KNOWLEDGE DISTILLATION FOR RANDOM DATA: SOFT LABELS AND SIMILARITY SCORES MAY CONTAIN MEMORIZED INFORMATION

Freya Behrens
Statistical Physics Of Computation Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
freya.behrens@epfl.ch

ABSTRACT

This work reexamines conventional views of how neural networks store and transfer memorized information by investigating knowledge distillation for random, unstructured data. While knowledge distillation typically focuses on transferring generalizable patterns, we demonstrate that teacher models can encode and transfer purely memorized associations on finite random i.i.d. datasets. Through systematic experiments with fully connected networks, we show that students trained on teacher logits or embedding similarities achieve non-trivial accuracy on memorized data they never directly observed. This phenomenon persists across varying network capacities, dataset compositions, and even with randomized real-world data. Our findings encourage moving beyond simple key-value views of memory in neural networks, and highlight the role of spurious yet learnable patterns that transfer across models.

1 INTRODUCTION

With the advent of foundation models, it has become more popular to exploit and adapt their capabilities to new settings or smaller models via knowledge and dataset distillation (Gou et al., 2021; Xu et al., 2024). Since using teacher logits as supervised soft labels was proposed by Hinton et al. (2015), a flurry of methods for different data modalities and architectures have incorporated this idea and others e.g. (Hsieh et al., 2023; Habib et al., 2024). While the goal of these advanced methods it to achieve sample and parameter efficiency in knowledge transfer from model to model, Qin et al. (2025) recently asserted that logit matching on its own is still competitive for modern architectures: A single logit vector can be worth a thousand images. However, despite some theoretical advances (Phuong & Lampert, 2019; Boix-Adsera, 2024), it still remains unclear what exactly the "dark knowledge"(Hinton et al., 2015) is that the logits seem to contain, or even how to quantify it.

Among other hypotheses, a common assumption is that logits are beneficial because they already encode a hidden structure that represents the data well. This implies that logits should largely be useful when the data distribution involves recurring patterns, and informative representations that allow for generalizing solutions. However, most of current large language and vision models involve not only the generalization on skills but also memorization of facts and associations. Transferring their capabilities to a different model therefore requires transferring knowledge on both aspects. While there has been a long line of literature dating back to Hopfield (1982) on how neural networks store associations, there is comparatively little work how memories can be transferred and compressed. Dataset distillation (Yu et al., 2023; Yang et al., 2024) aims to match the final performance without focusing on different skill modalities such as generalization and memorization. To fill this gap, our work asks the following: *"Do teacher logits encode memorized knowledge? – And if yes, can students pick up this non-trivial information?"*

We investigate this question experimentally, inspired by (Zhang et al., 2017), who examine how neural networks generalize well in vision tasks despite the ability to memorize randomly labeled training images. Subsequent work showed that networks trained on this random data still pick up image features that *generalize* (Maennel et al., 2020). Since we are interested in transferring

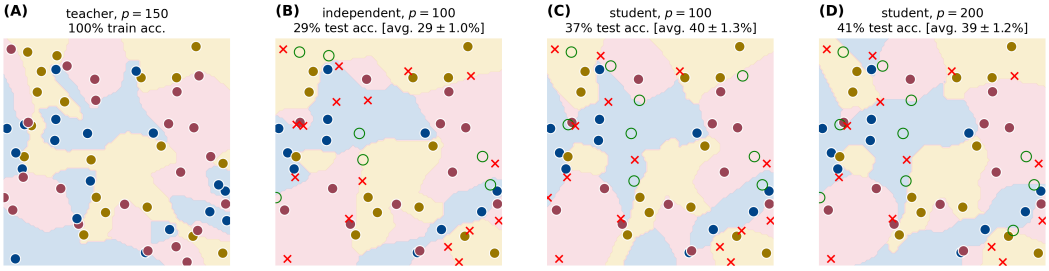

Figure 1: **Information leakage via logits.** We examine fully connected networks with ReLU activations and p hidden neurons. A teacher network is trained on 2-dimensional input data with i.i.d. random uniform labels drawn from $\{1, 2, 3\}$ (blue, crimson, yellow markers). **(A)** Visualizes the training data and decision boundaries, it memorizes this data perfectly and achieves 100% training accuracy. The teacher training data is then separated into a student-train and student-test part $(60\%, 40\%)$. We examine 3 settings: Training student networks via cross-entropy **(B)** on the class information only and **(C, D)** on the teacher logits. While a random network only achieves trivial accuracy of $\sim 30\%$, students that fit the teacher logits achieve *non-trivial test accuracy* of $\sim 40\%$. Red and green indicate data from the test set, and whether it was classified wrongly or correctly. The average test accuracy over 20 student initializations is given along with the standard error on the mean. Data that has not been seen by the teacher achieves $\sim 30\%$ accuracy.

memorized data, we consider teacher networks that memorized a finite number of random input-label pairs and examine knowledge that is distilled from them. More specifically, we train a student via logit or similarity score matching. Crucially, the student only has access to a subset of the memorized data and is tested on the other held out part. A simple example is shown in Fig. 1. To the best of our knowledge, in this context our experimental setup has not been used previously.

Using this method, we find that **(I)** via knowledge distillation and training on the teacher logits, a student can indeed obtain non-trivial information about memorized random data without full access to the full memorized dataset. **(II)** For the student to succeed, the finite random dataset needs to strike a balance between the input dimension, the number of labels per class and the overall number of dimensions. Generally, higher capacity students achieve higher test accuracies. **(III)** Beyond training on logits, we show that training a student to match the pairwise cosine similarity in the logit embedding space from teacher model can lead to a similar transfer of memorized data. However, this method is less efficient and leads to fewer correct predictions.

With these findings, we answer our introductory question *positively*: Teacher logits, dark knowledge, can contain and transfer information that we would intuitively consider memorized, since the original data was randomly sampled. Our observation challenges models of neural networks as simple key-value storage and motivates further investigation as to how spurious correlations (Ye et al., 2024) affect memorization, see the discussion in Section 4.

2 Setting and Notation

Data. We consider classification datasets with $i = 1, \ldots, n$ data samples, with inputs $\mathbf{x}_i \in \mathbb{R}^d$ and labels $y_i \in [1, \ldots, C]$ from c classes. Each input value is sampled i.i.d. uniformly from the range $[-1, +1]$. Since we are memorizing finite datasets, for a given dataset of size $n = C \cdot \texttt{samples\_per\_class}$ we assign exactly $\texttt{samples\_per\_class}$ labels to each class.
Models. All models are parameterized functions that map from the input dimension d to the number of classes c, i.e. $f_\theta(\mathbf{x}) = \mathbf{z}$, where $\mathbf{z}$ are the *logits*. The prediction is extracted by using the argmax over the logits. In this work except for Fig. 1 are 2-hidden layer ReLU neural networks with p hidden neurons in both layers.
Training. Training is conducted via the Adam optimizer (Kingma, 2014) with default pytorch settings. When not otherwise mentioned, we use the cross entropy loss for supervised training. For $\mathbf{y}_i \in \mathbb{R}^c$ being the one-hot encoded label vectors, the cross-entropy loss with temperature T is

$$\mathcal{L}_{\text{CE}}(\{\mathbf{x}_i, \mathbf{y}_i\}_n) = -\sum_i \sum_c (\mathbf{y}_i)_c \log\left[\sigma_T\left(f_\theta\left(\mathbf{x}_i\right)\right)_c\right] \;;\; \sigma_T(\mathbf{z})_c = \frac{\exp(z_c/T)}{\sum_{c=1}^{C} \exp(z_c/T)} \,.$$

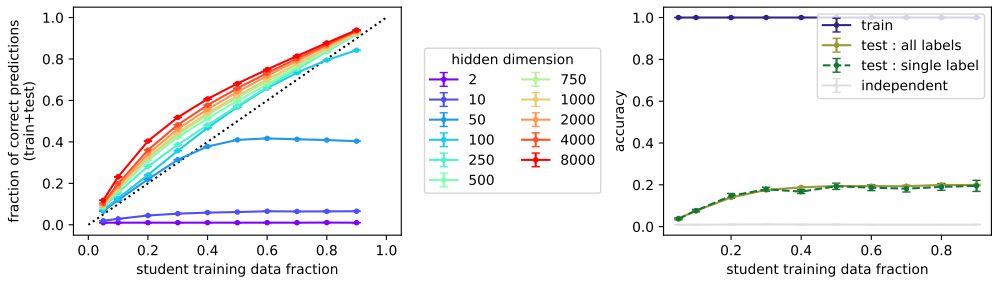

Figure 2: **Transferring memorized information under capacity constraints. (Left)** A student with 2 hidden layers of varying dimensions is trained on the logits of a given training fraction of the teacher training data (the teacher with $p = 500$ fits the complete training data perfectly). The fraction of correct predictions on the complete teacher training set is shown. Points above the diagonal line indicate that the student classifies more data correctly than seen from the teacher. **(Right)** A student with $p = 500$ trained on the teacher logits. The accuracy for the training set, the test set and the test set for only a single label. The independent accuracy is the accuracy achieved on data that is sampled fresh, and not seen by the teacher. Input dimension is 1,000 with 100 classes with 100 samples each in the teacher training data. Points are averages over 20 student initializations with the standard error on the mean.

To learn from a given teacher $t : \mathbb{R}^d \to \mathbb{R}^c$, we either train using logits or similarity scores. In the case of knowledge distillation via logits, we still use the cross-entropy loss, but instead of the ground truth $\mathbf{y}_i$ we use a given teacher network's logits as $\hat{y}_i = \sigma_T(t(\mathbf{x}_i))$.
For learning from similarity scores in an unsupervised context, we train models using a mean-squared-error loss on the cosine similarity of the embeddings, where we learn from a set of paired samples, as

$$\mathcal{L}_{\text{SIM}}(\{\mathbf{x}_i, \mathbf{x}_j\}) = \sum_{ij} (\rho[t]^{ij} - \rho[f_\theta]^{ij})^2 \; ; \; \rho[f]^{ij} = \frac{f(\mathbf{x}_i) \cdot f(\mathbf{x}_j)}{\|f(\mathbf{x}_i)\| \|f(\mathbf{x}_j)\|} \, .$$

A similar metric is is used to measure teacher-student similarity in e.g. (Li et al., 2024).

3 Knowledge Distillation for Memorized Data

In our experiments we randomly sample finite datasets $\mathcal{D} = \{\mathbf{x}_i, y_i\}_n$. A teacher model is trained to reach to perfect accuracy on this data – it perfectly memorized this random data, hence we call the full dataset $\mathcal{D}$ the *memorized data*. The *trivial accuracy* of the teacher the predictive performance on a different finite dataset $\mathcal{D}'$ which is sampled in the same manner. This accuracy is typically close to random guessing for random data, and represents the generalization error in conventional settings.
To train the student with an α-fraction of the memorized data $\mathcal{D}$, we randomly select for each class c an α-fraction of samples for the student to train on. The rest of the memorized data $\mathcal{D}$ is held out for testing. With some abuse of terminology we refer to the memorized data used for learning as *training* and the held-out memorized data as *test* data.

3.1 Supervised training with logit information

We first include the teachers logits in the training data for the student. In Fig. 2 we demonstrate how this strategy allows students of varying capacities to learn from a teacher with $p = 500$ hidden units. We vary the fraction of training data α and measure the fraction of memorized data that is predicted correctly by the student. This indicates that the student indeed learns data that was seen by the teacher but not by the student. In addition, higher capacity students are able to extract more information from the logits than lower capacity students. However, a higher capacity is not necessary: For some α a lower capacity of $p = 250$ is already enough to surpass trivial accuracy. Fig. 2, right, shows different measures of average accuracy for 20 initializations of students with $p = 750$. We observe that for all α it is able to memorize the training data, and even little training data allows for some non-trivial predictive accuracy on the test set. Further, the accuracy is distributed evenly over all 100 classes: As an example the test accuracy on the label 0 is shown, which matches the test accuracy over all test labels. Generally, we observe a strong dependence of the test performance

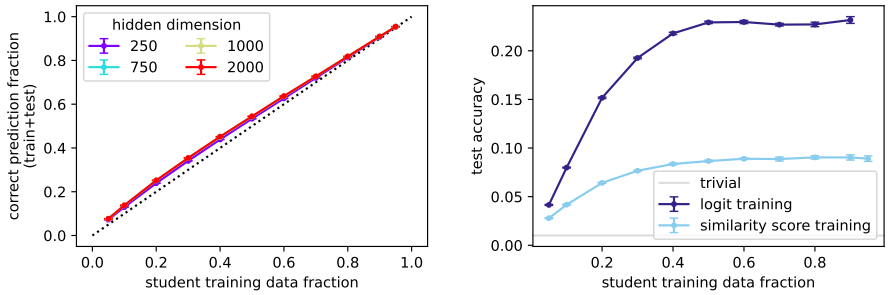

Figure 3: **Transferring memorized information via similarity scores.** In this setting the student learns embeddings for the training data with pairwise similarities that are close to those exhibited by the teacher that memorized the data. This training is done without labels in an unsupervised fashion. **(Left)** The overall number of correct samples that the student obtains after being trained from the similarity information stemming from the teacher. **(Right)** Comparison of the unsupervised training approach to the supervised training method from Fig. 2. The standard error on the mean is displayed for 20 runs per experiment.

both on the learning rate and the temperature of the softmax distribution, where in line with previous work on non-random data high temperatures seem to be advantageous (Nagarajan et al., 2024), which we show in Appendix A.1.
In Appendix A.2 we also examine how the test accuracy depends on the composition of the dataset, i.e. the input dimension, the number of samples per class and the overall number of classes. In order to compare the accuracy in different settings, we compute the ratio between the test accuracy and the trivial accuracy for a given number of labels. More samples per class are harder to memorize, and the input dimension needs to increase. Simultaneously, overly large input dimensions lead to lower test accuracy. For the optimal transferability, there seems to be a linear relationship where doubling the samples per class requires doubling the input dimension.
To understand where the information about the memorized data is located, we compare our results to three settings in which we modify the teacher information. In Appendix A.3, we (1) remove samples with a given class c from the training data, (2) set the probabilities $\sigma_T(\mathbf{z})_c$ manually to zero for samples that are not class c and (3) set the smaller fraction of $\sigma_T(\mathbf{z})$ to zero. We compare the general test accuracy to the class c test accuracy. In (1) and (2) the general test accuracy is not affected strongly. For (1) the class test accuracy is also maintained at a high level, similar to observations from label distillation on regular data (Qin et al., 2025), while for (2) it drops below the trivial accuracy. The memorized data about c must be contained in the other samples logits. For larger fractions of cut tails in (3) we observe a sharp drop in test accuracy starting from cutting 0.5% of the tail, indicating that the collective -rather than only the high-valued- logits are relevant for the non-trivial test-accuracy.

3.2 Unsupervised training with similarity information

In a second step, we consider transferring the memorized knowledge in an unsupervised fashion, by using the teacher-learned similarity scores of the embedded training data via $\mathcal{L}_{\text{SIM}}$. This approach is inspired by works from knowledge distillation (Passalis & Tefas, 2019) and representation learning, where regularizing with similarity scores from different foundation models are known to improve accuracy across data modalities (Huh et al., 2024). Here, we train the student with the similarity loss. While this means that the output embedding dimension can be of any size, we keep it to the number of classes. To obtain accuracy scores for the test data, we apply the student to the test sample to first obtain an embedding. Then nearest neighbor from the training dataset in this embedding space is used as a label. By definition the training accuracy is always 100% - each training sample is its own closest neighbor.

Fig. 3 shows that this procedure indeed also leads to some non-trivial test accuracies. We obtain marginal gains in accuracy on the overall memorized data, but this procedure proves less effective than the direct fitting of the logits. While this could indicate that the similarity scores contain less information, it is also possible that our training strategy is not optimal. For a better comparison, it would be interesting to understand whether both methods make use of similar information, by computing the overlap between their correct predictions given the same teacher – which we leave to future work.

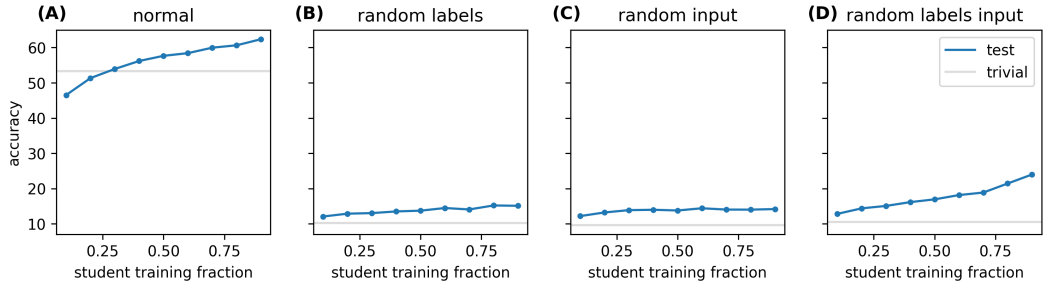

Figure 4: **Transferring normal and random labels for CIFAR-10.** The teacher and student are again 2-hidden with $p = 512$ hidden neurons respectively. Training is done with a temperature $T = 5$ and a learning rate of 0.01. In this setting $\mathcal{D}$ contains 80% of the CIFAR-10 train set, itself $50,000$ samples with 10 balanced classes. We train the teacher until 100% accuracy on the training data and compare the (A) uncorrupted data with settings where we (B) shuffle the class labels or (C) the pixels in the input, or (D) do both. Note that in this case the trivial accuracy is computed on the held out 20% of the CIFAR-10 train set, with the eventual random permutations applied.

3.3 Real world input distributions

In this section we conduct experiments for logit matching on more structured data, namely from the image domain via the CIFAR-10 dataset (Krizhevsky et al., 2009). In Fig. 4 we compare students that are distilled from different teachers: The teachers are trained on (A) the normal/clean data, (B) shuffled labels, (C) shuffled inputs and (D) shuffled labels and inputs. All teachers reach 100% accuracy on their memorized data set. In the setting with clean data, the trivial accuracy is higher than random guessing, at $\sim 52\%$, since the teacher learns to generalize to unseen data. Here, with enough training data, the student also surpasses this trivial accuracy of the teacher on the test data. This indicates it does not only learn the general patterns, but also some specific knowledge on the data seen by the teacher. For the settings with random permutations of the labels and image pixels we confirm the phenomena we observed for synthetic data previously: All students are able to reach a non-trivial accuracy on the test data. In addition, randomizing both labels and inputs of the images actually improves the student's ability to predict the test data. Appendix A.4 presents further results for tokenized random data with 1-layer transformers.

4 Discussion

In this work we provide evidence that both teacher logits and similarity scores between memorized data samples provide information beyond the samples themselves. This allows student models both in a supervised and unsupervised setting to reach a non-trivial accuracy on data that was memorized by the teacher but not seen by the student. While we provide evidence for several settings, it is not entirely clear yet to which generality our results hold, e.g. for facts that would be memorized by large foundation models. We discuss these limitations further in Appendix B.
While we understand that logit and similarity information can be useful for memories, we still do not have a conclusive answer on *how* exactly this works. This type of data is structural and relational by nature – and yet it can transfer i.i.d. random memorized data. A simple hypothesis can reconcile this fact with our observation: The neural networks simplicity bias towards generalizing solutions also holds for random data. When the dataset is finite and random, this would imply that the simplest pattern (even if it is fairly complicated) is used to fit the data. Fig. 1 mirrors this intuition: Even though there are many possibilities to fit the data, viewed from afar the decision boundaries of the unrelated models (A) and (B) look fairly similar. Even though these patterns and correlations are spurious and usually undesirable (Ye et al., 2024), we postulate that in the context of associative memories they are more use- than harmful, which possibly motivates the introduction of a term such as *neural mnemonics* to highlight the positive effect of spurious correlations.

On the one hand, our work motivates a more rigorous analysis of the information theoretically contained in different aspects of the data and the model's simplicity bias for finite random data. Does the neural networks simplicity bias towards generalizing solutions also holds for random data? On the other hand, our hypothesis challenges a common intuition, namely that random facts are learned by neural networks in a key-value style tabular memory. We motivate further investigations into how their (spurious) relations to one another come into play.

ACKNOWLEDGEMENTS

We thank Luca Biggio for interesting discussions and the reviewers of the NFAM and SCSL workshops at ICLR 2025 for useful feedback.

REFERENCES

Enric Boix-Adsera. Towards a theory of model distillation, May 2024.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021. ISSN 1573-1405. doi: 10.1007/s11263-021-01453-z. URL http://dx.doi.org/10.1007/s11263-021-01453-z.

Gousia Habib, Tausifa Jan Saleem, and Brejesh Lall. Knowledge Distillation in Vision Transformers: A Critical Review, February 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015.

J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes, July 2023.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The Platonic Representation Hypothesis, July 2024.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *arxiv*, 2009.

Xin-Chun Li, Wen-Shu Fan, Bowen Tao, Le Gan, and De-Chuan Zhan. Exploring Dark Knowledge under Various Teacher Capacities and Addressing Capacity Mismatch, May 2024.

Hartmut Maennel, Ibrahim Alabdulmohsin, Ilya Tolstikhin, Robert J. N. Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. What Do Neural Networks Learn When Trained With Random Labels?, November 2020.

Vaishnavh Nagarajan, Aditya Krishna Menon, Srinadh Bhojanapalli, Hossein Mobahi, and Sanjiv Kumar. On student-teacher deviations in distillation: Does it pay to disobey?, March 2024.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability, October 2023.

Nikolaos Passalis and Anastasios Tefas. Unsupervised Knowledge Transfer Using Similarity Embeddings. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3):946–950, March 2019. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2018.2851924.

Mary Phuong and Christoph Lampert. Towards Understanding Knowledge Distillation. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5142–5151. PMLR, May 2019.

Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A Label is Worth a Thousand Images in Dataset Distillation, January 2025.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A Survey on Knowledge Distillation of Large Language Models, October 2024.

William Yang, Ye Zhu, Zhiwei Deng, and Olga Russakovsky. What is Dataset Distillation Learning?, July 2024.

Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, and Aidong Zhang. Spurious Correlations in Machine Learning: A Survey, May 2024.

Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset Distillation: A Comprehensive Review, October 2023.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, February 2017.

A Additional Experiments

A.1 Learning Rate and Temperature

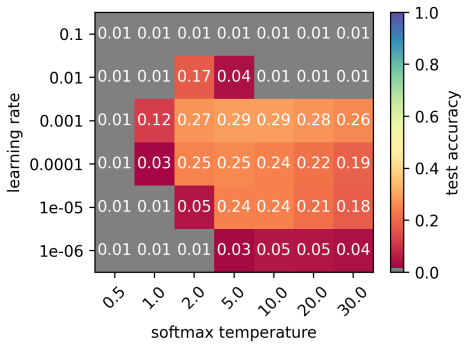

Figure 5: **Tuning the softmax temperature and learning rate.** A student with $p = 750$ trained on 40% of the teacher training data. The temperature of the softmax and the learning rate for training with Adam is varied. A higher temperature seems necessary, where correct predictions of smaller logits in the loss are weighted more compared to regular temperatures. Otherwise same setting as in Fig. 2.

A.2 Dataset Composition

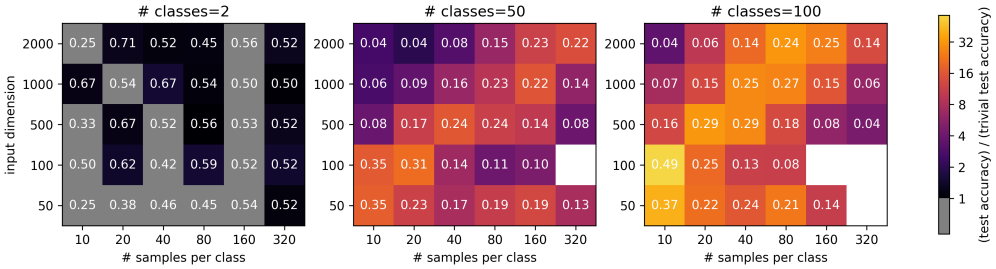

Figure 6: **Dataset composition impacts memory transferability.** From left to right the number of classes present in the memorized dataset are varied. In each plot, datasets with varying samples per class and input dimensions are tested. In every case, the teacher is trained to 100% accuracy. When training the teacher does not converge to 100% accuracy within a given number of epochs, the fields are left white. 40% of the memorized data are used as training data for a student via logits. The numbers in white on the fields represent the accuracy (as a fraction on remaining test data). The background colors represent the ratio between the test accuracy and the accuracy on data that stems from the same generation process but has not been memorized by the teacher. A ratio smaller than 1 (grey) indicates that the accuracy is trivial, whereas colorful accuracies indicate that some non-trivial information about the test set was acquired by the student.

A.3 Removing Information From the Logits

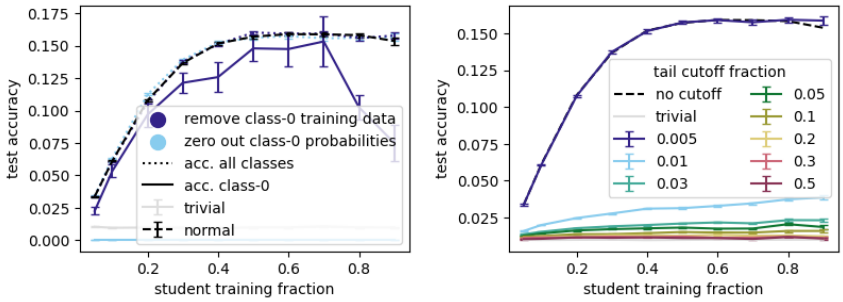

Figure 7: **Localizing memorized information in the logits.** A student ($p = 750$) learns from the teacher that memorized the training data. We show normal knowledge distillation via logits (black dashed) and compare it to some modification in logit/probability space. **(Left)** Either all data that has class 0 is removed (dark blue), or the probabilities that are fed into the cross-entropy are set to zero on all soft labels except for those samples with class 0 (light blue). **(Right)** The smallest fraction of logits per sample in the training data is cutoff, i.e. set to zero in its probability representation before being fed into cross entropy loss. The standard error on the mean for 20 runs is shown.

A.4 Tokenized Data

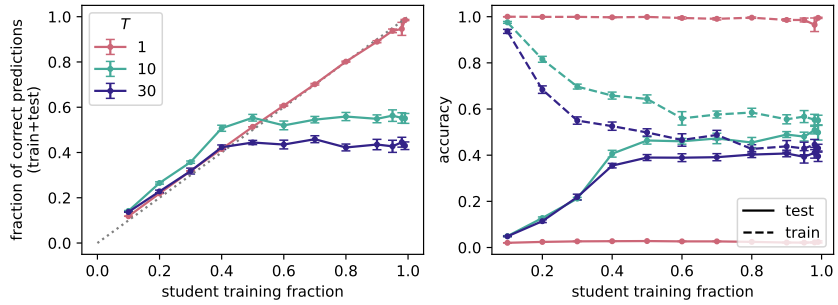

Figure 8: **Transferring memorized data for tokenized inputs with 1-layer transformers.** We use the modular addition and 1-layer transformer introduced in (Nanda et al., 2023) for modular addition. We take the dataset of adding two positive integers a, b modulo $k = 113$, where the integers range between 0 and k. In this experiment we train the transformer teacher to 100% accuracy and use a student of the same architecture. We train the student with different values of the temperature T for logit matching. In this setting we have not yet discovered a parameter configuration, in which the student achieves 100% train accuracy and non-trivial test accuracy. It is unclear why this is the case, and we will investigate this behaviour further. The standard error on the mean for 10 runs per experiment is shown.

B Limitations and Next Steps

Our study is limited to fully connected neural networks with ReLU activations. We have preliminary evidence that the type of activation function and the number of layers do not change the phenomenology of our results. The same holds for the match of these architectural details between the teacher and student, which seems to be not very important. However, a deeper investigation of these architectural variations would be desirable. This holds even more true, as we have some preliminary evidence that our findings (in the case of the CIFAR-10 input distribution) do not generalize to networks with convolutional input architectures, while it does uphold for the transformer architecture. Regarding the teacher, in this work we investigated teachers that were able to perfectly memorize the training data within a given budget or epochs. We did not incorporate the role of regularization such as early stopping or weight decay.