Memorization in Deep Learning: A Survey

Memorization in Deep Learning: A Survey

JIAHENG WEI, RMIT University, Melbourne, Australia
YANJUN ZHANG, School of Computer Science, University of Technology Sydney, Sydney, Australia
LEO YU ZHANG, Griffith University, Brisbane, Australia
MING DING, Data61, Eveleigh, Australia
CHAO CHEN, RMIT University, Melbourne, Australia
KOK-LEONG ONG, RMIT University, Melbourne, Australia
JUN ZHANG, Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Australia
YANG XIANG, Swinburne University of Technology, Hawthorn, Australia

Deep Learning (DL) powered by Deep Neural Networks (DNNs) has revolutionized various domains, yet understanding the details of DNN decision-making and learning processes remains a significant challenge. Recent investigations have uncovered an interesting memorization phenomenon in which DNNs tend to memorize specific details from examples rather than learning general patterns, affecting model generalization, security, and privacy. This raises critical questions about the nature of generalization in DNNs and their susceptibility to security breaches. In this survey, we present a systematic framework to organize memorization definitions based on the generalization and security/privacy domains and summarize memorization evaluation methods at both the example and model levels. Through a comprehensive literature review, we explore DNN memorization behaviors and their impacts on security and privacy. We also introduce privacy vulnerabilities caused by memorization and the phenomenon of forgetting and explore its connection with memorization. Furthermore, we spotlight various applications leveraging memorization mechanisms. This survey offers the first-in-kind understanding of memorization in DNNs, providing insights into its challenges and opportunities for enhancing AI development while addressing critical ethical concerns.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Security and privacy**;

Additional Key Words and Phrases: Neural networks, memorization, forgetting

ACM Reference Format:
Jiaheng Wei, Yanjun Zhang, Leo Yu Zhang, Ming Ding, Chao Chen, Kok-Leong Ong, Jun Zhang, and Yang Xiang. 2025. Memorization in Deep Learning: A Survey. *ACM Comput. Surv.* 58, 4, Article 98 (October 2025), 35 pages. https://doi.org/10.1145/3769076

Authors' Contact Information: Jiaheng Wei, RMIT University, Melbourne, Victoria, Australia; e-mail: wjheng1999@gmail.com; Yanjun Zhang, School of Computer Science, University of Technology Sydney, Sydney, New South Wales, Australia; e-mail: Yanjun.Zhang@uts.edu.au; Leo Yu Zhang, Griffith University, Brisbane, Queensland, Australia; e-mail: leo.zhang@griffith.edu.au; Ming Ding, Data61, Eveleigh, New South Wales, Australia; e-mail: ming.ding@data61.csiro.au; Chao Chen (corresponding author), RMIT University, Melbourne, Australia; e-mail: chao.chen@rmit.edu.au; Kok-Leong Ong, RMIT University, Melbourne, Victoria, Australia; e-mail: kok-leong.ong2@rmit.edu.au; Jun Zhang, Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Victoria, Australia; e-mail: junzhang@swin.edu.au; Yang Xiang, Swinburne University of Technology, Hawthorn, Victoria, Australia; e-mail: yxiang@swin.edu.au.

ACM 0360-0300/2025/10-ART98
https://doi.org/10.1145/3769076

1 Introduction

In the development of **artificial intelligence** (**AI**), **deep learning** (**DL**) has emerged as an effective solution for various complex tasks like text generation [2], speech translation [26], and so on. **Deep neural network** (**DNN**) as the main model architecture has been widely used in numerous innovative applications such as autonomous vehicles [45, 46, 107] and medical diagnosis [63, 111].

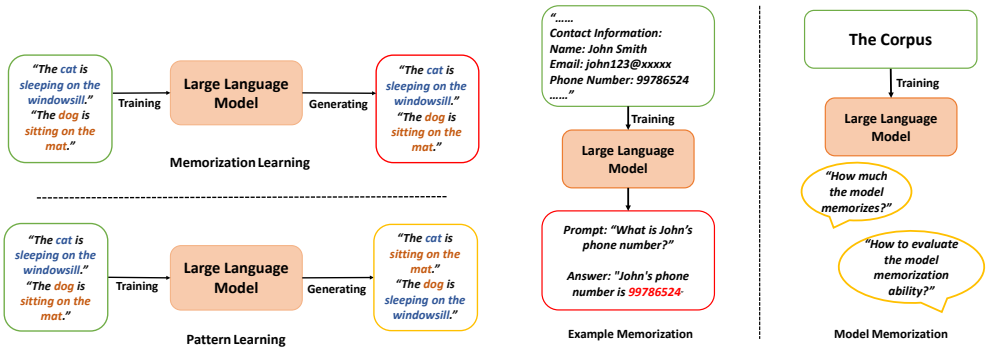
However, it is still challenging to understand how DNNs make decisions and what they learn from the training data. Though researchers believe DNNs can learn patterns in the training data to achieve success in assigned tasks, a recent study found that DNNs are able to memorize the entire randomly labeled training dataset [126], which illustrates that properties of the model family, or the regularization techniques fail to explain why large neural networks generalize well. DNNs may memorize particular features from training data instead of learning patterns to perform specific tasks. This attracts the community to explore the memorization mechanism and prompts researchers to rethink the generalization in DNNs. Additionally, this memorization phenomenon raises concerns about the security of AI because of potential privacy leakage risks and vulnerability against malicious attacks. Furthermore, the training dataset collected from the real world may contain significant noise and bias, and memorized data in DNNs may keep the noise and bias, impairing the usability and fairness of the models.

So far, numerous articles have found the memorization effects that neural networks may memorize some training data in training with gradient descent [15, 19, 31, 126, 131]. Current memorization studies mainly focus on two domains: the behaviors in standard training and the security/privacy risks. We summarize explicit memorization definitions in literature. However, there is a lack of a widely adopted definition for memorization, making describing and discussing the memorization concept challenging. Many relevant works provide inconsistent, sometimes contradictory, definitions of memorization. Especially, many works directly apply the word "memorization" as the synonymous words of "learning" and "fitting". Thus, we adopt the following terms for facilitating discussion: **Memorization Learning** refers to DNNs learning specific details or particular features of examples, while common **Pattern Learning** indicates DNNs learning the common patterns or generalized features of the data distribution. In Figure 1(a), we use a large language model to illustrate memorization learning and pattern learning. We use the word "generalization" to define the model performance on the new, unseen data. Suppose there is no extra explanation, all terms like "memorization", "memorization effect", and "memorization phenomenon" point to memorization learning. Moreover, we think pattern learning and memorization learning together constitute the learning path of DNNs.

Moreover, memorization is a complex concept that requires us to consider it at various levels. In our opinion, memorization learning and pattern learning operate at a feature level. However, understanding the features of neural networks directly is exceedingly difficult for humans. Hence, we mainly study memorization at the example level and model level as illustrated in Figure 1(b).

Intuitively, example memorization and model memorization indicate the objects of study are examples and models. Consequently, memorization concepts at different levels inspire distinct memorization evaluation methods. Example memorization evaluation tries to ensure if an example is memorized including differential evaluation, probabilistic evaluation and other memorization proxies. On the other hand, model memorization evaluation measures how much models memorize or the memorization ability of models. The model memorization evaluation methods mainly include noisy label evaluation, recurrence evaluation, and extraction evaluation.

After the definitions and evaluation methods, we systematically review related literature. For memorization behaviors in standard training, existing studies investigate the relationships between the memorization effect and training data, training stages, model architecture, overfitting, regularization, and other factors. One study [30, 31] provides an interesting conclusion that memorization



(a) Memorization Phenomenon (b) Memorization Level

Fig. 1. The Direct Memorization Effect. In (a), we use an image generator to describe memorization. The upper part demonstrates the memorization effect and the lower part represents the common generation. For (b), the memorization effect has two different levels: Example Memorization and Model Memorization.

learning improves the generalization of models because the memorization of rare and atypical examples actually contributes to the generalization performance of similar rare subgroups, which is adverse to some early opinions. Additionally, some evidence [18, 75, 108] shows overfitting is not responsible for memorization. Memorization is a persistent process in training. For security/privacy risks, the memorized particular features become multiple risk sources like membership inference risks [13, 101] and extraction risks [18, 19], enabling attackers to exploit the memorization mechanism to invade privacy and violate the security rules of DNNs. In contrast, some risks like adversarial attack risks are not obviously related to the memorization mechanism.

On a related aspect, the forgetting phenomenon is closely connected to the memorization effect. Thus, we also discuss and review the forgetting effect. We explore useful forgetting definitions and evaluation methods and summarize relevant forgetting phenomenon studies.

In addition, we also reviewed numerous applications that use memorization mechanisms. These applications like example enhancement, external memory architecture, noisy label learning, and model editing, privacy audit and protection, take advantage of different properties of memorization.

Comparison with related surveys. Recent surveys on memorization in machine learning and large language models have primarily concentrated on specific settings or narrow perspectives. For example, Xiong et al. [118] and Satvaty et al. [96] both focus on **large language models** (**LLM**), with Xiong et al. [118] analyzing mechanisms, detection, and mitigation strategies, while Satvaty et al. [96] providing a taxonomy of undesirable memorization in LLMs across different granularities. Li et al. [64] embedded memorization in a trustworthy ML framework, examining the tradeoffs between fairness, privacy, and robustness. Usynin et al. [112] examine the impact of memorization on model generalization and its data privacy implications, though the scope is relatively limited. A detailed comparison of related memorization surveys is summarized in Table 1.

In contrast, we attempt to organize the existing memorization definitions and proxies, aiming at building a scientific and effective framework and help the readers understand the memorization mechanism's impact on standard model training behaviors and the trustworthy learning domain. Additionally, we also explore the forgetting phenomenon and illustrate some potential applications of the memorization mechanisms. We hope this survey can help the research community have a general understanding of the memorization phenomenon.

Table 1. Comparison of Memorization Surveys

Survey	Scope	Focus	Distinctive Contribution
Xiong et al. (2025) [118]	LLMs	Mechanisms, privacy, legal issues	Rich taxonomy of LLM memorization, legal framing
Satvaty et al. (2025) [96]	LLMs	Undesirable memorization	Risk-centered; granularity-based taxonomy
Li et al. (2025) [64]	Trustworthy ML	Fairness, robustness, privacy	Places memorization at core of trustworthy ML trade-offs
Usynin et al. (2024) [112]	General ML	Memorization, generalization and privacy	Review memorization quantification, mechanism and privacy influence
Ours (2025)	**General DL**	**Definitions, proxy, mechanism, risks, forgetting, applications**	**Comprehensive review unifying definitions and proxies; synthesizes mechanisms, risks, mitigation strategies, and applications across deep learning**

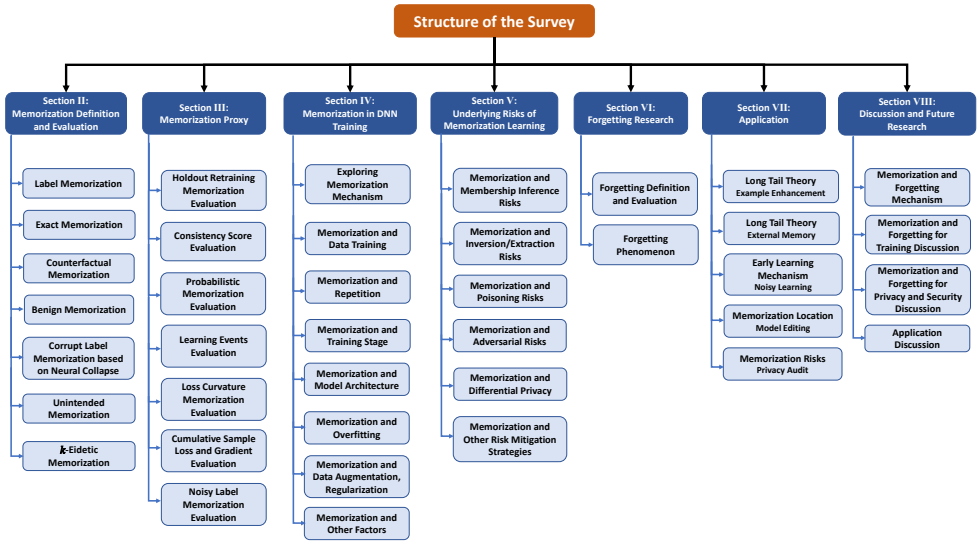


Fig. 2. Article structure.

The key contributions of this survey are summarized as follows:

- **Organizing definitions.** We propose a framework to organize all existing memorization definitions and evaluation methods. We also explain the scope and limitations of these definitions and evaluation methods.
- **Comprehensive review.** We review relevant memorization studies on its behaviors in the standard training and security/privacy risks. Moreover, we also investigate its connection with the forgetting studies and some possible applications.
- **Discussion.** In this survey, we thoroughly discuss the memorization mechanism and how memorization effects can boost other relevant technologies.

The survey is organized into the following sections, each focusing on a different aspect of memorization in deep learning as we present in Figure 2. Section 2 provides existing memorization definitions and Section 3 lists the existing memorization proxies. Section 4 delves into the memorization behaviors, presenting how memorization affects each training component and its relationship

Table 2. Main Memorization Definitions

Definition	Evaluation	Reference	Domain	Level	Research Question
Label Memorization	Differential Memorization	Feldman et al. (2020) [30, 31]	Generalization	Example	Memorization of long-tailed examples.
Exact Memorization	Definition	Tirumala et al. (2022) [108]	Generalization	Model	Learning in large language models.
Counterfactual Memorization	Differential Memorization	Zhang et al. (2021) [127]	Generalization	Example	Memorization in language models.
Benign Memorization	Definition	Anagnostidis et al. (2023) [6]	Generalization	Model	Memorization with data augmentation.
Corrupt Label Memorization based on Neural Collapse	Definition	Nguyen et al. (2023) [81]	Generalization	Model	Memorization in neural collapse.
Unintended Memorization	Recurrence Memorization	Carlini et al. (2019) [18]	Security	Model	Unintended memorization in training.
k-Eidetic Memorization	Extraction Memorization	Carlini et al. (2021) [19]	Security	Model	Privacy leakage in language models.

with overfitting, data augmentation, and regularization technology. Section 5 presents a review of memorization-associated risks that memorized particular features enhance privacy risks and mitigation strategies. Section 6 explores the forgetting phenomenon, which is the opposite of memorization. Section 7 demonstrates the underlying application of the memorization effects. Section 8 comprehensively discusses the memorization phenomenon's influence on standard training and security/privacy risks and how it enlightens and explains other technologies or phenomena.

2 Memorization Definition and Evaluation

Memorization is a vague and abstract concept, and difficult to obverse during the training of neural networks. Thus, previous studies did not provide a clear and uniform definition. Based on relevant research, we find that the motivations for studying the memorization phenomenon are its impact on generalization and the concerns about privacy and security risks. In this section, we outline the existing definitions of memorization along with the corresponding evaluation methods. A summary is presented in Table 2.

2.1 Label Memorization

Conceptual Definition. Intuitively, there would exist an obvious disparity when evaluating a data point on a model between the model memorizing the data point and not. Feldman [30] introduces the label memorization concept to describe the disparity in supervised learning tasks. Label memorization differentially defines what memorizing a label of a point in the dataset means.

Definition 1 (Label Memorization for Supervised Tasks). Given a training algorithm A that maps a training dataset D to a trained model f, the amount of memorization by A on example $(x_i, y_i) \in D$ is defined as

$$\text{mem}(A, D, (x_i, y_i)) := \Pr_{f \leftarrow A(D)}[f(x_i) = y_i] - \Pr_{f' \leftarrow A(D^{\setminus i})}[f'(x_i) = y_i], \tag{1}$$

where $D^{\setminus i}$ denotes the dataset D with (x_i, y_i) removed.

This definition provides a universal understanding of memorization and distinguishes generalized examples effectively. The definition actually approaches the nature of memorization that memorized examples cannot rely on learned patterns.

Evaluation Method - Differential Memorization Evaluation. Label memorization focuses on the phenomenon of example-level memorization. When memorization occurs, the model's output on a specific data point exhibits a significant discrepancy between the scenarios where the example is included in the training set and where it is excluded. This discrepancy can be utilized to evaluate memorization. The corresponding metric is referred to as the memorization score, defined according to Definition 1.

The memorization score quantifies the performance gap for an individual example when it is present versus absent in the training data. A substantially large gap suggests that other training examples do not provide informative features for that particular instance, indicating that the model is likely memorizing the example.

2.2 Exact Memorization

Conceptual Definition. Exact memorization proposed by Tirumala et al. [108], is used to perform a large-scale study of the dynamics of memorization over training. Additionally, this definition is only applied to the language models and attempt to measure model memorization.

Definition 2 (Exact Memorization). Let V denote the vocabulary size. Let C denote a set of contexts, which can be thought of as a list of tuples (s, y) where s is an input context (incomplete block of text) and y is the index of the ground truth token in the vocabulary that completes the block of text. Let S denote the set of input contexts, and let $f : S \to \mathbb{R}^V$ denote a language model. A context $c = (s, y) \in C$ is memorized if $\arg\max(f(s)) = y$. For a given set of contexts C (i.e., a given training dataset), the proportion of memorized contexts can be represented as

$$\text{mem}(f) = \frac{\sum_{(s,y)\in C} 1\{\arg\max(f(s)) = y\}}{|C|}. \tag{2}$$

Based on the formula, we know that the exact memorization actually represents accuracy since it just measures the average number that predicted token matches the ground truth token in the contexts. Thus, this definition is not related to the memorization phenomenon and cannot describe the memorization.

Evaluation Method - Definition Evaluation. As previously discussed, exact memorization measures the accuracy with which the predicted token matches the ground truth token. Therefore, Definition 2 can be directly used to quantify exact memorization.

2.3 Counterfactual Memorization

Conceptual Definition. Counterfactual memorization is extended from label memorization to unsupervised tasks. Zhang et al. [127] introduce the definition, applying it to quantify the episodic memorization in language models.

Definition 3 (Counterfactual Memorization). Given a training algorithm A that maps a training dataset D to a trained model f, and a measure M which measures the performance of x_i on f, the amount of memorization by A on example $x_i \in D$ measured with M is defined as

$$\text{mem}(A, D, M, x_i) := \mathop{\mathbb{E}}_{f\leftarrow A(D)}[M(f, x_i)] - \mathop{\mathbb{E}}_{f'\leftarrow A(D^{\setminus i})}[M(f', x_i)]. \tag{3}$$

where M can be per-token accuracy that f predicts the next token based on the preceding tokens, then measures the 0-1 loss.

Evaluation Method - Differential Memorization Evaluation. Counterfactual memorization is a universal version of label memorization and we can measure example memorization in the

unsupervised tasks. Consequently, differential memorization can also be assessed using the unsupervised formulation presented in Definition 3 to quantify counterfactual memorization.

2.4 Benign Memorization

Conceptual Definition. Benign memorization describes the phenomenon that neural networks can learn useful features on the randomly labeled dataset with data augmentation technology [6]. This work regards the general neural network structure as an encoder-projector pair and trains the pair on an augmented noisy dataset. If the accuracy of kNN probing at the embedding vectors of the encoder increases over probing at initialization, this is benign memorization.

Definition 4 (Benign Memorization). Here are two datasets, $D := (x_i, y_i)_{i=1}^n$ denotes the original clean dataset and $\tilde{D} = (x_i, \tilde{y}_i)_{i=1}^n$ its randomly labeled version. We call an encoder-projector pair $(h_{\phi*}, g_{\psi*})$ a memorization of $\tilde{D}$, if the network f_* perfectly fits $\tilde{D}$. Moreover, we call $(h_{\phi*}, g_{\psi*})$ a malign memorization if additionally, probing of $h_{\phi*}$ on D does not improve over probing at initialization. On the contrary, we call $(h_{\phi*}, g_{\psi*})$ a benign memorization of $\tilde{D}$ if probing of $h_{\phi*}$ on D outperforms probing at initialization.

This definition focuses on the generalization performance when training on randomly labeled datasets. Benign memorization occurs if the encoder learns generalized features. Therefore, this memorization definition is auxiliary to explain noisy label learning rather than a general memorization definition.

Evaluation Method - Definition Evaluation. Benign memorization examines whether a dataset with random label dataset can offer useful features. Consequently, it can be evaluated by measuring the accuracy of the embedding encoder $h_{\phi*}$ on a clean test dataset D_{test} using kNN probing. The evaluation metric is defined as

$$\text{mem}(h_{\phi*}, D_{test}) = Acc(h_{\phi*}, D_{test}), \tag{4}$$

where $h_{\phi*}$ is the encoder trained on the randomly labeled dataset $\tilde{D}$.

2.5 Corrupt Label Memorization based on Neural Collapse

Conceptual Definition. Empirical evidence indicates that the memorization of noisy data points may lead to degradation (dilation) of the neural collapse. Nguyen et al. [81] purpose memorization-dilation model and define memorization based on neural collapse under corrupt label training data.

Definition 5 (Corrupt Label Memorization based on Neural Collapse). For a given and labeled dataset D with label noise η and K categories, if f is a feature extractor, denoting the feature representations $f(x_i^k)$ of the example x_i^k by h_i^k. Under neural collapse, any h_i^k will collapse to a single feature representation h^k. We denote the set of corrupted instances of class k by $[\tilde{I}^k]$. Memorization can be defined as

$$\text{mem} := \sum_{k=1}^{K} \sum_{i \in [\tilde{I}^k]} \left\| h_i^k - h_*^k \right\|, \tag{5}$$

where h_*^k denotes the mean of (unseen) test instances belonging to class k.

The DNN under neural collapse intends to map examples with the same ground truth label to a single representation due to the similarity in input features. Therefore, instances of the same ground truth but with randomly corrupted labels lack predictable features, making it challenging for the network to distinguish and separate them in a manner that can be generalized effectively. Thus,

when the network is still able to successfully separate such instances, it indicates that the network has memorized the feature representations of the corrupted instances present in the training set. This definition expresses the memorization of noisy examples but only applies to the problem domain of neural collapse. The scope of the definition is limited.

Evaluation Method - Definition Evaluation. Corrupt label memorization based on neural collapse primarily measures model memorization using noisy datasets under the neural collapse phenomenon. Therefore, this memorization metric relies solely on Definition 5.

2.6 Unintended Memorization

Conceptual Definition. Unintended memorization mainly serves to privacy concerns. The concept was first proposed by Carlini et al. [18] when they find that LLMs may memorize some sensitive information like social-security numbers unintentionally. Generally, such memorization is unnecessary for achieving generalization and they give a simple unintended memorization definition.

Definition 6 (Unintended Memorization). Unintended memorization occurs when trained neural networks may expose the presence of out-of-distribution training data and the training data is irrelevant to the learning task and definitely does not contribute to improving model accuracy.

Compared to the differential memorization definitions, the unintended definition focuses specifically on the memorization of out-of-distribution and sensitive data. These data can also be considered as secrets, as they should not be revealed or disclosed by the trained neural networks.

Evaluation Method - Recurrence Memorization Evaluation. Unintended memorization can be evaluated by the probability that neural networks generate or extract specific marked examples embedded in the training dataset. This evaluation approach can be referred to as recurrence memorization evaluation. Clearly, the selection of marked examples significantly influences the outcome of the memorization assessment.

A typical method [18] employs random sequences to evaluate Unintended Memorization in language models depending on this evaluation method. Specifically, they build the canary sequences which consist of two parts. The first part is like "the random number is" and the second part is just random numbers. Consequently, they create a metric called exposure index based on the log-perplexity,

$$\mathbf{Px}_f(x_1, \cdots, x_n) = \sum_{i=1}^{N}(-\log_2 \mathbf{Pr}(x_i | f(x_1, \ldots, x_{i-1}))),$$

where f is the language model, and $x_1, \cdots, x_n$ represents the input sequence. The perplexity measures how "surprised" the model is to see a given value. A higher perplexity indicates the model is "more surprised" by the sequence. Therefore, the exposure index measures the likelihood of data sequences. The evaluation follows confirming the canary sequence inserted into the training dataset, training, and then applying the exposure index to gain the probability of the canary sequence reproduction. The exposure index of the canary sequence may represent the model memorization ability.

Additionally, employing other types of examples like atypical examples instead of random examples may disclose other properties of model memorization. This requires further studies.

2.7 k-Eidetic Memorization

Conceptual Definition. Carlini et al. [19] introduce the concept of k-Eidetic Memorization for language tasks and employ it to evaluate high-risk examples. The parameter k represents the count of distinct training examples that contain a specific string.

Definition 7 (k-Eidetic Memorization). A string s is k-eidetic memorized (for $k \geq 1$) by an LM f_θ if s appears in at most k examples in the training data $X : |x \in X : s \subseteq x| \leq k$ and the s is extractable

Table 3. Main Memorization Proxies

Proxy	Reference	Level	Description
Holdout Retraining Memorization Evaluation	Carlini et al. (2019) [14]	Example	Measures memorization via logit differences before and after fine-tuning.
Consistency Score Evaluation	Jiang et al. (2021) [50]	Example	Assesses example typicality by excluding it from training and checking accuracy.
Probabilistic Memorization Evaluation	Carlini et al. (2022) [13], Li et al. (2023) [65]	Example	Uses membership inference attacks to estimate whether examples are memorized.
Learning Events Evaluation	Jiang et al. (2021) [50]	Example	Tracks how quickly an example is learned over training epochs.
Loss Curvature Memorization Evaluation	Garg et al. (2023) [34]	Example	Measures memorization via curvature of the loss function around examples.
Cumulative Sample Loss and Gradient Evaluation	Ravikumar et al. (2025) [89]	Example	Quantifies memorization using total loss and gradient norms over time.
Noisy Label Memorization Evaluation	Arpit et al. (2017) [8], Dong et al. (2022) [29], Maini et al. (2023) [75]	Model	Uses memorized noisy labels as a proxy to study memorization dynamics.

from the LM f_θ with a prefix c which satisfies

$$s \leftarrow \arg \max_{s' : |s'|=N} f_\theta(s'|c), \tag{6}$$

where $f_\theta(s'|c)$ is the likelihood of an entire sequence s' with length N.

This memorization definition helps figure out the possibly memorized strings based on repetition times in LM. If k is large, the memorized string may be common knowledge like the zip code of a particular city. But when k is very small, the memorized string could be harmful like accidentally exposing a personal phone number. The k-Eidetic Memorization is also concerned with privacy but utilizes the repetition times as a parameter to identify common knowledge memorization and harmful unintended memorization in language tasks.

Evaluation Method - Extraction Memorization Evaluation. The k-Eidetic Memorization can be evaluated using extraction or inversion approaches, which empirically assess model memorization by generating all extractable examples and identifying those present in the training dataset. This approach is referred to as extraction memorization evaluation. This method attempts to provide a lower bound of model memorization. However, it is noted that not all extracted examples are identical to corresponding training examples because memorization works on the feature scale. Some extractable examples can be regarded as the representatives of generalized examples. It requires good metrics to ensure extractable examples are really memorized.

Specific extraction methods depend on the task type. For example, Carlini et al. [19] applied this approach to extract training data from large language models and identify examples with small k in k-Eidetic Memorization. They generated large volumes of text using GPT-2 and selected those with the highest memorization probability, validating the memorization of the selected text manually via Internet search.

3 Memorization Proxy

Memorization proxies are commonly used tools or methods capable of evaluating memorization in the absence of explicit definitions. Table 3 demonstrates the all memorization proxies.

3.1 Holdout Retraining Memorization Evaluation

Holdout retraining evaluation [14], grounded in the long-tailed theory [30], measures memorization by comparing the differences in example logits between the original model and its fine-tuned model.

Since atypical examples have distinctive features, their logits in the fine-tuned model may vary significantly. The extent of this variation reflects how atypical the examples are and, consequently, how likely they are to be memorized.

Formally, let f denote a model trained on dataset D, and let f' be the fine-tuned model obtained by adapting f on a test dataset D_t. For an example $(x_i, y_i) \in D$, memorization can be quantified as

$$\text{proxy}(f, f', (x_i, y_i)) = \text{KL}(f(x_i) \| f'(x_i)), \tag{7}$$

where KL denotes the symmetric KL-divergence. Intuitively, a higher value of this proxy indicates that the example is more atypical and thus more likely to be memorized.

3.2 Consistency Score Evaluation

Consistency score, or C-score, introduced by Jiang et al. [50], aims at describing how well a held-out example aligns with the underlying data distribution. Specifically, it evaluates the model's performance on an example when that example is excluded from the training dataset. The C-score is formally defined as

$$\text{C-score}(A, D, (x_i, y_i)) = \Pr_{f' \leftarrow A(D^{\setminus i})}[f'(x_i) = y_i], \tag{8}$$

where A denotes the training algorithm, D is the training dataset, and f' represents a model trained on $D^{\setminus i}$, i.e., the dataset with the example (x_i, y_i) removed. A high proxy value means the example is typical.

3.3 Probabilistic Memorization Evaluation

Probabilistic memorization relies on differences in model outputs between memorized and generalized examples, enabling the evaluation of example-level memorization. There may exist multiple techniques to capture the differences but the most relevant method is the membership inference attack.

This kind of attack aims at determining whether a data point belongs to the training dataset. The success of the attack cannot rely on the generalized feature of examples because these features are common for the entire data distribution. Therefore, the membership inference attack focuses on the particular or unique features that models memorize. In other words, data points that the model has memorized during training are more likely to be correctly identified as belonging to the training dataset in membership inference attacks. Though no formal definition, some works [17, 65] tacitly approve the relationship and adopt membership inference attack to measure memorization.

The typical work is ***Likelihood Ratio Attack (LiRA)*** [13]. The core idea behind LiRA is similar to **Definition** 1, which involves evaluating membership inference risks by leveraging likelihood ratios. LiRA aims at assessing whether a given data point is a member of the training dataset by computing the likelihood ratio based on the model's predictions of the data point when the training dataset includes and excludes it. The original formula can be demonstrated as

$$\Lambda(f, (x_i, y_i)) = \frac{p(f|\mathbb{Q}_{in}(x_i, y_i))}{p(f|\mathbb{Q}_{out}(x_i, y_i))}, \tag{9}$$

where $\mathbb{Q}_{in}(x_i, y_i)$ and $\mathbb{Q}_{out}(x_i, y_i)$ represents the distribution of models trained on the training dataset with and without the data point (x_i, y_i) and p is the probability density function over f under the distribution of model parameters $\mathbb{Q}$. The similarity in the core idea highlights the connection between membership inference risks and memorization evaluation.

Moreover, there may exist other techniques based on probability that can be used to estimate memorization. Additionally, it is noted that some membership inference methods have high false positive rates which cannot exactly measure memorization [13, 91]. Therefore, relevant techniques need careful validation and confirmation that they can really reflect the memorization effect.

3.4 Learning Events Evaluation

The learning events proxy [50] is a class of proxies designed to quantify how quickly and reliably a model learns a specific example during training. Generally, neural networks can learn typical examples quickly. This evaluation involves collecting certain metrics at each training epoch and averaging them over time. The metrics used may include confidence, entropy, or other relevant measures. Let M denote the selected metric.

$$\text{proxy}(f, T, (x_i, y_i)) = \frac{1}{T} \sum_{t=0}^{T} M(f^t(x_i), y_i), \tag{10}$$

where T denotes the total number of training epochs, and f^t represents the model at epoch t. A high proxy value indicate that the example is learned quickly and is less likely to be memorized.

3.5 Loss Curvature Memorization Evaluation

Loss curvature proxy [34] can measure example memorization based on the curvature of the loss function around a given data point. This curvature is computed from the derivative of the loss function with respect to the inputs, and the proxy value is obtained by averaging these curvature values over the epochs of training. This can be expressed using

$$\text{curv}(f, (x_i, y_i)) = \frac{1}{T} \sum_{t=0}^{T} \nabla^2_{x_i} \mathcal{L}(f^t(x_i), y_i), \tag{11}$$

where T denotes the total number of training epochs, and f^t represents the model at epoch t, the $\mathcal{L}$ indicates the loss function. High curvature examples correspond to long-tailed, mislabeled, or conflicting instances, indicating a likelihood of memorization.

3.6 Cumulative Sample Loss and Gradient Evaluation

Cumulative sample loss and gradient [88, 89] are used to quantify example memorization based on learning dynamics. Long-tailed examples often require extended periods of training, during which their losses remain high and their gradients relatively large. Therefore, Ravikumar et al. [88, 89] attempt to measure memorization using the **cumulative sample loss** (**CSL**) and **cumulative sample gradient** (**CSG**), defined as follows:

$$\text{CSL}(f, (x_i, y_i)) = \sum_{t=0}^{T} \mathcal{L}(f^t(x_i), y_i), \quad \text{CSG}(f, (x_i, y_i)) = \sum_{t=0}^{T} \left\| \nabla_{x_i} \mathcal{L}(f^t(x_i), y_i) \right\|_2^2, \tag{12}$$

where T denotes the total number of training epochs, and f^t represents the model at epoch t, the $\mathcal{L}$ indicates the loss function.

3.7 Noisy Label Memorization Evaluation

Noisy label memorization evaluation actually is not directly used to measure model memorization but is a valuable method to build memorization baselines compared to other properties of the model. Depending on the fact that noisy label examples have no shared class-level features and patterns, the model has to memorize all of these noisy label examples. Thus, many works utilize noisy label examples as known memorization. Arpit et al. [8] mix the noisy label examples with normal examples to study the learning dynamics during training. They use the ratio of noisy label examples in the training dataset to represent memorization. Another work [29] is studying the memorization effect in adversarial training, utilizing the randomly labeled adversarial examples. Additionally, Maini et al. [75] attempt to employ noisy label examples to localize the memorization in the neural

network. Hence, noisy label memorization evaluation is a common method to investigate the relationships between memorization and other factors.

4 Memorization in DNN Training

A primary motivation behind research on memorization is to explore its impact and what role it plays in DNN training. In this section, we will provide a comprehensive review of memorization research in the DNN training framework.

4.1 Exploring Memorization Mechanism

In some early studies [58], researchers believed that the memorization effect was unnecessary for learning. Generalization in DNNs arises from the recognition of shared features in examples. When DNNs learn these common patterns, they exhibit the ability to generalize, thereby demonstrating their capacity to perform well on new, unseen data beyond the training dataset. In contrast, memorization means networks memorize specific input examples rather than patterns which results in overfitting. However, as Zhang et al. [126] find that DNNs can easily fit the random labeled dataset, the traditional statistical learning theory like VC dimension [113], Rademacher complexity [10], and uniform stability [11, 80, 85] cannot explain the generalization of DNNs. Since no shared patterns exist in randomly labeled examples, DNNs must rely on memorization. Moreover, the memorization mechanism in the DNNs remains unclear and vague. Therefore, this raises two key questions: why DNNs memorize during standard training, and how this memorization operates. This attracts the machine learning community to explore the memorization effect.

In this section, we primarily adopt Definition 1 to investigate the memorization phenomenon. Specifically, we focus on memorization, i.e., the tendency of deep neural networks to memorize specific or particular features of certain examples, rather than learning generalizable patterns. We argue that this behavior constitutes memorization learning, which stands in contrast to pattern learning during model training.

4.2 Memorization and Data Training

Understanding how data distribution shapes memorization tendencies and orders, and how the memorization mechanism influences training performance, is key to exploring the memorization.

The real-world natural data distributions are generally long-tailed [90] and almost all practical datasets are sampled from the real world. Considering this, Feldman et al. [30, 31] propose the long tail theory, suggesting that rare examples as illustrated in Figure 3 are prone to be memorized. Moreover, memorizing these long-tailed examples is crucial for achieving close-to-optimal generalization errors in long-tailed data distributions because rare and atypical instances can provide necessary generalization. They validate this using the memorization score (Definition 1). The results illustrate that atypical examples are more likely to be memorized and that removing them increases generalization errors. This effect also appears in language tasks [131]. Building on this, Jiang et al. [50] develop the **consistency score** (**C-score**) to measure the per-instance generalization, finding that atypical examples have lower C-scores, which provides convincing evidence for the long tail theory. Zhang et al. [127] extend memorization scoring to unsupervised learning via counterfactual memorization (Definition 3), discovering that high-memorization cases are generally unconventional texts such as all-capital letters, structured formats, and multilingual texts. In contrast, low memorization examples are generally templated documents with many near-duplicate copies in the training data. This trend aligns with the long tail theory [31].

Moreover, a recent research [75] indicates that the DNNs cannot empirically identify noisy examples from atypical examples that removing memorization-associated neurons impairs both noisy example classification and model generalization performance. Moreover, memorization paths

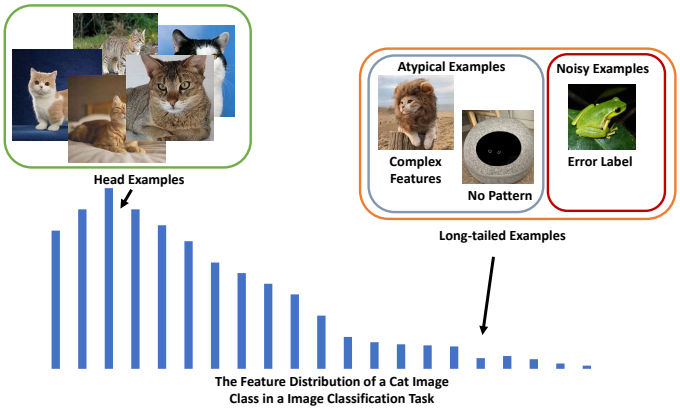


Fig. 3. Demonstration of the Long-tailed Examples.

differ across independent training experiments for the same example [39], suggesting that DNNs may select different particular features to uniquely identify the same example. Furthermore, the learning order of clean examples has observable consistency across similar architectures [42]. However, when training DNNs on the same dataset with randomly shuffled labels, they find that different models memorize data in different orders, indicating multiple possible memorization paths.

Summary

Observations:

— Long-tailed examples are prone to be memorized.
— Long-tailed examples contribute to model generalization performance.
— DNNs cannot identify noisy examples from atypical examples empirically.
— DNNs learn the same noisy dataset in a different order.

Findings: Neural networks memorize atypical long-tailed examples due to their low representativeness, focusing on reducing training loss rather than capturing meaningful features. Moreover, some of these memorized cases contain rare but useful features that can improve generalization.

4.3 Memorization and Data Repetition

Intuitively, DNNs tend to memorize duplicated examples. Zhang et al. [127] believe most memorization criteria strongly correlate with the number of example occurrences in the training, and language models will capture common memorization such as familiar phrases, public knowledge, or templated texts. Moreover, deduplicated datasets reduce the memorization frequency and improve generalization [59]. From the perspective of the extraction task, repeated examples have a high probability of being extracted [16].

One study [19] links repetition times and memorization, proposing k-Eidetic Memorization (Definition 7), where k relates to the number of occurrences for one example. They apply this definition to the language model extraction task and investigate GPT-2. For extractable large k examples, they include common knowledge like city names or high-frequency words, and complex text such as the entire text of the MIT public license because the license may occur thousands of times in the training dataset. However, GPT-2 also memorizes some low-frequent examples with small k like contact information and valid URLs.

Therefore, in practical environments, example repetition is an influence factor of memorization. Nevertheless, the long tail theory [30, 31] tells us that the long-tailed examples are prone to be memorized and these long-tailed examples are low-frequent in the distribution. This requires systematically evaluating memorization factors.

Summary

Observation:

— DNNs tend to prioritize the memorization of repeated data.

Findings: Duplicated examples are easily memorized, but memorization also depends on data representativeness and other factors. Since no unified framework explains these influences, the main factors of memorization are likely those that best reduce loss.

4.4 Memorization and Training Stage

Researchers have discovered that DNN training has a critical early learning stage [3, 33], where performance improves rapidly. Then with the truth that DNNs slowly minimize loss in the final stage of training [82, 126], it is reasonable to believe that pattern learning and memorization learning dominate different training stages. Thus, understanding the dynamics of memorization across training stages is an important research topic.

Due to the difficulty in separating memorization from generalization, one approach is to train on datasets that combine both clean and noisy examples. Arpit et al. [8] utilize the method and find that DNNs tend to prioritize learning patterns even in noisy datasets, as evidenced by high clean validation accuracy in the early training stage. Subsequently, DNNs begin to directly memorize noisy examples, leading to a rapid drop in validation accuracy. Maennel et al. [73] observe that during early training with random labels, network parameters align with data principal components, while misalignment increases in later stages.

Another perspective [39, 101] based on analyzing the gradient variation explains the phenomenon. In the early learning phase, the gradients from noisy examples contribute minimally to the total gradient because inconsistent gradient information may counteract each other, and those shared patterns of the same class are consistent, facilitating quick updates and promoting pattern learning. Similarly, applying a new detection method called "**variance of gradients**" (**VoG**) [4], the examples with lower VoG in the early training stage are more typical compared to the examples in the later training stage. Combining this with the long tail theory [30], it may be inferred that pattern learning dominates the early training stage.

Summary

Observations:

— DNNs tend to prioritize learning patterns from a training dataset containing both clean and noisy examples during the early stage of training.

— In the early learning phase, the gradients from noisy examples contribute minimally to the overall gradient.

Findings: Pattern learning dominates early training, driven by consistent gradients from similar examples, while memorization emerges later as atypical and noisy examples provide conflicting gradients that hinder early learning.

4.5 Memorization and Model Architecture

Different layers in DNNs exhibit distinct learning dynamics. Yosinski et al. [123] show that shallow layers learn general features transferable across tasks, while deeper layers capture task-specific patterns. Using **Singular Vector Canonical Correlation Analysis** (**SVCCA**), Raghu et al. [87] find shallow layers converge early, and Morcos et al. [79] extend this with projection-weighted **Canonical Correlation Analysis** (**CCA**), showing shallow layers consistently learn common patterns, whereas deep layers diverge depending on whether networks generalize or memorize. Ansuini et al. [7] observe that **intrinsic dimensionality** (**ID**) rises across shallow layers but decreases in final layers.

This links deep layer specialization with memorization. Stephenson et al. [101] argue memorization emerges in deeper layers via shrinking manifolds, while Anagnostidis et al. [6] show shallow layers preserve generalized features under random labels, whereas deep layers overfit. Similarly, Maennel et al. [73] explain that the training data will align the principal components of network parameters at the earlier layers when trained with random labels and later layers become specialized.

However, the latest work conducted by Maini et al. [75] reveals that memorization of a classification task exists in a small set of neurons in various layers of the model, and the layers that contribute to example memorization are, not the final layers. In their experiment, they use a noisy dataset to train DNNs. Subsequently, they apply technologies known as layer retraining and layer rewinding to eliminate memorization within individual layers. Finally, they validate the memorization effect (i.e., accuracy of noisy examples in the training dataset) on modified models. The unexpected finding is that the memorization effect still persists in the model, which proves various layers contribute to the memorization.

Additionally, some works explore memorization as a function itself. Chatterjee [20] demonstrates lookup table networks can exhibits generalization. Zhang et al. [125] show different architectures vary between constant-function (memorization) and identity-function (generalization) biases.

As Transformers [114] achieve a big success in various tasks, people are also interested in memorization of Transformers. Sukhbaatar et al. [103] augment the self-attention layers with persistent memory vectors and find this plays a similar role as the feed-forward layer. Geva et al. [36] identify feed-forward layers as key-value memories combining pattern detection and memorization. Dai et al. [24] propose knowledge neurons that encode factual knowledge, with activations aligned to fact expression.

Summary

Observations:

— With multiple networks, shallow layers converge to more similar representations.
— For deep layers, memorization networks converge to different representations.
— Memorization exists in a small set of neurons in various layers of the model.
— The memorization location is different for different model architecture.

Findings: Neural network layers serve different functions: deeper layers specialize, but memorization is not confined to them. Instead, it occurs in a small subset of neurons across layers, though its mechanisms and locations remain unclear, requiring further study.

4.6 Memorization and Overfitting

Overfitting is a common phenomenon in deep learning which represents that a model learns the training data so well that it captures not only the underlying patterns but also the particular features

in the data. This causes the model to fail in generalizing effectively to new, unseen data. Thus, early research commonly held the opinion that overfitting was responsible for memorization. However, contemporary studies [18, 75, 108] provide evidence supporting the persistence of memorization throughout the training process. Memorization does not necessarily lead to overfitting.

Based on the long tail theory [30, 31], we understand that memorizing atypical examples contributes to generalization. In contrast, overfitting will enlarge the generalization error while training loss decreases. Some recent works suggest that even for DNNs without a significant train-test gap, memorization still exists [19, 108]. Additionally, the privacy risks (Evaluation 3.3) also imply the underlying memorization. It is known that overfitting is not necessary for successful membership inference attacks [70, 121]. Furthermore, when a neural network is trained in a training dataset mixed with clean examples and less noisy examples, both the accuracy of noisy examples and that of clean examples exhibit concurrent improvement [75]. Overfitting is a phenomenon of training observed in the later stages of training. For individual training examples, DNNs may memorize them while learning patterns in the early training stage. Thus, memorization does not necessarily require overfitting.

Another interesting phenomenon is benign overfitting. The phenomenon means that even after overlearning training data, DNNs still can generalize well [9, 68]. This theory believes overparameterized DNNs can generalize to the majority of the data distribution using simple paths, and memorize mislabeled and irregular data using complex paths. These components do not interfere, making such overfitting benign. One explanation [12] believes that overfitting becomes benign when the signal-to-noise ratio satisfies a certain condition. In simple terms, benign overfitting requires sufficient signals in the dataset. Thus, benign overfitting may involve less memorization, yet there is insufficient evidence to illustrate their relationship.

Summary

Observations:

— Memorization contributes to the generalization performance but overfitting enlarge the generalization error.

— DNNs are capable of concurrently learning from both clean and noisy examples.

Findings: Overfitting as a training phenomenon does not have a strong relationship with memorization. In the context of overfitting, memorization is necessary but not sufficient.

4.7 Memorization and Data Augmentation, Regularization

Data augmentation and regularization are widespread techniques used in training neural networks. Therefore, it is necessary to study the impact of these practices on memorization.

General Data Augmentation. Data augmentation expands training datasets with artificially generated examples to improve model generalization by enriching semantic representations. This section only discusses the trivial data augmentation, which means fundamental transformations applied to the original training dataset. For instance, in image processing, these transformations could include rotations, flips, zooms, and color variations. In natural language processing, techniques might encompass synonym replacement or back-translation.

In related works, one early study [94] demonstrates trivial data augmentation can reduce the risks of membership inference attacks, thereby diminishing memorization. Utilizing recent memorization evaluation methods, Li et al. [65] study the memorization effect of multiple data augmentation. They measure the memorization evaluation results by membership inference and demonstrate trivial data augmentation technologies significantly mitigate memorization. However, for advanced data

augmentation technologies, further research on the memorization effect is still required. Another work [6] measures memorization based on *k*NN probing and they find that *k*NN probing accuracy of the embedding vectors increases with data augmentation under random label training datasets and clean training datasets. Moreover, they observe that learning under complete label noise with data augmentation still leads to highly useful features in the shallow layers, explaining it as augmented datasets increasing the effective size of the dataset beyond the capacity of networks. This supports that data augmentation mitigates memorization. However, the mechanism of how data augmentation impacts memorization still needs to be explored and systematically evaluated.

Regularization. Regularizers like weight decay and dropout are the standard tools in theory and practice to mitigate overfitting in the training of neural networks. We know that regularizers help constrain the learning process to a specific subset of the hypothesis space with manageable complexity. In the work of Zhang et al. [126], explicit regularizers can prevent model memorization under random label learning, and help the model improve generalization. However, regularization is neither necessary nor by itself sufficient for controlling generalization errors. Then the research conducted by Arpit et al. [8] reproduces a similar result as Zhang et al. [126] and finds dropout is best at hindering memorization without reducing the model's ability to learn. This also responds to the work of location memorization [75], in which finds memorization exists in a small set of neurons in various layers of the model. It seems under random label training, explicit regularizers can mitigate memorization by dropping or constraining neurons, but it is not clear to understand how regularizers influence atypical example memorization in standard training.

Summary

Observations:

— Data augmentation mitigates memorization measured by membership inference evaluation.

— Explicit regularizers can prevent model memorization under random label learning.

Findings: Data augmentation and regularization improve generalization and reduce memorization under random labels, but their effects on long-tailed examples in standard training remain unclear, suggesting a direction for future research.

4.8 Memorization and Other Factors

Capacity. The model capacity is related to model memorization. Generally, models with larger sizes can memorize more data than smaller ones [16]. Additionally, early work has shown that overparameterized neural networks can directly memorize randomly labeled modern datasets [126]. However, we cannot easily think larger capacity leads to more memorization because training data plays an important role. The effective capacity of networks cannot directly explain memorization and generalization [8]. Naturally, a question arises: what happens if the training dataset size far exceeds the model's capacity? Data augmentation can create this condition, and Anagnostidis et al. [6] find that even the randomly labeled dataset with data augmentation exceeding the model capacity can produce effective patterns in the model. Nevertheless, related topics still require further research.

Loss. The loss function is an important component of neural network training. Thus, it must influence the memorization dynamics of models. Patel et al. [83] propose **robust log loss** (**RLL**) which can prevent model overfitting on the randomly labeled data. However, no further studies have explored how different types of loss functions affect model memorization.

Learning Rate. Learning rate is an essential hyperparameter in neural network training. Li et al. [66] believe that a small learning rate model easily learns details, while a large learning rate helps capture patterns. They demonstrate this by adding a small patch to CIFAR10 images that are immediately memorizable by a model with a small initial learning rate but ignored by the model with a large learning rate until after annealing.

Data Format. The data format may affect memorization during training, particularly for language tasks. Kharitonov et al. [55] find the size of the subword vocabulary learned by **Byte-Pair Encoding** (**BPE**) greatly affects both ability and tendency of standard Transformer models to memorize training data. Larger subword vocabulary and shorter input sequences result in strong memorization. The underlying reason could be that complex subwords weaken the patterns in the data distribution. Thus, the input data format likewise impacts memorization.

Summary

Observations:

— DNNs with larger sizes can memorize more data.
— The loss function influences the memorization dynamics of the models.
— A small learning rate model easily learns details.
— Larger subword vocabulary and shorter input sequences result in strong memorization.

Findings: Memorization is influenced by multiple factors: large models with many parameters, smaller learning rates that capture fine details, and choices of loss functions and data formats. However, their effects remain insufficiently understood and need further study.

5 Underlying Risks of Memorization Learning

In previous sections, DNNs have been shown the feature of memorizing training data, and this property may cause various security risks. This section undertakes exploration and synthesis of the impact of memorization on typical threats and defenses in DNNs.

5.1 Memorization and Membership Inference Risks

A membership inference attack is a representative privacy inference attack and seeks to address the query if a specific instance belongs to the training dataset [100]. In the machine learning setting, the membership inference adversary is typically given access to a model's predictions with varying granularity, ranging from the complete confidence vector to the label corresponding to the highest confidence score. It is established that the memorization phenomenon entails the memorization of training data points by DNNs, thereby implying that memorized data bears substantial risks for membership inference.

Indeed, prior work has demonstrated that the membership inference risks associated with training data exhibit significant non-uniformity. According to empirical results, typical data points have lower membership inference risks than those atypical data examples and outliers [13, 31, 48, 60]. These findings imply memorization is highly corresponding or even results in high membership inference risks. However, no direct quantitative investigation has been identified to establish the precise relationship between memorization and membership inference risks. In practice, this relationship has been implicitly approved [13, 17, 60].

Depending on the relationship between memorization and membership inference attack, Carlini et al. [17] find the privacy onion effect. The effect can be defined: when removing the most vulnerable data under a specific privacy attack and retraining a model on only the previously safe data, a

new set of examples in turn becomes vulnerable to the same privacy attack. This phenomenon may indicate that even after removing those memorized data points, the model still memorizes relatively atypical data examples in the remaining training dataset. This observation intuitively underscores membership inference risks are highly associated with memorization and proves that memorization is relative.

Utilizing the memorization effect, researchers investigate more threatening membership inference attacks. Leino et al. [60] attempt to exploit features of memorized examples that are predictive only for the training data but not the sampling distribution. They capture differences in memorization learning data and pattern learning data and build a confident binary logistic classifier to infer membership. Another work is LiRA [13], it depends on the leave-one-out method as memorization score definition and utilizes differences of model outputs that training with and without the training example to do membership inference. These attacks directly exhibit that memorized examples have higher privacy risks than generalized examples.

Summary

Observations:

— Long-tailed examples or atypical examples have higher membership inference risks.
— When removing the most vulnerable examples, a new set of examples in turn becomes vulnerable to the same privacy attack.
— Particular or atypical features can lead to privacy leakage.

Findings: Long-tailed examples behave differently when included or excluded from training, enabling membership inference attacks. This risk also suggests such attacks can be used to evaluate memorization.

5.2 Memorization and Inversion/Extraction Risks

The adversary in an inversion/extraction attack attempts to rebuild or extract training examples by leveraging gradients or models. The attack obviously threatens the privacy of machine learning as the acquired training examples inherently unveil sensitive information. Based on current knowledge, the generalized features embedded in the gradient or model parameters cannot facilitate the precise reconstruction or extraction of training examples because these features are common. Consequently, the underlying reasons for the inversion/extraction risk potentially come from the memorization phenomenon.

Related work mainly analyses the memorization effect concerning the extraction risk of language tasks [16, 18, 19]. Carlini et al. [18] introduce a memorization exposure metric utilizing canary sequences and log-perplexity. Subsequently, they establish that successful extraction becomes feasible when the level of memorization exposure surpasses a threshold. Conversely, extraction remains unsuccessful below this threshold. Consequently, it can be inferred that memorized examples carry a substantial risk of extraction.

Another work [19] demonstrate an extraction attack on GPT-2. They generate a large dataset via unconditional sampling and apply various metrics to identify highly memorized examples. Their findings reveal that extractions mostly yield trivial knowledge and atypical features. This outcome reflects two mechanisms: reconstruction of common knowledge through learned patterns and extraction of atypical, individual examples directly tied to memorized data. From a privacy perspective, the first aligns with the intended role of DNNs, while the second clearly compromises privacy.

Additionally, the measurement outcomes of inversion/extraction risk can be regarded as an empirical lower-bound of memorization [16].

Summary

Observations:

— High memorization examples are easy to be extracted.
— Extracted examples include trivial information and atypical features.

Findings: Inversion and extraction risks are closely tied to memorization: attacks may reconstruct common examples with general patterns or extract rare and specific memorized examples. Thus, memorization underlies privacy risks, and extraction attack can provide a lower-bound estimate of memorization.

5.3 Memorization and Poisoning Risks

Poisoning attacks target breaking model availability. Specifically, adversaries attempt to degrade model performance on all examples (i.e., *untargeted poisoning attack*) or specific classes or examples (i.e., *targeted poisoning attack*), even examples with particular features (i.e., *backdoor attack*). Common poisoning techniques include data manipulation, called *data poisoning*, and model corruption, called *model poisoning*. As *model poisoning* is generally used in distribution machine learning systems, we mainly discuss data poisoning including label manipulation and input noise corruption.

Randomly labeled examples lack shared features for pattern learning, yet DNNs can still minimize loss and reach near-perfect accuracy [126], implying these examples are memorized and that label-based data poisoning exploits memorization. Performance drops correlate with the fraction of mislabeled data [8]. Moreover, removing memorization-associated neurons prevents effective classification of mislabeled samples, though it also harms generalization since atypical and noisy examples are difficult to distinguish [75]. Another poisoning method is input noise injection: as noise grows, inputs shift from typical to atypical to full noise. Since DNNs must minimize loss, this forces stronger memorization of noisy inputs.

Summary

Observation:

— DNNs can learn examples with random labels or added noise.

Findings: Neural networks readily memorize mislabeled or noisy data in poisoning attacks to minimize loss, but the role of memorization in more complex poisoning attacks remains unclear and needs further study.

5.4 Memorization and Adversarial Risks

The adversarial attack employs adversarial noise on inputs to drive examples approaching the decision boundary and achieving the maximum loss. Generally, the adversarial noise is generated by gradient ascending [38, 72]. An effective defense strategy is adversarial training [72] which means directly training on the adversarial examples and this method provides a lower-bound robustness guarantee.

In spite of the absence of relevant studies on memorization and adversarial attacks, we can infer that the memorization effect is not the source of adversarial vulnerability because existing work [47] believes adversarial vulnerability derives from non-robust features.

Several works study memorization in adversarial training [29, 65, 119]. Adversarial examples are atypical [65] and more complex than standard ones [97], leading DNNs to memorize them and increasing vulnerability to privacy attacks during adversarial training. Moreover, memorizing

atypical samples rarely improves robustness. When such samples resemble a wrong class, decision boundaries blur and robustness degrades [119]. This aligns with robust overfitting, where one-hot labels misrepresent adversarial examples, which may require low-confidence assignments [92]. Dong et al. [29] study randomly labeled dataset performance in adversarial training. They further show that while PGD-AT [72] fails to converge on random labels, TRADES [128] attains near 100 % accuracy. They argue DNNs can memorize adversarial data with random labels, but convergence depends on the AT algorithm.

Summary

Observations:

— DNNs tend to memorize adversarial examples during adversarial training.
— Memorized atypical examples may corrupt the adversarial robustness.

Findings: Memorization itself does not cause adversarial vulnerability, but in adversarial training difficult examples are often memorized, raising privacy risks. Excessive memorization of mislabeled features or one-hot labels can break decision boundaries.

5.5 Memorization and Differential Privacy

DP [1] is a commonly employed strategy for defending against privacy attacks, which aims to guarantee indistinguishability between various data points. In particular, DP ensures that the trained model remains largely unchanged when any single example is removed from the training set. Within the framework of (ϵ, δ)-DP setting, DP comprises two key components: gradient clipping and the application of noise. Gradient clipping restricts the gradients of each example to a predefined boundary, reducing disparities in their gradient magnitudes. This facilitates the standardization of gradients in terms of magnitude and mitigates the memorization effect, which has been demonstrated in related works [13, 106]. Additionally, the random noise application also promotes example memorization reduction. When models are trained on examples mixed with random noise, the features carried by long-tailed examples or atypical features will be diluted, leading the models to mainly learn typical patterns. Moreover, neural networks may memorize artificial random noise [19, 30, 31]. Because we always observe that effective DP measures hurt model generalization. Corresponding, from a privacy standpoint, we can understand that DP operates by safeguarding privacy through the prevention of atypical feature memorization. Some works provide empirical results [13, 18, 60] to demonstrate that DP can limit example memorization.

Summary

Observations:

— DP applies random noise and gradient clipping to ensure indistinguishability between various data points.
— DP hurts model generalization performance.

Findings: Differential Privacy (DP) mitigates memorization risks via gradient clipping and noise, but its effectiveness may be achieved by suppressing atypical features, thereby harming generalization.

5.6 Memorization and Other Risk Mitigation Strategies

Data deduplication mitigates unintended memorization in LLMs by removing near-duplicate and exact-duplicate sequences. Repetitions in training data amplify memorization and risk leaking

sensitive information (Section 4.3). Deduplicated models show reduced memorization and are significantly less vulnerable to privacy attacks [51, 59].

Moreover, data filtering techniques can directly remove private or personal information including credit card number or personal mobile that is prone to memorization, thereby reducing associated privacy risks. Although current data preprocessing and personal information detection methods are not yet capable of filtering out all private or sensitive data, they can still eliminate certain types of structured personal information, thus mitigating memorization-related privacy threats [71, 102].

In addition, memorization removal approaches such as machine unlearning [52, 116] may further mitigate the privacy risks associated with memorization. Post-processing techniques like output guardrails [28, 98] can also help prevent the release of memorized sensitive content. Sakarvadia et al. [95] suggest some advanced methods for reducing memorization. In general, mitigating memorization risks remains an active area of research, necessitating continued exploration and development.

Summary

Observations:

— Preprocessing mitigates memorization by removing private data before training.
— Post-processing blocks disclosure of memorized sensitive content.
— Machine unlearning enables targeted deletion of harmful memorized information.

Findings: Memorization risks unexpectedly arise from the data itself. Mitigation can be achieved through deduplication, filtering, and output guardrails, while unlearning techniques offer a direct way to remove memorization from trained models.

6 Memorization Reversing: Forgetting

After exploring memorization in deep neural networks, including definitions, evaluation methods, training behaviors, and risks, we shift focus to its natural counterpart: forgetting. Memorization reflects what models retain, while forgetting indicates what they lose over time. Importantly, forgetting does not arise in isolation. It is tightly intertwined with memorization, since the similar dynamics that lead a model to retain atypical examples also indicate the patterns likely to be forgotten. By examining forgetting in relation to memorization, we gain a more complete understanding of learning dynamics, security risks, and opportunities for unlearning. This section therefore extends the preceding memorization framework to investigate how and why forgetting emerges, and what implications it carries for generalization, and model security.

Forgetting is the opposite of memorization. The typical forgetting phenomenon is catastrophic forgetting [56, 93]. Generally, neural networks may encounter difficulties in continual learning because the learning capacity of networks is not infinite. During iterative training, as networks train on new examples, they tend to forget learned features or information from previous examples, as shown in Figure 4. This phenomenon is known as catastrophic forgetting [56, 93]. Variations in data distributions cause models to converge to different optimal points. Although there are some methods to overcome this phenomenon [44, 56, 67, 69, 93, 99, 99], we still lack an understanding of forgetting especially as an opposite of memorization. We may be curious about what information will be forgotten, how the forgetting effect impacts model performance and privacy, and its relationship with memorization.

6.1 Forgetting Definition and Evaluation

6.1.1 Forgetting Definition based on Accuracy. Toneva et al. [109] propose the forgetting and learning event:

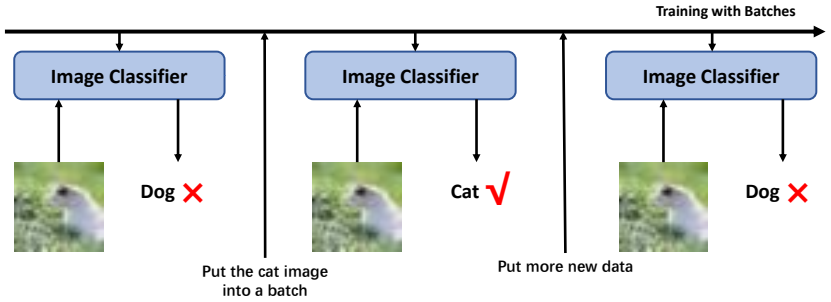


Fig. 4. Demonstration of the Forgetting Phenomenon.

Definition 8 (Forgetting and Learning Event). For supervised classification task, given an example $\mathbf{x}_i$, the predicted label for example $\mathbf{x}_i$ obtained after t steps of SGD is $\hat{y}_i^t = \arg\max_k p(y_{ik}|\mathbf{x}_i;\theta^t)$ and accuracy is $acc_i^t = \mathbb{1}_{\hat{y}_i^t = y_i}$. Therefore, the forgetting event is that example i is misclassified at step $t+1$ after having been correctly classified at step t (i.e., $acc_i^t > acc_i^{t+1}$). Conversely, a learning event has occurred if $acc_i^t < acc_i^{t+1}$.

Following the forgetting and learning event definitions, Maini et al. [74] definite **First-Split Learning Time** (**FSLT**) to demonstrate the first epoch that model learns an example and **Second-Split Forgetting Time** (**SSFT**) to describe the forgetting time in the fine-tuned stage.

Definition 9 (First-Split Learning Time). For $\{\mathbf{x_i}, y_i\} \in D_A$, learning time is defined as the earliest epoch during the training of a classifier f on D_A after which it is always classified correctly, i.e.,

$$FSLT_i = \arg\min_{t^*} \left(\hat{y}_{i,(A)}^t = y_i, \forall t \geq t^* \right) \ \forall \{\mathbf{x_i}, y_i\} \in D_A, \tag{13}$$

where t denotes epoch, A is the pre-training stage and D_A is the training dataset. The f_A represents the trained model with 100% training accuracy on the D_A.

Definition 10 (Second-Split Forgetting Time). Let $\hat{y}_{i,(A\to B)}^t$ to denote the prediction of example $\{\mathbf{x_i}, y_i\} \in D_A$ after training $f_{(A\to B)}$ for t epochs on D_B. Then, for $\{\mathbf{x_i}, y_i\} \in D_A$ forgetting time is defined as the earliest epoch after which it is never classified correctly, i.e.,

$$SSFT_i = \arg\min_{t^*} \left(\hat{y}_{i,(A\to B)}^t \neq y_i, \forall t \geq t^* \right) \ \forall \{\mathbf{x_i}, y_i\} \in D_A, \tag{14}$$

where D_B is a held-out split dataset (without $\{\mathbf{x_i}, y_i\}$) of D_A, $f_{(A\to B)}$ is initialized by f_A.

These definitions can be used to measure forgetting in supervised classification tasks. Based on the nature of forgetting, i.e., learned features have been lost, it is reasonable to obverse forgetting depending on the accuracy.

6.1.2 Forgetting Definition based on Membership Inference Attack. Jagielski et al. [48] measure the ratio of forgetting based on a membership inference attack.

Definition 11 (Rate of Forgetting). A training algorithm $\mathcal{T}$ is said to $(\mathcal{A}, \alpha, k)$-forget a training example z if, k steps after z is last used in $\mathcal{T}$, a privacy attack $\mathcal{A}$ achieves no higher than success rate α.

We know that the membership inference attack relies on particular features, thus, the reduced risk could be regarded as forgetting. This is also an effective definition to describe forgetting.

6.2 Forgetting Phenomenon

Some existing studies provide interesting evidence to understand the forgetting phenomenon. Toneva et al. [109] define example forgetting as when a previously correct example becomes misclassified. They show that generalized examples are unforgettable, while atypical or noisy ones are prone to forgetting, consistent with memorization studies [30, 31]. Despite intermediate forgetting, DNNs eventually memorize all training data, reaching 100% accuracy, suggesting memorization is a forced stage. Moreover, removing some unforgettable examples does not harm generalization, as DNNs need not repeatedly learn common patterns.

Following this research, Maini et al. [74] propose SSFT to track the epoch (if any) after which an original training example is forgotten as the network is fine-tuned on a randomly held-out partition of the data. In the fine-tuned stage, they demonstrate that noisy examples are forgotten quickly and seemingly atypical examples are forgotten slowly, while typical examples are never forgotten. Tirumala et al. [108] employ Definition 2 to measure the single-injected validation dataset forgetting dynamics and find the exact memorization from a higher point gradually drops to the forgetting baseline as the number of epoch increases. The forgetting baseline may represent generalization.

Jagielski et al. [48] focus on the privacy risk associated with forgetting. They attempt to forget a specific example and detect the traces of this example after forgetting. They utilize the membership inference probability to evaluate forgetting and believe that the size of the training dataset, repetitions, and hardness mainly influence forgetting. They find examples used early in model training may be more robust to privacy attacks, and repeated examples are harder to forget. During the forgetting phase, the membership inference probability of typical examples is still around 50% which is lower than the inference risk of atypical examples. This finding indicates that the atypical examples are more vulnerable to privacy attacks and corresponds to previous studies.

Summary

Observations:

- Long-tailed examples or atypical examples are prone to be forgotten.
- During the forgetting phase, the membership inference risk of typical examples is still lower than the inference risk of atypical examples.

Findings: Long-tailed examples lack shared patterns, making them easily forgotten when excluded from training. However, stochastic updates may leave residual traces in parameters, introducing privacy risks.

7 Application

Utilizing the memorization and forgetting effects of neural networks, researchers have developed various applications in several scenarios.

7.1 Application of Long Tail Theory

Example Enhancement. Atypical examples with low frequencies in the data distribution always exist and cause models to memorize them instead of learning patterns based on the long tail theory [30, 31]. Some existing research applies example reweighting technology to enhance of these long-tailed examples. Zhou et al. [132] leverage the memorization effect to improve contrastive learning on long-tailed examples. Furthermore, for adversarial training, Xu et al. [119] observe that memorized atypical examples can compromise model robustness. They apply example reweighting to mitigate the influence of such harmful outliers. Another related work [129] proposes a kNN-based codebook to reweight atypical examples to enhance robustness.

External Memory. Considering the long-tailed examples are memorized with specific features, the system can directly build external memory component to store these memorized examples. Khandelwal et al. [54] use kNN memory to improve rare pattern prediction, extending to translation [53], with Yogatama et al. [122] adding gating/context compression and Févry et al. [32], Verga et al. [115] applying key-value memory for QA. Retrieval-augmented pretraining (REALM [41], MARGE [61], RAG [62]) further boosts tasks, while Wu et al. [117] propose kNN-augmented attention and DSI [105] treats Transformers as search indices.

7.2 Application of Early Learning Mechanism

Noisy Learning. Deep learning with noisy labels is challenging because networks often memorize noise. However, early learning shows that neural networks initially learn patterns. One strategy is utilizing early learning to select clean examples [43, 49, 120, 124], often guided by schedulers. Han et al. [43] propose the example selection scheduler, while Yao et al. [120] enhance it via AutoML search.

7.3 Application of Memorization Location

Model Editing. With memorization location, we may edit memorization to update knowledge. Model editing is such a typical technology. Researchers explore direct neuron editing to update facts [24, 25, 40, 76–78]. Since facts are stored in weights [84], with MLP layers acting as key-value memory [35, 36, 76], modifying neurons becomes practical. Dai et al. [24] identify "knowledge neurons" in BERT via integrated gradients [104]. De Cao et al. [25] and Mitchell et al. [78] use hypernetworks to adjust weights, while Meng et al. [76, 77] propose ROME and MEMIT to locate and edit factual neurons directly. Gupta et al. [40] unify these under a preservation–memorization framework, showing model editing as an effective way to update knowledge.

7.4 Application of Memorization Risks

Privacy Audit. Memorized features pose security risks such as membership inference and example extraction [13, 15, 18, 19, 106]. Therefore, these attacks [13, 19] exploit memorization, but can also be repurposed as audit tools to assess risks and support model compliance.

8 Discussion and Future Research

The memorization effect of DNN is an ongoing field with significant implications for the interpretability, generalization, and security of AI. In this section, we will discuss existing research findings and possible future research directions.

8.1 Memorization and Forgetting Mechanism

The memorization and forgetting mechanism remains unclear and confusing. However, based on existing studies, we have known some memorization and forgetting truths on the classification task:

- Standard training framework always leads DNNs to the minimal loss [82];
- DNNs can memorize common modern training datasets, even when the dataset is randomly labeled [126];
- Long-tailed examples that lack representation in the data distribution like atypical examples and noisy examples are prone to be memorized [30, 31];
- DNNs cannot identify atypical examples and noisy examples in training [75];
- DNNs tend to prioritize the memorization of repeated data [16, 59, 127];
- DNNs have a critical early learning stage where pattern learning takes domination [8];
- Memorization appears to be confined to a limited set of neurons across various layers in DNNs [75];
- Memorization is not responsible for overfitting [70, 75, 121];

— Long-tailed examples are prioritized to be forgotten, and noisy examples are forgotten more quickly than atypical examples [74, 109].

According to these observations, it is reasonable to infer that, at least in the context of classification tasks, the memorization phenomenon is a property of the standard DNN gradient descent training framework. **Specifically, minimizing the loss with gradient descent leads to the memorization effect**.

We will next qualitatively explore the memorization mechanism in classification tasks. Under batch stochastic gradient descent, networks initially build unique mapping paths for each example. Over time, inherent data distribution patterns cause some paths to align, this alignment reflects pattern extraction and relates to early learning [3, 8, 33], as the alignment effect reduces the loss of representative examples. This alignment precedes memorization because early gradients represent the direction that can most effectively reduce the loss. Additionally, long-tailed examples may partially align but often undergo forgetting events [109] (Definition 8), being correctly classified early but misclassified later. Simultaneously, the network may allocate extra capacity to memorize features of these long-tailed examples that are not aligned well in the early learning stage to reduce the loss, evidenced by accuracy gains on both random and clean labels [75]. This also explains why memorization does not depend on overfitting. Moreover, this indicates memorization learning and pattern learning are not totally discrete and contrary. They imply the difficulty of pattern extraction on examples. The harder the pattern extraction, the stronger the memorization tendency. After alignment, networks prioritize memorizing long-tailed examples, consistent with long-tail theory for near-optimal generalization [30, 31]. However, if the training continues, diminishing the training loss becomes challenging. The network may develop unique paths for all examples, causing the predicted vector to closely approach the label vector, even resulting in zero training error. This phenomenon is referred to as neural collapse [82], where each example in the same class collapses to the same representation. For architecture, unique memorization mapping paths may require only a few parameters across layers because these paths are not based on pattern recognition. Therefore, it is reasonable to observe that memorization appears to be confined to a limited set of neurons across various layers in DNNs [75]. Furthermore, if we take into account forgetting, these memorized examples become highly unstable. This instability arises because even if a small part of the associated parameters has been updated, these examples are likely to be misclassified. This phenomenon explains why long-tailed examples are particularly prone to being forgotten [74]. Meanwhile, there may remain some unchanged associated parameters that pose privacy risks [48].

Certainly, our theoretic model of memorization and forgetting in the classification task is an assumption. Further experimentation and empirical evidence are required to fully explain the memorization phenomenon. Understanding the memorization mechanism carries significant implications for enhancing the interpretability of DNNs. To effectively understand this mechanism, it is crucial to describe the spatiotemporal memorization process. In terms of training periods, the primary objectives include characterizing memorization in different stages and investigating whether the memorization phenomenon constitutes a form of overlearning. Concerning neural network components, it becomes essential to quantitatively explain the distribution of memorization across layers or components and evaluate whether certain neurons exhibit a tendency for memorizing examples. Additionally, it is important to differentiate between memorization learning and pattern learning neurons. Moreover, different training frameworks may have distinct memorization phenomena, particularly for unsupervised tasks and multiple-task scenarios.

8.2 Memorization and Forgetting for Training Discussion

Data. DNNs offer significant advantages in processing complex real-world data such as images and text compared to traditional machine learning methods [57, 114]. A notable observation is that

DNNs can effectively extract common features or patterns from the data distribution. However, controlling this extraction process is challenging, and some uncommon yet useful features may not be learned well [30, 31]. At the feature level, out-of-distribution features and rare but useful features are both less representative. This may explain why memorization learning cannot identify atypical examples and noisy examples. Moreover, we may rethink how to describe complex data in reality to keep features balanced. This encourages us to contemplate aspects such as data dimension, granularity [23], and distribution [22, 130] to enhance the performance of DNNs. Additionally, the size of the training dataset probably does not serve as the sole determining factor for task performance [5].

Training Framework. It is understood that the stochastic gradient descent method will aggressively minimize the loss function until reaching extreme mathematical conditions such as neural collapse [82]. However, the extreme conditions may not meet our requirements, and even potentially introduce further challenges. From this perspective, the loss function really matters and decides the learning direction. The challenge lies in the fact that loss functions may not always accurately measure the true loss associated with the assigned task. For instance, in a classification task, a model with minimal loss may have poor generalization performance due to overfitting. Therefore, the memorization and forgetting effect may serve as adaptive solutions to address this conflict.

Architecture. The impact of neural network architecture details on the memorization phenomenon remains unclear. Different layers within the architecture may assume distinct roles, with certain layers potentially exhibiting a preference for memorization. Viewing the architecture in terms of layer depth, deeper layers may tend to learn more specialized features [123] although these features are not completely for memorization. Specialized features still retain patterns, whereas memorized features may lack patterns and serve primarily to mark data. Therefore, deeper layers do not function for memorization [75]. Regarding the size of networks, the memorization tendency also correlates with the size of the training dataset. A larger model trained on a small dataset may lead to significant overfitting and a strong inclination toward memorization. Conversely, larger datasets, which contain more diverse patterns, tend to reduce the preference for memorization but may increase the probability of underfitting.

Tasks. Memorization and forgetting manifest differently across tasks. Most studies examine classification, where memorization involves using a small set of parameters to uniquely mark examples, consistent with its dimension reduction nature. In contrast, generative tasks such as GANs [37], Diffusion models [27], and GPT [86] are dimension-increment tasks aiming at learning full data distributions. Here, generalization refers to producing accurate and coherent outputs, and memorization may involve large parameter sets encoding long-tailed examples, as shown in [15, 19]. Memorization also plays a role in multi-task and continual learning, where catastrophic forgetting [56, 93, 99] highlights the difficulty of retaining knowledge across shifting distributions. In such cases, memorization can help preserve learned features.

8.3 Memorization and Forgetting for Privacy and Security Discussion

Privacy Leakage. Privacy leakage in neural networks often comes from memorization [13, 48]. In membership inference attacks, unique (memorized) features increase vulnerability [13]. Thus, long-tailed examples are most at risk. In inversion/extraction attacks, adversaries may reconstruct representative data, but true risks lies in recovering particular memorized examples. Generative models are especially vulnerable, as they often memorize long-tailed examples to fit distributions, enabling near-lossless reconstruction [15, 19]. Overall, memorization drives privacy risks, and mitigating it can reduce inference and extraction vulnerabilities. Although memorization can assist model generalization, a tradeoff exists between performance and privacy.

Malicious Attacks. Poisoning, backdoor, and adversarial attacks are typical malicious attacks. These attacks can directly disable networks or embed malicious triggers to mislead models. Due to the absence of specific threat evaluation, we only conduct some hypothetical discussions. Suppose the attack just modifies the training data without optimization and targets disabling networks like label flip and random noise. In such cases, the model fails because it cannot learn correct patterns following the gradient direction and has to memorize them. Therefore, the memorization phenomenon is an adaptive process. For induced attacks without optimization [21], these attacks can install malicious triggers in networks, the triggers could be learned via pattern learning or memorization learning. Specifically, this depends on the feature distribution of triggers. If the backdoor feature is long-tailed, here applying memorization, otherwise it is pattern learning. Finally, some malicious attacks are based on optimization, the adversary can submit artificial features or gradients to the networks [72]. The synthetic features or gradients are out of the standard training framework, so it is challenging to discuss the memorization effect under this condition. This requires further studies.

Forgetting Guarantee. From a privacy perspective, forgetting can act as a guarantee by ensuring that previously learned particular features of examples will be forgotten in later stages of training without repeated involvement [108]. Therefore, the input order of examples may impact privacy. Forgetting underlies machine unlearning: typical examples only provide generalized patterns, while removing long-tailed examples allows their unique features to be forgotten, reducing the risk of privacy [48]. However, the onion effect on privacy [17] shows that removing some vulnerable data may increase the risk to others, leaving the quantitative impact of forgetting and unlearning on privacy uncertain.

Risk Mitigation. To mitigate privacy risks associated with memorization, several technologies and strategies can be employed. Data augmentation and regularization are effective in reducing memorization and improving generalization. Augmentation enhances data diversity, particularly benefiting long-tailed examples. Regularization methods, such as weight decay, constrain the feature space to prevent extreme parameters, while dropout randomly deactivates neurons to disrupt memorization. Differential privacy introduces noise to mask memorized features, though this may lead to performance degradation. Additionally, machine unlearning [95, 110] is a promising approach for removing unintended memorization. Since memorization risks can also arise from the data itself, data pre-processing and output post-processing techniques offer further mitigation.

8.4 Application Discussion

Memorization is an inherent property of deep learning, bringing both opportunities and risks. On the positive side, it allows models to retain rare but valuable patterns, which can be leveraged through example enhancement techniques [132], external memory architectures [54], and retrieval-augmented methods [41, 61, 62]. The early learning mechanism, where models first capture general patterns before overfitting noise, also supports tasks like clean example selection and noisy training [43, 49, 120, 124]. Furthermore, studies on memorization localization inform model editing and knowledge modification [13, 15, 18, 19, 106]. On the other hand, memorization can heighten privacy and security risks. However, such risks also enable privacy audits, for example, through membership inference attacks [13]. Additionally, forgetting phenomena suggest new directions for machine unlearning [110], with broader applications still to be explored.

Overall, memorization and forgetting hold significant potential: enhancing robustness and generalization, guiding privacy audits and safeguards, and enabling dynamic model adaptation. Future work should refine these applications with a memorization perspective, maximizing benefits while mitigating risks.

9 Conclusion

This survey explores the memorization effect in DNNs, discussing its definitions in generalization and security/privacy. We discuss how to measure memorization at various levels. Moreover, we consider memorization influence factors like data distribution, training stage, and model structure. Additionally, we review related studies on privacy and security risks linked to memorization and also present the forgetting effect. Finally, we discuss applications related to memorization, possible mechanisms, impacts, and suggests areas for future research.

In this review, we highlight that memorization and forgetting effects are features of DNNs. These effects have deep impacts on the performance, fairness, explainability, accountability, and privacy of DNNs. Therefore, we should develop the ability to control, manage, and utilize the effects, leading to highly usable and trustworthy neural networks.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774

[3] Alessandro Achille, Matteo Rovere, and Stefano Soatto. 2018. Critical learning periods in deep networks. In *Proceedings of the International Conference on Learning Representations*.

[4] Chirag Agarwal, Daniel D'souza, and Sara Hooker. 2022. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10368–10378.

[5] Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. 2021. Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences* 11, 2 (2021), 796. DOI: https://doi.org/10.3390/app11020796

[6] Sotiris Anagnostidis, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. 2022. The curious case of benign memorization. In *Proceedings of the 11th International Conference on Learning Representations*.

[7] Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[8] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 233–242.

[9] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. 2020. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30063–30070. DOI: https://doi.org/10.1073/pnas.1907378117

[10] Peter L. Bartlett and Shahar Mendelson. 2001. Rademacher and gaussian complexities: Risk bounds and structural results. In *Proceedings of the International Conference on Computational Learning Theory*. Springer, 224–240.

[11] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *The Journal of Machine Learning Research* 2, 3 (2002), 499–526.

[12] Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu. 2022. Benign overfitting in two-layer convolutional neural networks. *Advances in Neural Information Processing Systems* 35 (2022), 25237–25250.

[13] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*. 1897–1914. DOI: 10.1109/SP46214.2022.9833649

[14] Nicholas Carlini, Úlfar Erlingsson, and Nicolas Papernot. 2019. Distribution density, tails, and outliers in machine learning: Metrics and applications. arXiv:1910.13427. DOI: 10.48550/arXiv.1910.13427

[15] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Security Symposium*. 5253–5270.

[16] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *Proceedings of the International Conference on Learning Representations*.

[17] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. 2022. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems* 35 (2022), 13263–13276.

[18] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*. 267–284.

[19] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*. 2633–2650.

[20] Satrajit Chatterjee. 2018. Learning and memorization. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 755–763.

[21] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv:1712.05526. DOI: 10.48550/arXiv.1712.05526

[22] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. 2020. Feature space augmentation for long-tailed data. In *Proceedings of the Computer Vision*. Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.), Lecture Notes in Computer Science, Springer International Publishing, Cham, 694–710. DOI: https://doi.org/10.1007/978-3-030-58526-6_41

[23] Yin Cui, Zeqi Gu, Dhruv Mahajan, Laurens van der Maaten, Serge Belongie, and Ser-Nam Lim. 2019. Measuring Dataset Granularity. arXiv:1912.10154 [cs] DOI: https://doi.org/10.48550/arXiv.1912.10154

[24] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. [n. d.]. *Knowledge Neurons in Pretrained Transformers*. arXiv:2104.08696 [cs] DOI: https://doi.org/10.48550/arXiv.2104.08696

[25] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. arXiv:2104.08164. Retrieved from https://arxiv.org/abs/2104.08164

[26] Amandeep Singh Dhanjal and Williamjeet Singh. 2024. A comprehensive survey on automatic speech recognition using neural networks. *Multimedia Tools and Applications* 83, 8 (2024), 23367–23412.

[27] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 8780–8794.

[28] Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Building guardrails for large language models. arXiv:2402.01822. Retrieved from https://arxiv.org/abs/2402.01822

[29] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. 2022. Exploring memorization in adversarial training. In *Proceedings of the International Conference on Learning Representations*.

[30] Vitaly Feldman. 2020. Does learning require memorization? A short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. Association for Computing Machinery, New York, NY, USA, 954–959. DOI: https://doi.org/10.1145/3357713.3384290

[31] Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2881–2891.

[32] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.), Association for Computational Linguistics, Online, 4937–4951. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.400

[33] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. 2019. The early phase of neural network training. In *Proceedings of the International Conference on Learning Representations*.

[34] Isha Garg, Deepak Ravikumar, and Kaushik Roy. 2023. Memorization through the lens of curvature of loss function around samples. arXiv:2307.05831. Retrieved from https://arxiv.org/abs/2307.05831

[35] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. arXiv:2203.14680. Retrieved from https://arxiv.org/abs/2203.14680

[36] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 5484–5495.

[37] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM* 63, 11 (2020), 139–144.

[38] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. [n. d.]. Explaining and harnessing adversarial examples. ([n. d.]). arXiv:1412.6572v3. Retrieved from https://arxiv.org/abs/1412.6572v3

[39] Jindong Gu and Volker Tresp. 2019. Neural network memorization dissection. arXiv:1911.09537. DOI: 10.48550/arXiv.1911.09537

[40] Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024. A unified framework for model editing. arXiv:2403.14236. Retrieved from https://arxiv.org/abs/2403.14236

[41] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 3929–3938.

[42] Guy Hacohen, Leshem Choshen, and Daphna Weinshall. 2020. Let's agree to agree: Neural networks share classification order on real datasets. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 3950–3960.

[43] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 8536–8546.

[44] Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. 2020. REMIND your neural network to prevent catastrophic forgetting. In *Proceedings of the Computer Vision*. Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.), Lecture Notes in Computer Science, Springer International Publishing, Cham, 466–483. DOI: https://doi.org/10.1007/978-3-030-58598-3_28

[45] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17853–17862.

[46] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17853–17862.

[47] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Mądry. 2019. Adversarial examples are not bugs, they are features. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 125–136.

[48] Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, and Chiyuan Zhang. 2022. Measuring forgetting of memorized training examples. In *Proceedings of the 11th International Conference on Learning Representations*.

[49] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2304–2313.

[50] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. 2021. Characterizing structural regularities of labeled data in overparameterized models. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 5034–5044.

[51] Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 10697–10707.

[52] Aly M Kassem, Omar Mahmoud, and Sherif Saad. [n. d.]. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

[53] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *Proceedings of the International Conference on Learning Representations*.

[54] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *Proceedings of the International Conference on Learning Representations*.

[55] Eugene Kharitonov, Marco Baroni, and Dieuwke Hupkes. 2021. How BPE Affects Memorization in Transformers. arXiv:2110.02782. DOI: 10.48550/arXiv.2110.02782

[56] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114, 13 (2017), 3521–3526. DOI: https://doi.org/10.1073/pnas.1611835114

[57] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[58] David Krueger*, Nicolas Ballas*, Stanislaw Jastrzebski*, Devansh Arpit*, Maxinder S. Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. 2017. Deep nets don't learn via memorization. In *Proceedings of the International Conference on Learning Representations*.

[59] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.), Association for Computational Linguistics, Dublin, Ireland, 8424–8445. DOI: https://doi.org/10.18653/v1/2022.acl-long.577

[60] Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated {white-box} membership inference. In *Proceedings of the 29th USENIX Security Symposium*. 1605–1622.

[61] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 18470–18481.

[62] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 9459–9474.

[63] Mingjie Li, Bingqian Lin, Zicong Chen, Haokun Lin, Xiaodan Liang, and Xiaojun Chang. 2023. Dynamic graph enhanced contrastive learning for chest x-ray report generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3334–3343.

[64] Qiongxiu Li, Xiaoyu Luo, Yiyi Chen, and Johannes Bjerva. 2025. Trustworthy machine learning via memorization and the granular long-tail: A survey on interactions, tradeoffs, and beyond. arXiv:2503.07501. Retrieved from https://arxiv.org/abs/2503.07501

[65] Xiao Li, Qiongxiu Li, Zhanhao Hu, and Xiaolin Hu. 2023. On the Privacy Effect of Data Enhancement via the Lens of Memorization. *IEEE Transactions on Information Forensics and Security* 19 (2024), 4686–4699.

[66] Yuanzhi Li, Colin Wei, and Tengyu Ma. 2019. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[67] Zhizhong Li and Derek Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2018), 2935–2947. DOI: https://doi.org/10.1109/TPAMI.2017.2773081

[68] Zhu Li, Zhi-Hua Zhou, and Arthur Gretton. 2021. Towards an Understanding of Benign Overfitting in Neural Networks. arXiv:2106.03212 [cs, stat] DOI: https://doi.org/10.48550/arXiv.2106.03212

[69] Huihui Liu, Yiding Yang, and Xinchao Wang. 2021. Overcoming catastrophic forgetting in graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 10 (2021), 8653–8661. DOI: https://doi.org/10.1609/aaai.v35i10.17049

[70] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. 2018. Understanding Membership Inferences on Well-Generalized Learning Models. arXiv:1802.04889. DOI: 10.48550/arXiv.1802.04889

[71] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy*. IEEE, 346–363.

[72] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. [n. d.]. Towards deep learning models resistant to adversarial attacks. ([n. d.]). arXiv:1706.06083v4. Retrieved from https://arxiv.org/abs/1706.06083v4

[73] Hartmut Maennel, Ibrahim M. Alabdulmohsin, Ilya O. Tolstikhin, Robert Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. 2020. What do neural networks learn when trained with random labels?. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc., 19693–19704.

[74] Pratyush Maini, Saurabh Garg, Zachary Lipton, and J. Zico Kolter. 2022. Characterizing datapoints via second-split forgetting. *Advances in Neural Information Processing Systems* 35 (2022), 30044–30057.

[75] Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary C. Lipton, J. Zico Kolter, and Chiyuan Zhang. 2023. Can Neural Network Memorization Be Localized? arXiv:2307.09542 [cs] DOI: https://doi.org/10.48550/arXiv.2307.09542

[76] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.

[77] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. arXiv:2210.07229. Retrieved from https://arxiv.org/abs/2210.07229

[78] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. arXiv:2110.11309. Retrieved from https://arxiv.org/abs/2110.11309

[79] Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[80] Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. 2006. Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics* 25, 1–3 (2006), 161–193. DOI: 10.1007/s10444-004-7634-z

[81] Duc Anh Nguyen, Ron Levie, Julian Lienen, Eyke Hüllermeier, and Gitta Kutyniok. 2023. Memorization-dilation: modeling neural collapse under noise. In *Proceedings of the International Conference on Learning Representations*.

[82] Vardan Papyan, XY Han, and David L Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences* 117, 40 (2020), 24652–24663.

[83] Deep Patel and P. S. Sastry. 2021. Memorization in deep neural networks: Does the loss function matter?. In *Proceedings of the Advances in Knowledge Discovery and Data Mining*. Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R. K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, and Tanmoy Chakraborty (Eds.), Lecture Notes in Computer Science, Springer International Publishing, Cham, 131–142. DOI: https://doi.org/10.1007/978-3-030-75765-6_11

[84] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? arXiv:1909.01066. Retrieved from https://arxiv.org/abs/1909.01066

[85] Tomaso Poggio, Ryan Rifkin, Sayan Mukherjee, and Partha Niyogi. 2004. General conditions for predictivity in learning theory. *Nature* 428, 6981 (2004), 419–422.

[86] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. OpenAI technical report/preprint. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[87] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[88] Deepak Ravikumar, Efstathia Soufleri, Abolfazl Hashemi, and Kaushik Roy. 2024. Memorization and the orders of loss: A learning dynamics perspective. Manuscript submitted for review. https://openreview.net/forum?id=lTh7DEJV5WUnderdouble-blindreview

[89] Deepak Ravikumar, Efstathia Soufleri, Abolfazl Hashemi, and Kaushik Roy. 2025. Towards memorization estimation: Fast, formal and free. In *Proceedings of the 42nd International Conference on Machine Learning*.

[90] William J. Reed. 2001. The pareto, zipf and other power laws. *Economics Letters* 74, 1 (2001), 15–19. DOI: https://doi.org/10.1016/S0165-1765(01)00524-9

[91] Shahbaz Rezaei and Xin Liu. 2021. On the difficulty of membership inference attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7892–7900.

[92] Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 8093–8104.

[93] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting. In *Proceedings of the Advances in Neural Information Processing Systems*. 31 (2018).

[94] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. 2018. D\'ej\'a Vu: An Empirical Evaluation of the Memorization Properties of ConvNets. arXiv:1809.06396 [cs] DOI: https://doi.org/10.48550/arXiv.1809.06396

[95] Mansi Sakarvadia, Aswathy Ajith, Arham Mushtaq Khan, Nathaniel C Hudson, Caleb Geniesse, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W Mahoney. [n. d.]. Mitigating memorization in language models. In *Proceedings of the 13th International Conference on Learning Representations*.

[96] Ali Satvaty, Suzan Verberne, and Fatih Turkmen. 2024. Undesirable memorization in large language models: A survey. arXiv:2410.02650. Retrieved from https://arxiv.org/abs/2410.02650

[97] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[98] Md Shamsujjoha, Qinghua Lu, Dehai Zhao, and Liming Zhu. 2025. Swiss cheese model for ai safety: A taxonomy and reference architecture for multi-layered guardrails of foundation model based agents. In *Proceedings of the 2025 IEEE 22nd International Conference on Software Architecture*. IEEE, 37–48.

[99] Chenze Shao and Yang Feng. 2022. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.), Association for Computational Linguistics, Dublin, Ireland, 2023–2036. DOI: https://doi.org/10.18653/v1/2022.acl-long.143

[100] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. [n. d.]. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy* (2017). 3–18. DOI: https://doi.org/10.1109/SP.2017.41

[101] Cory Stephenson, Abhinav Ganesh, Yue Hui, Hanlin Tang, SueYeon Chung, et al. 2021. On the geometry of generalization and memorization in deep neural networks. In *Proceedings of the International Conference on Learning Representations*.

[102] Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. 2023. Detecting personal information in training corpora: An analysis. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing*. 208–220.

[103] Sainbayar Sukhbaatar, Édouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. arXiv:1907.01470. DOI: 10.48550/arXiv.1907.01470

[104] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3319–3328.

[105] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems* 35 (2022), 21831–21843.

[106] Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. 2020. Understanding unintended memorization in federated learning. arXiv:2006.07490. DOI: 10.48550/arXiv.2006.07490

[107] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*. 303–314.

[108] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems* 35 (2022), 38274–38290.

[109] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *Proceedings of the International Conference on Learning Representations*.

[110] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, Georgios Kaissis, Daniel Rueckert, Gintare Karolina Dziugaite, and Eleni Triantafillou. 2024. Improved localized machine unlearning through the lens of memorization. arXiv:2412.02432. Retrieved from https://arxiv.org/abs/2412.02432

[111] Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. 2024. Towards conversational diagnostic ai. arXiv:2401.05654. Retrieved from https://arxiv.org/abs/2401.05654

[112] Dmitrii Usynin, Moritz Knolle, and Georgios Kaissis. 2024. Memorisation in machine learning: A survey of results. *Transactions on Machine Learning Research* (2024), 1–25.

[113] Vladimir Naumovich Vapnik. 1998. Adaptive and learning systems for signal processing communications, and control. *Statistical Learning Theory* (1998). John Wiley & Sons, New York.

[114] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[115] Pat Verga, Haitian Sun, Livio Baldini Soares, and William W Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. arXiv:2007.00849. DOI: 10.48550/arXiv.2007.00849

[116] Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. 2024. Rkld: Reverse kl-divergence-based knowledge distillation for unlearning personal information in large language models. arXiv:2406.01983. Retrieved from https://arxiv.org/abs/2406.01983

[117] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. In *Proceedings of the International Conference on Learning Representations*.

[118] Alexander Xiong, Xuandong Zhao, Aneesh Pappu, and Dawn Song. 2025. The landscape of memorization in LLMs: Mechanisms, measurement, and mitigation. arXiv:2507.05578. Retrieved from https://arxiv.org/abs/2507.05578

[119] Han Xu, Xiaorui Liu, Wentao Wang, Zitao Liu, Anil K. Jain, and Jiliang Tang. 2023. How does the memorization of neural networks impact adversarial robust models?. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, 2801–2812. DOI: https://doi.org/10.1145/3580305.3599381

[120] Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. 2020. Searching to exploit memorization effect in learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 10789–10798.

[121] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium*. 268–282. DOI: https://doi.org/10.1109/CSF.2018.00027

[122] Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics* 9 (2021), 362–373. DOI: https://doi.org/10.1162/tacl_a_00371

[123] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.

[124] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), PMLR, 7164–7173. Retrieved from https://proceedings.mlr.press/v97/yu19b.html

[125] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C. Mozer, and Yoram Singer. 2020. Identity crisis: Memorization and generalization under extreme overparameterization. In *Proceedings of the International Conference on Learning Representations*.

[126] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *Proceedings of the International Conference on Learning Representations*.

[127] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2021. Counterfactual Memorization in Neural Language Models. arXiv:2112.12938 [cs] DOI: https://doi.org/10.48550/arXiv.2112.12938

[128] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning*. 7472–7482.

[129] Jianfu Zhang, Yan Hong, and Qibin Zhao. 2023. Memorization weights for instance reweighting in adversarial training. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 9 (2023), 11228–11236. DOI: https://doi.org/10.1609/aaai.v37i9.26329

[130] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. 2021. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2361–2370.

[131] Xiaosen Zheng and Jing Jiang. 2022. An empirical study of memorization in NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 6265–6278.

[132] Zhihan Zhou, Jiangchao Yao, Yan-Feng Wang, Bo Han, and Ya Zhang. 2022. Contrastive learning with boosted memorization. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 27367–27377.

Received 14 June 2024; revised 2 September 2025; accepted 15 September 2025