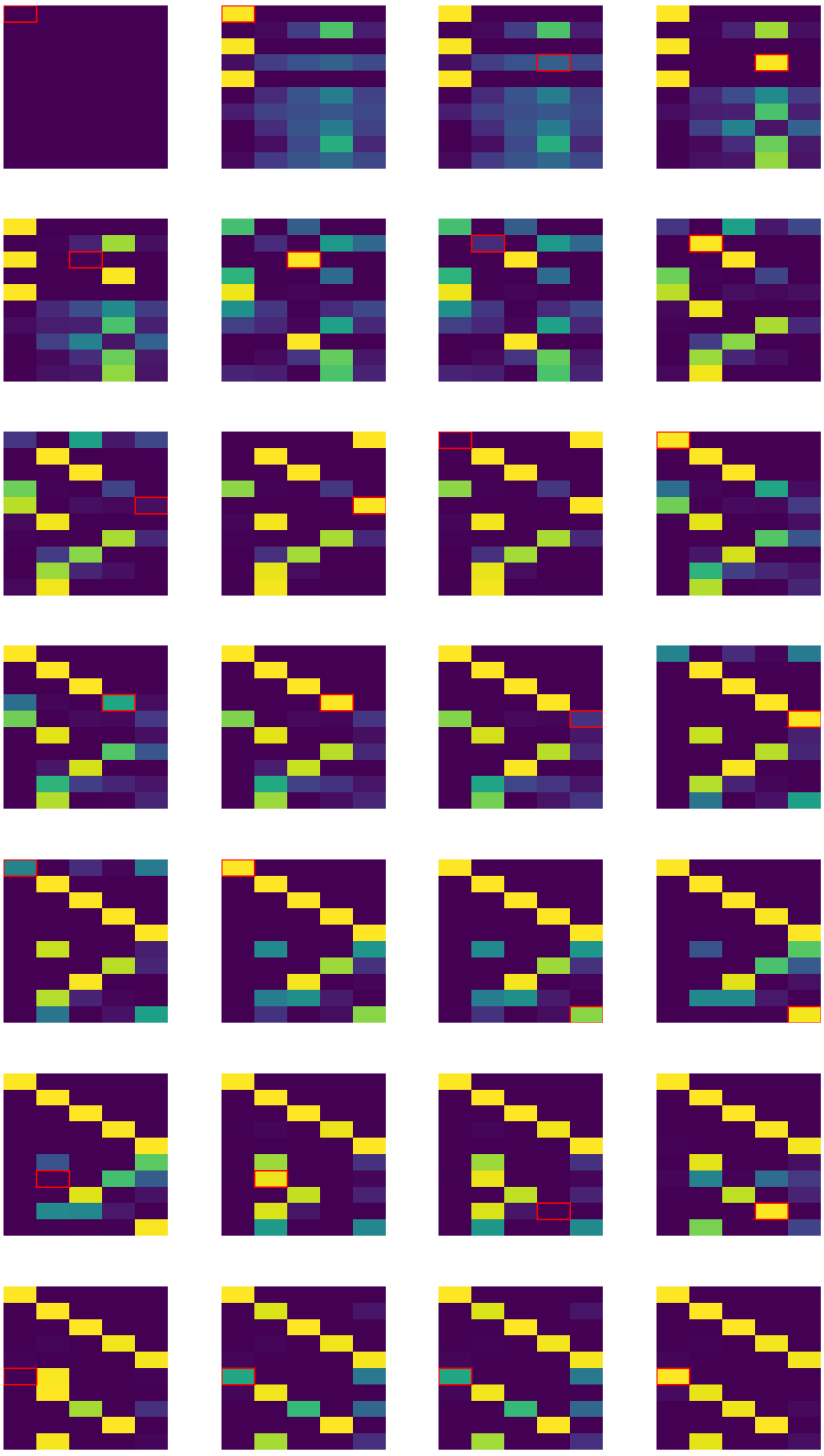

Figure 14: Gradient descent dynamics similar to Figure 6 with $d = 10$ and a fixed step size $= 10$. From time to time, we represent here $t \in \{0, 4, 5, 6, 8, 9, 11, 30, 32, 37, 49, 62, 75, 90\}$, stochastic gradient descent will hit an association that is not properly stored in memory yet (the red boxes). It will consequently update the weight matrix $W_t \to W_{t+1}$ (side by side pairs) to store it. When d is big enough, here $d = 10$, W will end by storing correctly all associations, leading to perfect generalization for future examples.

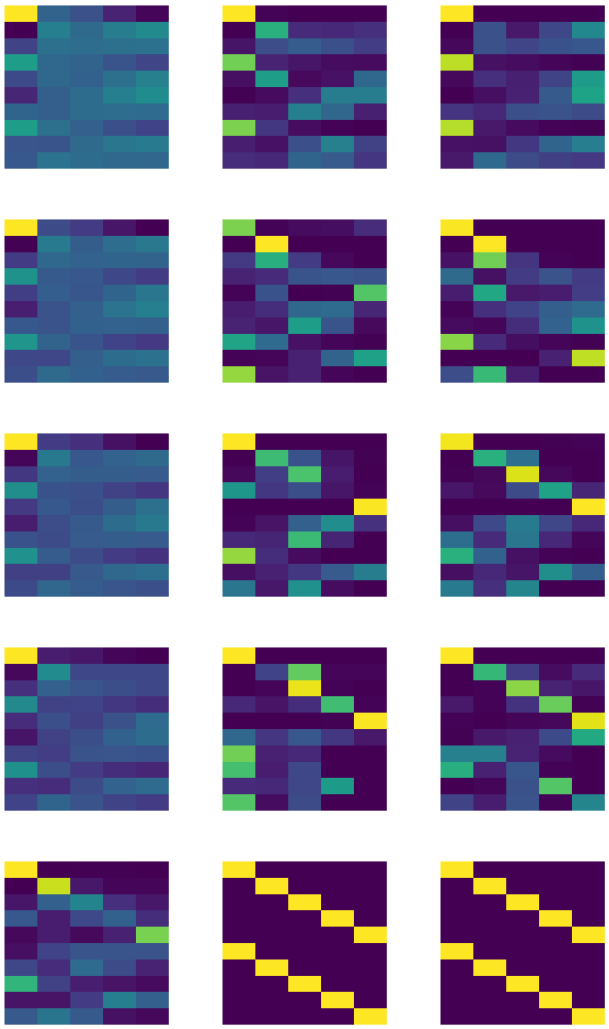

Figure 15: Comparison between SGD, signSGD and SGD with normalized variance on population gradient seen from the association matrix W_t at different times in the setting of Figure 14. The different rows correspond to the matrices W_t at time $t \in \{1, 2, 3, 7, 100\}$. Left: Plain SGD. Middle: Adam with $\beta_1 = \beta_2 = 0$, i.e., SignSGD. Right: SGD with normalized variance.

Figure 16: Left: Generalization error in the setting of Figure 15. Observe how SGD with rescaled variance (in green), an effect that can be done with SGD after adapting the learning rate, actually performs better than sign SGD (i.e., Adam with $\beta_1 = \beta_2 = 0$). Right: Variance of SGD along the training. As the training goes, SGD is losing momentum due to smaller gradient variances, hence smaller updates.