DISSECTING CATASTROPHIC FORGETTING FOR UNDERSTANDING CLASS-INCREMENTAL LEARNING MECHANISM

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper explains catastrophic forgetting in class incremental learning (CIL) from a novel perspective of interactions (non-linear relationship) between different input variables. Specifically, we make the first attempt to explicitly identify and quantify which interactions *w.r.t.* previous classes that are forgotten and preserved over incremental steps, and reveal their distinct behaviors, so as to provide a more fine-grained explanation of catastrophic forgetting. Based on the forgotten interactions, we reveal a shared mechanism for the effectiveness of some classical CIL methods in mitigating catastrophic forgetting, *i.e.,* these methods all reduce the forgetting of interactions *w.r.t.* previous classes, particularly those of low complexities, although these methods are originally designed based on different intuitions and observations. Intrigued by this, we further propose a simple-yet-efficient method with theoretical guarantees to investigate and verify that low-complexity interaction serves as an effective factor in resisting catastrophic forgetting. *The code will be released if the paper is accepted.*

1 INTRODUCTION

Deep neural networks (DNNs) have been widely used in class incremental learning to solve a common real-world problem of learning new classes continually. However, they often suffer from catastrophic forgetting, *i.e.,* directly training the DNN to learn new classes will erase the knowledge of previous classes and result in a decline in performance. Thus, previous research mainly focused on proposing various CIL methods to mitigate catastrophic forgetting, such as regularization-based methods (Li & Hoiem, 2017; Zhao et al., 2020), memory-based methods (Zhou et al., 2023b; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), expansion-based methods (Yan et al., 2021; Wang et al., 2022; Zheng et al., 2025), *etc*.

However, less attention has been devoted to explaining catastrophic forgetting, and existing studies typically employed accuracy-based metrics for explanation. For example, Chaudhry et al. (2018); Wang et al. (2024) computed the difference between the highest test accuracy on a previous task and the final test accuracy to measure the forgetting of this task. Lopez-Paz & Ranzato (2017) calculated the average accuracy drop of all previous tasks to evaluate the DNN's forgetting in the continual learning process. Whereas, these accuracy-based metrics can only provide **a coarse-grained explanation for the outcome** of the catastrophic forgetting, *i.e.,* the decline in the classification accuracy, but they fail to provide **a fine-grained understanding of its underlying cause**, *i.e.,* what knowledge is forgotten or preserved and to what extent during CIL.

Thus, unlike previous studies, this paper aims to provide a detailed explanation for catastrophic forgetting from a new perspective, by defining and extracting the knowledge encoded in an incrementally learned DNN, quantifying its forgetting of old knowledge over incremental steps, and summarizing its distinct property.

However, how to explicitly mathematically define the knowledge[1] encoded in the DNN still remains a challenge. To this end, Ren et al. (2024a; 2023a) have derived a set of theorems to justify the convincingness of considering interactions to represent the knowledge encoded by the DNN for

[1]Please see Appendix A for related works on extracting knowledge of the DNN.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

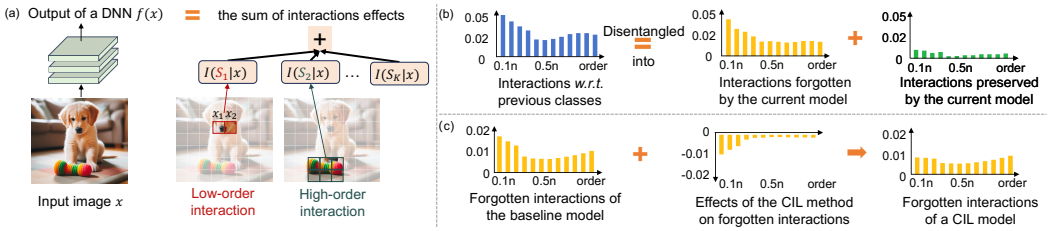


Figure 1: Overview of this paper. (a) The network output of a certain input sample is proven to be the sum of interaction effects. (b) We identify and quantify forgotten and preserved interactions over incremental steps. (c) We use forgotten interactions to explain the intrinsic mechanism of different CIL methods in mitigating catastrophic forgetting, *i.e.,* reducing the forgetting of interactions of previous classes, particularly low-order interactions.

inference. Specifically, given a sample $\boldsymbol{x}$, a DNN usually does not employ each single input variable of $\boldsymbol{x}$ independently for prediction. Instead, the DNN lets each input variable interact with each other to form a certain pattern for inference. For a better understanding, let us consider the toy example in Fig. 1. The DNN encodes the interaction between two image patches in $S = \{x_1, x_2\}$ to form a *dog nose* pattern. Each interaction represents an AND relationship between variables in S. That is, only when all two variables in S are all present, the *dog nose* interaction is activated, and make a numerical effect $I(S|\boldsymbol{x})$ on the network output. The absence/masking[3]of any patch in S will deactivate this *dog nose* interaction, and remove its numerical effect $I(S|\boldsymbol{x})$.

More crucially, it is proven that a well-trained DNN usually encodes a few salient interactions, and its output $v(\boldsymbol{x})$ can be well explained as the sum of numerical effects of these interactions Ren et al. (2024a), $v(\boldsymbol{x}) = \sum_S I(S|\boldsymbol{x})$, as shown in Fig. 1(a), which serves as mathematical evidence to **faithfully take countable interactions between different input variables to represent uncountable knowledge encoded by the DNN**.

Surprisingly, we find interactions can explain different aspects of catastrophic forgetting, which may help both practitioners and theoreticians gain some new insights into the learning dynamics of an incrementally trained DNN. Specifically, this paper aims to answer three questions.

(1) How to identify what is forgotten in class incremental learning. To address this, given a current step t and a previous step $k \in \{1, 2, \cdots, t-1\}$, we propose metrics to disentangle the interaction of each complexity encoded by the previous DNN f_k (incrementally trained from steps 1 to k) into interaction components forgotten and preserved by the current DNN f_t. As shown in Fig. 1 (b), a large number of interactions *w.r.t.* previous classes are forgotten, particularly those of low complexity, while only a small number are preserved, ultimately resulting in catastrophic forgetting. This quantitative finding provides **more direct and concrete evidence** to support the widely accepted qualitative claim that the erasure of a large amount of knowledge *w.r.t.* previous classes leads to catastrophic forgetting (Zhou et al., 2024a), compared to previous accuracy-based metrics. Notably, the complexity (or *order*) of an interaction is defined as the number of variables involved in this interaction S. As Fig. 1 shows, a low-order interaction usually represents the simple AND relationship between a few variables.

(2) How to explain the effectiveness of different CIL methods in mitigating catastrophic forgetting in a single theoretic system. Based on the forgotten interactions, we discover a clear difference between the DNN trained with a CIL method to mitigate catastrophic forgetting (abbreviated as the *CIL model*) and the DNN trained solely with the cross-entropy loss, without any anti-forgetting method (abbreviated as the *baseline model*). As shown in Fig. 1 (c), Fig. 3, and Fig. 4, different CIL methods exhibit a shared mechanism that they consistently make the CIL model reduce the forgetting of interactions *w.r.t.* previous classes, particularly the forgetting of low-order interactions, to resist catastrophic forgetting, although they are originally designed based on different observations and intuitions. This provides a common interaction-based explanation for the effectiveness of different anti-forgetting CIL methods.

(3) What role low-order interactions play in the resistance of catastrophic forgetting. To investigate this, we propose a simple-yet-efficient method to penalize an incrementally learned DNN from encoding low-order interactions, and compare its stability to that of a DNN encoding low-order interactions, where the stability refers to the DNN's ability to resist catastrophic forgetting (Wang

et al., 2024). In experiments, we discover that suppressing low-order interactions can significantly weaken the stability of the DNN. Thus, low-order interactions may, to some extent, serve as an effective factor for resisting catastrophic forgetting.

Contributions of this paper are summarized as follows. (1) We disentangle and quantify the forgotten interactions of different complexities in the incremental learning process, so as to provide a detailed explanation for catastrophic forgetting. (2) We provide a common interaction-based explanation for different CIL methods in mitigating catastrophic forgetting. (3) We explain the role of low-order interactions in resisting catastrophic forgetting.

2 PRELIMINARIES: USING INTERACTIONS TO REPRESENT KNOWLEDGE

Although there is no consensus on the definition of knowledge encoded by a DNN so far, Ren et al. (2024a; 2023a) have proven two theorems to mathematically justify why we can take interactions to represent knowledge in a DNN.

Definition of interactions. Given a trained DNN f, let $\boldsymbol{x} \in \mathbb{R}^n$ denote an input sample consisting of n input variables totally, and $N = \{1, 2, \cdots, n\}$ denote the indices of all variables. The input variable can be defined differently depending on the task, such as an image patch for image classification or a word/token for text classification. Let $f(\boldsymbol{x}) \in \mathbb{R}^2$ represent the scalar network output or a certain output dimension of the DNN. Then, Ren et al. (2023a) employed the Harsanyi dividend (Harsanyi, 1963), a typical metric in game theory, to quantify the numerical effect of the interaction between variables inside a subset $S \subseteq N$ on the network output f.

$$I(S|\boldsymbol{x}) = \sum\nolimits_{U \subseteq S} (-1)^{|S|-|U|} \cdot f(\boldsymbol{x}_U), \tag{1}$$

where $\boldsymbol{x}_U \in \mathbb{R}^n$ denotes a masked sample crafted by masking[3] input variables in $N \setminus U$ and keeping variables in U unchanged. Thus, $I(S|\boldsymbol{x})$ measures the **additional effect contributed exclusively** to the DNN by the interaction between input variables in the set S, subtracting the additional effects $I(U|\boldsymbol{x})$ that are already created by subsets $U \subseteq S$. Please refer to Harsanyi (1963) and Appendix B for detailed explanation of Eq. (1).

Understanding of interactions. Each interaction represents an AND relationship between variables in the subset S. For instance, let us consider the interaction inside the subset $S = \{x_1, x_2\}$ shown in Fig. 1, which forms a semantic pattern of *dog nose*. Only when all two input variables in S are all present, the nose interaction is triggered, and contribute a numerical effect $I(S|\boldsymbol{x})$ on the network output. Otherwise, the masking of any variable will break the nose interaction, and remove the numerical effect $I(S|\boldsymbol{x})$.

Faithfulness of the interaction-based explanation. The following properties serve as convincing evidence that interactions can faithfully represent knowledge encoded by a DNN, rather than a mathematical formulation without clear meanings, since the inference logic of the DNN can be well explained by interactions.

(1) *Sparsity property.* Given a sample $\boldsymbol{x} \in \mathbb{R}^n$, a DNN theoretically can encode at most 2^n different interactions[4] *w.r.t.* 2^n different subsets $S \subseteq N$. However, Ren et al. (2024a) have proven that a well-trained DNN for classification usually encode very sparse salient interactions, which has also been widely observed on various DNNs for different tasks (Ren et al., 2023a;b; Zhou et al., 2024b; Cheng et al., 2024; Ren et al., 2025). We also observe this on DNNs trained for CIL (cf. Appendix H).

Theorem 1 (Sparsity property, proven in (Ren et al., 2024a)). *Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, let $\Omega_{salient} = \{S|S \subseteq N, |I(S|\boldsymbol{x})| \geq \tau\}$ denote a set of salient interactions whose absolute value exceeds a*

[2]People can apply different settings for $f(\boldsymbol{x})$, such as setting $f(\boldsymbol{x}) = \log\big(p(y = y^{\text{truth}}|\boldsymbol{x})/(1 - p(y = y^{\text{truth}}|\boldsymbol{x}))\big)$ as the confidence of predicting $\boldsymbol{x}$ to the ground-truth category y^{truth} in multi-category classification tasks (Ren et al., 2024a; Li & Zhang, 2023; Ren et al., 2023a).

[3]In practice of masking input variables in $N \setminus U$, people commonly use baseline values $\{b_i\}$ to replace their original values (Ancona et al., 2019; Covert et al., 2020), *i.e.,* setting $x_i = b_i$ if $i \in N \setminus U$.

[4]We use the method in (Li & Zhang, 2023) to accelerate the computation of interactions of each input sample, which takes only a few seconds with a single NVIDIA 4090 GPU. Please see Appendix I for computation details and specific time costs.

Table 1: Main notations used in Section 3.1.

Notation	Description
t	**Current step**
k	**Previous step**, $k \in \{1, \cdots, t-1\}$
T	The total number of incremental steps
$\mathcal{D}_k$	The dataset for step k
f_k, f_t	DNNs trained incrementally on $\{\mathcal{D}_i\}_{i=1}^{k}$ and $\{\mathcal{D}_i\}_{i=1}^{t}$
$I^m(S\|\boldsymbol{x}_k, f_k)$	The m-order interaction of the sample $\boldsymbol{x}_k \in \mathcal{D}_k$ encoded by f_k
$\mathcal{I}^m(S\|\boldsymbol{x}_k, f_k)$	The strength of $I^m(S\|\boldsymbol{x}_k, f_k)$, $\mathcal{I}^m(S\|\boldsymbol{x}_k, f_k) \triangleq \|I^m(S\|\boldsymbol{x}_k, f_k)\|$
$\mathcal{I}^{(k,t),m}_{\text{preserve}}(S\|\boldsymbol{x}_k)$	The m-order interaction encoded in f_k and later preserved by f_t
$\mathcal{I}^{(k,t),m}_{\text{forget}}(S\|\boldsymbol{x}_k)$	The m-order interaction encoded in f_k but later forgotten by f_t
$\mathcal{I}^{m}_{\text{preserve}}$	Preserved m-order interactions averaged over all T steps
$\mathcal{I}^{m}_{\text{forget}}$	Forgotten m-order interactions averaged over all T steps

threshold τ. If the DNN can generate relatively smooth inference outputs $f(\boldsymbol{x}_T)$ on masked samples[5]*, then the size of the set $|\Omega_{salient}|$ is proven to have an upper bound of $\mathcal{O}(n^{\kappa}/\tau)$, where κ is an intrinsic parameter for the smoothness of the network function $f(\cdot)$. Empirically, κ is usually within the range of $[1.9, 2.2]$ (Ren et al., 2024b), thus $|\Omega_{salient}|$ is much less than 2^n.*

(2) *Universal matching property.* Theorem 2 proves that we can use a few salient interactions in Ω_{salient} to universally match network outputs $f(\boldsymbol{x}_S)$ on all 2^n masked samples.

Theorem 2 (Universal matching property, proven in (Ren et al., 2023a))**.** *The network output $f(\boldsymbol{x}_S)$ on each masked sample $\{\boldsymbol{x}_S|\forall S \subseteq N\}$ is proven to be well mimicked by the sum of interaction effects.*

$$f(\boldsymbol{x}_S) = \sum\nolimits_{U\subseteq S} I(U|\boldsymbol{x}) \approx \sum\nolimits_{U\subseteq S \& U\in\Omega_{salient}} I(U|\boldsymbol{x}). \tag{2}$$

Complexity of the interaction. The complexity (or *order*) of an interaction is defined as the number of variables involved in this interaction, order $= |S|$ (Ren et al., 2023a). As Fig. 1 shows, a low-order interaction denotes simple AND relationship between a few variables, while a high-order interaction refers to complex AND relationship between numerous variables.

3 Explaining Catastrophic Forgetting Via Interactions

Let us first revisit the class incremental learning, which aims to train a DNN on data that arrives incrementally with new classes (Rebuffi et al., 2017). Specifically, given a sequence of T steps $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_T\}$ without overlapping classes, the t-th step $\mathcal{D}_t = \{(\boldsymbol{x}_{j,t}, y_{j,t})\}_{j=1}^{N_t}$ contains N_t tuples of input samples $\boldsymbol{x}_{j,t} \in \mathbb{R}^n$ and its corresponding label $y_{j,t}$. Then, class incremental learning is to optimize the DNN to minimize the average loss of all classes, where the data from previous training steps is not available or restricted during the current training step (Zhou et al., 2024a).

Owing to the guaranteed faithfulness of the interaction, we use interactions to (i) quantify what knowledge is forgotten and preserved throughout the CIL process in Section 3.1, (ii) explain the effectiveness of CIL methods in Section 3.2, and (iii) reveal the crucial role of low-order interactions in mitigating catastrophic forgetting in Section 3.3.

3.1 Quantifying Forgotten Interactions in CIL

In this section, we quantitatively identify which interactions used for the inference of previous classes are forgotten over incremental steps and ultimately lead to catastrophic forgetting, while accuracy-based metrics (Chaudhry et al., 2018; Wang et al., 2024) fail to reveal what is forgotten and to what extent, limiting deeper insights into catastrophic forgetting. *For ease of understanding, notations used hereafter are summarized in Table 1.*

To this end, given a current step t and a previous step $k \in \{1, \cdots, t-1\}$, we consider that the DNN f_k, trained incrementally on $\{\mathcal{D}_i\}_{i=1}^{k}$, encodes richest interactions relevant to the inference of $\mathcal{D}_k$. However, these interactions are gradually forgotten over incremental steps, with some preserved and utilized by subsequent DNNs $\{f_t\}$ for classification. Explicitly speaking, let us use

[5]This is formulated by three common mathematical conditions. Please see Appendix H for details.

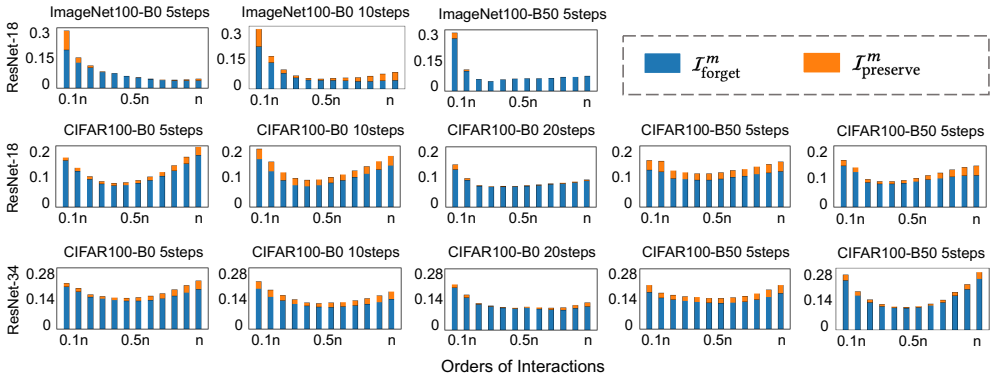


Figure 2: The forgotten ($\mathcal{I}^m_{\text{forget}}$) and preserved ($\mathcal{I}^m_{\text{preserve}}$) interactions *w.r.t.* previous classes.

$\mathcal{I}^m(S|\boldsymbol{x}_k, f_k) \triangleq |I^m(S|\boldsymbol{x}_k, f_k)|$ to quantify the strength of the m-order interaction $I^m(S|\boldsymbol{x}_k, f_k)$ of the sample $\boldsymbol{x}_k \in \mathcal{D}_k$ encoded by the DNN f_k. We propose the following two metrics to disentangle $\mathcal{I}^m(S|\boldsymbol{x}_k, f_k)$ into interaction components preserved and forgotten by the current DNN f_t, where $\mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k)$ quantifies the preserved interaction shared by both f_t and f_k, and $\mathcal{I}^{(k,t),m}_{\text{forget}}(S|\boldsymbol{x}_k)$ measures the interaction encoded in f_k but later forgotten by f_t.

$$\begin{aligned}\mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k)&=\Gamma^t_k(S|\boldsymbol{x}_k)\cdot\min(\mathcal{I}^m(S|\boldsymbol{x}_k, f_k), \mathcal{I}^m(S|\boldsymbol{x}_k, f_t)),\\ \mathcal{I}^{(k,t),m}_{\text{forget}}(S|\boldsymbol{x}_k)&=\mathcal{I}^m(S|\boldsymbol{x}_k, f_k) - \mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k),\end{aligned} \tag{3}$$

where $\Gamma^t_k(S|\boldsymbol{x}_k) = \mathbb{1}((I^m(S|\boldsymbol{x}_k, f_k) \cdot I^m(S|\boldsymbol{x}_k, f_t)) > 0)$ measures whether the interaction $I^m(S|\boldsymbol{x}_k, f_t)$ has the same effect as the interaction $I^m(S|\boldsymbol{x}_k, f_k)$, and $\mathbb{1}(\cdot)$ is the indicator function. If $I^m(S|\boldsymbol{x}_k, f_t)$ and $I^m(S|\boldsymbol{x}_k, f_k)$ have opposite effects, then the preserved interaction $\mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k) = 0$, indicating the current DNN f_t completely forgets the m-order interaction learned at the step k. Otherwise, the preserved interaction is quantified as the shared component $\mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k) = \min(\mathcal{I}^m(S|\boldsymbol{x}_k, f_k), \mathcal{I}^m(S|\boldsymbol{x}_k, f_t))$. $\mathcal{I}^m(S|\boldsymbol{x}_k, f_t)$ denotes the strength of m-order interaction extracted from the input sample $\boldsymbol{x}_k \in \mathcal{D}_k$ encoded by the DNN f_t, which is calculated based on Eq. (1), and $|S| = m$.

Hence, based on Eq. (3), we can identify which interactions of each complexity *w.r.t.* previous steps k are forgotten or preserved during the entire T-step class-incremental learning process, as well as quantifying their exact amounts.

$$\begin{aligned}\mathcal{I}^m_{\text{forget}}&=\frac{1}{T-2}\sum_{t=2}^{T}\left(\frac{1}{t-1}\sum_{k=1}^{t-1}\mathbb{E}_{\boldsymbol{x}_k\in\mathcal{D}_k}\mathbb{E}_{\substack{S\subseteq N,\\|S|=m}}\left[\mathcal{I}^{(k,t),m}_{\text{forget}}(S|\boldsymbol{x}_k)\right]\right)\\ \mathcal{I}^m_{\text{preserve}}&=\frac{1}{T-2}\sum_{t=2}^{T}\left(\frac{1}{t-1}\sum_{k=1}^{t-1}\mathbb{E}_{\boldsymbol{x}_k\in\mathcal{D}_k}\mathbb{E}_{\substack{S\subseteq N,\\|S|=m}}\left[\mathcal{I}^{(k,t),m}_{\text{preserve}}(S|\boldsymbol{x}_k)\right]\right)\end{aligned} \tag{4}$$

Experiments. We conducted experiments to explain how incrementally trained DNNs forgot and preserved interactions *w.r.t.* previous classes. Notably, we focused on the conventional CIL setting, where catastrophic forgetting was typically more pronounced. To this end, we followed settings in (Rebuffi et al., 2017; Hou et al., 2019) to incrementally train ResNet-18 and ResNet-34 models (He et al., 2016) on the CIFAR-100 dataset (Krizhevsky et al., 2009), and ResNet-18 model on the ImageNet-100 dataset (Rebuffi et al., 2017) under different class splits. Specifically, we employed two widely-used class splits to train DNNs. First, we followed (Rebuffi et al., 2017) to equally divide all classes of the CIFAR-100 dataset into 5, 10, and 20 incremental steps (denoted as ***CIFAR100-B0***), and all classes of the ImageNet-100 dataset into 5 and 10 incremental steps (denoted as ***ImageNet100-B0***). Second, we followed (Hou et al., 2019) to allocate half of the total classes to the first step, and further equally divide the rest classes of the CIFAR-100 dataset into 5 and 10 incremental steps, and divide the remaining classes of the ImageNet-100 dataset into 5 incremental steps. These settings were referred to as ***CIFAR100-B50*** and ***ImageNet100-B50***, respectively. Please see Appendix I for more experimental details.

Fig. 2 reports the forgotten $\mathcal{I}^m_{\text{forget}}$ (blue bars) and preserved $\mathcal{I}^m_{\text{preserved}}$ (orange bars) interactions of different orders m. We discovered that for each incrementally learned DNN, $\mathcal{I}^m_{\text{forget}}$ consistently

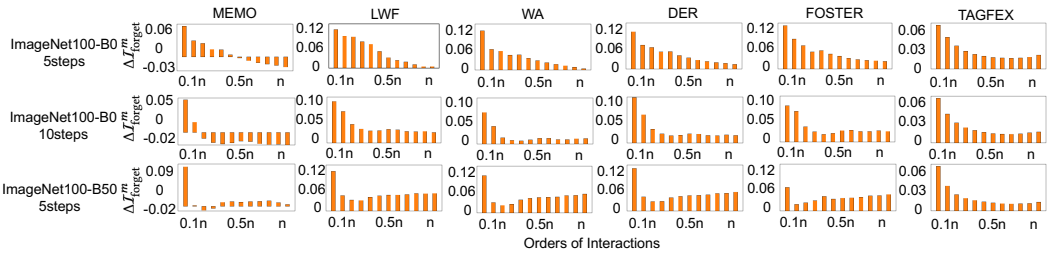


Figure 3: The difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ *w.r.t.* previous classes between the CIL model and the baseline model, both incrementally trained on the ImageNet-100 dataset.

exceeded $\mathcal{I}^m_{\text{preserve}}$. Explicitly speaking, the strength of forgotten interactions was on average approximately 6 times that of preserved interactions, indicating that the DNN suffered severe forgetting of a large number of interactions *w.r.t.* previous classes in CIL. This phenomenon echoed and provided concrete quantitative support for the widely accepted qualitative claim that the erasure of a large amount of knowledge *w.r.t.* previous classes resulted in catastrophic forgetting (Zhou et al., 2024a).

Besides, we also observed from Fig. 2 that the incrementally learned DNN was prone to forgetting relatively more low-order interactions *w.r.t.* previous classes. In comparison, we found that different DNNs exhibited relatively different tendencies in preserving interactions.

3.2 A Common Interaction-based Explanation for CIL Methods

In this section, we use forgotten interactions to reveal a shared mechanism underlying the effectiveness of different CIL methods in mitigating catastrophic forgetting, *i.e.,* whether and how differently designed CIL methods help the DNN resist the forgetting of such interactions.

To this end, let us consider two DNNs f_{base} and f_{CIL}, sharing the same architecture and both incrementally trained on $\{\mathcal{D}_i\}_{i=1}^T$. Among them, f_{CIL} is trained with a certain CIL method (*e.g.,* TagFex (Zheng et al., 2025)) to resist catastrophic forgetting, which is abbreviated as the ***CIL model***. In comparison, f_{base} is trained merely with the cross-entropy loss, without any anti-forgetting methods, which is abbreviated as the ***baseline model***. Then, we compare the difference between m-order interactions *w.r.t.* previous classes forgotten by the CIL model $\mathcal{I}_{\text{forget,CIL}}$ with those forgotten by the baseline model $\mathcal{I}_{\text{forget,base}}$, to explain the effectiveness of the CIL model in mitigating catastrophic forgetting.

$$\Delta\mathcal{I}^m_{\text{forget}} = \mathcal{I}^m_{\text{forget,base}} - \mathcal{I}^m_{\text{forget, CIL}}, \tag{5}$$

where $\mathcal{I}_{\text{forget,CIL}}$ and $\mathcal{I}_{\text{forget,base}}$ are computed based on Eq. (4) with $f_k = f_{k,\text{base}}, f_t = f_{t,\text{base}}$ for the baseline model and $f_k = f_{k,\text{CIL}}, f_t = f_{t,\text{CIL}}$ for the CIL model, respectively. A positive value of $\Delta\mathcal{I}^m_{\text{forget}}$ indicates the CIL model forgets fewer m-order interactions *w.r.t.* previous classes than the baseline model, and vice versa.

Experiments. To explain the effectiveness of CIL methods, we trained ResNet-18 and ResNet-34 models on CIFAR-100 dataset, and learned ResNet-18 model on ImageNet-100 dataset under different class splits in Section 3.1 for class incremental learning. For each DNN, we trained 8 versions based on PyCIL (Zhou et al., 2023a), including a baseline model learned merely with cross-entropy loss, and **7** CIL models trained with different classical and open-sourced CIL methods from different categories to ensure the generality of our explanation, *i.e.,* LWF (Li & Hoiem, 2017) and WA (Zhao et al., 2020) for regularization-based, MEMO (Zhou et al., 2023b) for memory-based, DER (Yan et al., 2021), FOSTER (Wang et al., 2022), and TagFex (Zheng et al., 2025) for expansion-based, and DS-AL (Zhuang et al., 2024) for analytic-learning-based.

Fig. 3 and Fig. 4 report the difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ between the CIL model and the baseline model. We discovered that the CIL model mitigated the forgetting of interactions $\mathcal{I}^m_{\text{forget, CIL}}$ *w.r.t.* previous classes compared to the baseline model, *i.e.,* $\mathbb{E}_m[\Delta\mathcal{I}^m_{\text{forget}}] > 0$. Interestingly, such a phenomenon was shared by different CIL models, although they were not originally designed for this purpose, but based on different intuitions and observations. Moreover, this shared phenomenon was also observed in **additional CIL settings**, including audio-visual CIL with autoencoders (Pian et al., 2023) and CLIP-based CIL (cf. Appendix D for details), providing further evidence for its generality. Thus, this shared phenomenon provided a common understanding for

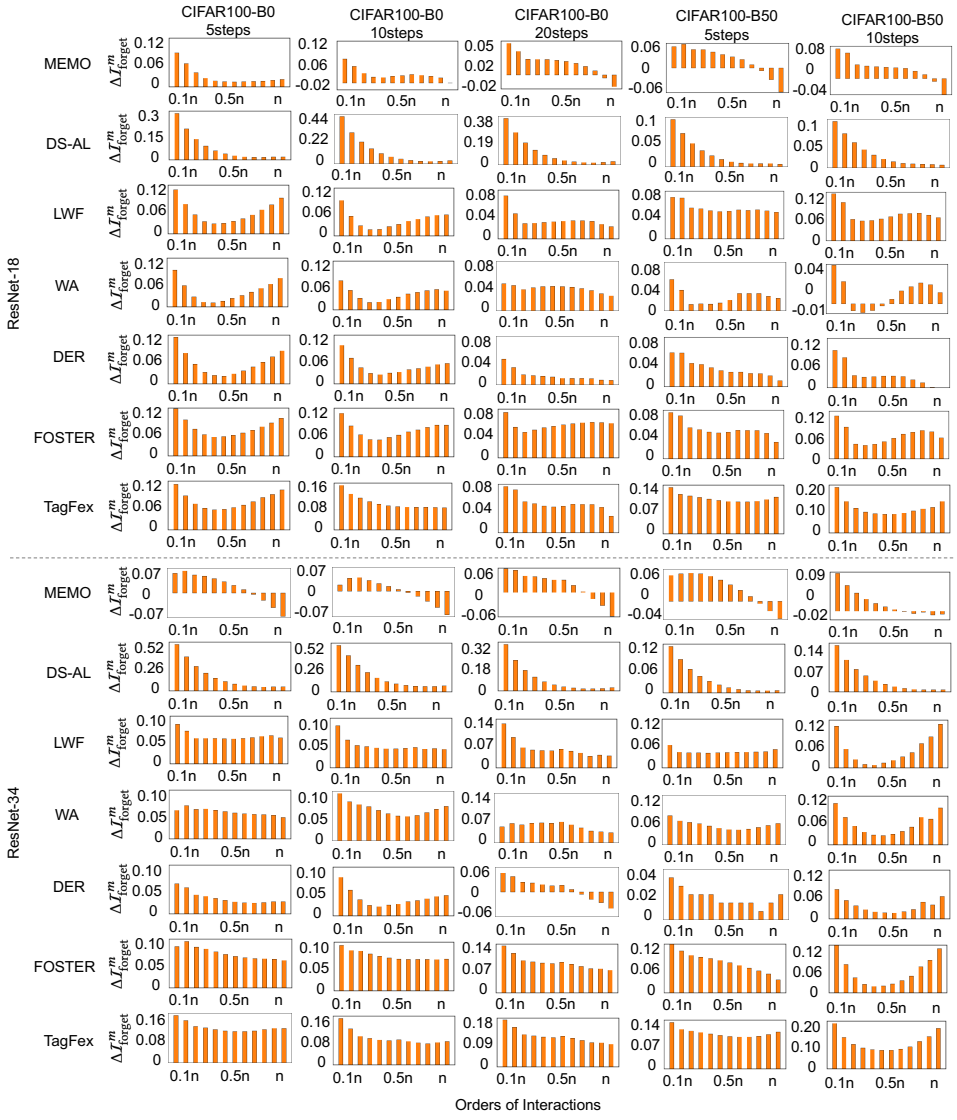


Figure 4: The difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ *w.r.t.* previous classes between the CIL model and the baseline model, both incrementally trained on the CIFAR-100 dataset.

the effectiveness of different CIL methods, *i.e.,* making the incrementally learned DNN forget less interactions *w.r.t.* previous classes to mitigate catastrophic forgetting, which also partially echoed the conclusion in Section 3.1 that the forgetting of old interactions $\mathcal{I}^m_{\text{forget}}$ could result in catastrophic forgetting.

Besides, we also observed another phenomenon that these CIL methods all made the DNN significantly mitigate the forgetting of low-order interactions, *i.e.,* low-order $\Delta\mathcal{I}^m_{\text{forget}}$ was consistently positive, and generally larger than that of higher orders. This indicated CIL models usually forgot significantly fewer low-order interactions than the baseline model, which could be intuitively understood as follows. Low-order interactions usually represented local and common features with considerable generalization power (Zhou et al., 2024b), and high-order interactions (usually global features) were usually constructed by low-order interactions. Thus, low-order interactions learned in previous steps could not only contribute to the inference of previous data, but also generalize and combine with low-order interactions newly learned in the current step to form high-order interactions for current data inference.

Notably, the CIL model trained with MEMO often forgot more high-order interactions *w.r.t.* previous classes than the baseline model, *i.e.,* $\Delta\mathcal{I}^m_{\text{forget}} < 0$ for high order m. This might be because MEMO

made the DNN learn more new high-order interactions *w.r.t.* current classes, as shown in Fig. 8 of Appendix E, where we proposed a new metric to quantify the learning of new interactions at each step. Please refer to Appendix E for detailed discussion and results.

3.3 Exploring the Role of Low-Order Interactions in Resisting Forgetting

In this section, we further explore the role of low-order interactions in CIL. To this end, we propose a simple-yet-efficient method to investigate whether low-order interactions have a potential influence on the resistance of catastrophic forgetting, *i.e.,* checking whether the stability of an incrementally trained DNN significantly decreases when we prevent it from encoding low-order interactions, where the stability measures its ability to resist catastrophic forgetting (Wang et al., 2024).

Specifically, we design a loss function to force the DNN to encode interactions of specific orders. According to the *universal matching property* in Theorem 2, the network output f can be decomposed into the sum of interaction effects of varying orders, Thus, let us first focus on the network output change $\Delta f(m_1, m_2)$ between different masked samples $\{\boldsymbol{x}_{S_1}\}$ and $\{\boldsymbol{x}_{S_2}\}$, which serves as the foundation for designing the loss function.

$$\Delta f'(m_1, m_2) = \mathbb{E}_{\substack{S_1, S_2: \emptyset \subseteq S_1 \subsetneq S_2 \subseteq N, \\ |S_1|=m_1 n,\, |S_2|=m_2 n}} [f(\boldsymbol{x}_{S_2}) - f'(\boldsymbol{x}_{S_1})], \tag{6}$$

where subsets S_1 and S_2 are randomly sampled from all input variables N, containing $m_1 n$ and $m_2 n$ variables, respectively, such that $\emptyset \subseteq S_1 \subsetneq S_2 \subseteq N$, and $0 \le m_1 \le m_2 \le 1$. Then, we have proven that the output change $\Delta f(m_1, m_2)$ mainly encodes interactions of $[0, m_2 n]$ orders, as follows.

Theorem 3 (proven in Appendix C)**.** *The change of the network output $\Delta f(m_1, m_2)$ is proven to be decomposed into interaction effects of different orders, where $\binom{m_2 n}{m}$ and $\binom{m_1 n}{m}$ denote the combinatorial number.*

$$\begin{aligned} \Delta f(m_1, m_2) &= \sum\nolimits_{m=0}^{n} w^{(m)} \cdot \mathbb{E}_{S \subseteq N, |S|=m}[I(S|\boldsymbol{x})], \\ w^{(m)} &= \begin{cases} \binom{m_2 n}{m} - \binom{m_1 n}{m}, & m \le m_1 n, \\ \binom{m_2 n}{m}, & m_1 n < m \le m_2 n, \\ 0, & m_2 n < m \le n. \end{cases} \end{aligned} \tag{7}$$

In this way, based on Theorem 3, we propose the following loss function to prevent the DNN from using $[0, m_2 n]$-order interactions encoded in $\Delta f(m_1, m_2)$ for inference, thereby penalizing the encoding of these interactions. Explicitly speaking, **we maximize the entropy of classification $L_{\text{inter}}(m_1, m_2)$ based on $\Delta f(m_1, m_2)$, in order to make $\Delta f(m_1, m_2)$ non-discriminative**.

$$L_{\text{inter}}(m_1, m_2) = -\mathbb{E}_{\boldsymbol{x}}[\sum\nolimits_{c=1}^{C} [p(\hat{y} = c|\Delta f_c(m_1, m_2, \boldsymbol{x})) \cdot \log(p(\hat{y} = c|\Delta f_c(m_1, m_2, \boldsymbol{x})))]], \tag{8}$$

where $\Delta f_c(m_1, m_2, \boldsymbol{x}) = f_c(\boldsymbol{x}_{S_2}) - f_c(\boldsymbol{x}_{S_1})$ represents the change of the logits *w.r.t.* the category c. C and $\hat{y}$ are referred to as the total number of classes and the predicted label, respectively. We compute the probability $p(\hat{y} = c|\Delta f_c(m_1, m_2, \boldsymbol{x}))$ of classifying the input sample $\boldsymbol{x}$ to a certain category c by feeding the vector $[\Delta f_1(m_1, m_2, \boldsymbol{x}), \Delta f_2(m_1, m_2, \boldsymbol{x}), \cdots, \Delta f_C(m_1, m_2, \boldsymbol{x})]$ into the softmax layer.

Thus, we can train a DNN to force it to use interaction of specific orders for classification, by minimizing the following loss function $L(m_1, m_2)$, where $L_{\text{classification}}$ is employed as the cross-entropy loss in practice, and the positive constant α is used to balance two loss terms.

$$L(m_1, m_2) = L_{\text{classification}} - \alpha \cdot L_{\text{inter}}(m_1, m_2). \tag{9}$$

Experiment 1: investigating effects of the loss $L_{\text{inter}}(m_1, m_2)$. Before explaining the role of m-order interactions in mitigating catastrophic forgetting, we first examine whether $L(m_1, m_2)$ could prevent the incrementally learned DNN from encoding interactions of specific orders. To this end, we trained three versions of each DNN, including a normally trained DNN without any interaction penalization by setting $\alpha = 0$ (*i.e.,* the baseline model), and two DNNs trained with $\alpha = 1.0$ to penalize interactions of specific orders, setting $m_1 = 0, m_2 = 0.3$ and $m_1 = 0.7, m_2 = 1.0$ in $L(m_1, m_2)$, respectively. This training process is summarized in Algorithm 1 of Appendix I.

Fig. 5 reports the average interaction strength $\mathcal{I}^m$ of each DNN, $\mathcal{I}^m = \mathbb{E}_{\boldsymbol{x}} \mathbb{E}_{S \subseteq N, |S|=m}[|I^m(S|\boldsymbol{x})|]$. We discovered that $L_{\text{inter}}(m_1, m_2)$ could successfully prevent DNNs from encoding $[m_1 n, m_2 n]$-order interactions, instead of $[0, m_2 n]$-orders. That is, when discouraging the DNN to encode interactions $[0, 0.3n]$ orders (or $[0.7n, n]$ orders), the interaction strengths of $[0, 0.3n]$ orders (or $[0.7n, n]$ orders)

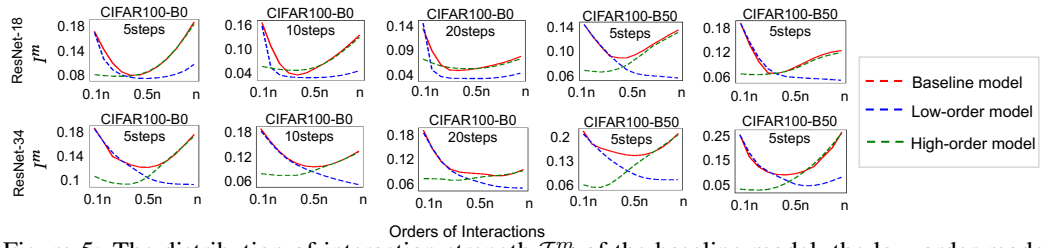


Figure 5: The distribution of interaction strength $\mathcal{I}^m$ of the baseline model, the low-order model, and the high-order model.

Table 2: The stability difference ΔFM_{low} between the low-order model and the baseline model, as well as ΔFM_{high} between the high-order model and the baseline model.

Model	Metric	CIFAR100-B0 5 steps	CIFAR100-B0 10 steps	CIFAR100-B0 20 steps	CIFAR100-B50 5 steps	CIFAR100-B50 10 steps
ResNet-18	ΔFM_{low}	0.08	0.08	0.06	0.02	0.16
	ΔFM_{high}	0.22	0.21	0.24	0.23	0.31
ResNet-34	ΔFM_{low}	0.10	0.05	0.08	0.14	0.08
	ΔFM_{high}	0.22	0.30	0.28	0.30	0.34

were significantly decreased, compared to the baseline model. Thus, we named the DNN trained to penalize $[0, 0.3n]$-order interactions ***high-order model*** for simplicity, as it mainly encoded high-order interactions. Accordingly, the DNN trained to penalize $[0.7n, n]$-order interactions was termed the ***low-order model***.

Experiment 2: exploring the role of low-order interactions in mitigating catastrophic forgetting. To this end, we compared the stability of the baseline model with that of DNNs trained to penalize interactions of specific orders. Specifically, we used the Forgetting Measure metric $FM = \frac{1}{T-1}\sum_{k=1}^{T-1}[\max_{i\in\{1,\cdots,T-1\}}(Acc^{(k,i)} - Acc^{(k,T)})]$ (Chaudhry et al., 2018; Wang et al., 2024) to evaluate the stability of an incrementally learned DNN, which quantified the average decline in the performance on each previous step k. Here, $Acc^{(k,i)}$ denoted the classification accuracy evaluated on the data $\mathcal{D}_k$ by the DNN f_i incrementally trained over i steps. A large FM value implied low stability of the DNN, *i.e.,* low capacity in resisting catastrophic forgetting.

Table 2 illustrated the stability difference $\Delta FM_{\text{low}} = FM(f_{\text{low}}) - FM(f_{\text{base}})$ between the low-order model f_{low} and the baseline model f_{base}, as well as $\Delta FM_{\text{high}} = FM(f_{\text{high}}) - FM(f_{\text{base}})$ between the high-order model f_{high} and the baseline model f_{base}. We discovered that ΔFM_{high} was consistently larger than ΔFM_{low}, which indicated that compared to the baseline model, the stability of the high-order model was much worse than that of the low-order model. Thus, preventing the DNN from encoding low-order interactions could significantly harm its stability in resisting catastrophic forgetting, which suggested low-order interactions might, to some extent, be an effective factor for mitigating catastrophic forgetting. Besides, Table 4 in Appendix G also shows a similar phenomenon based on different CIL models, *i.e.,* penalizing the learning of low-order interactions in the CIL model could significantly reduce its stability.

Additionally, we conducted a preliminary attempt in Appendix F by encouraging the DNN to rely more on low-order interactions, which showed improved resistance to catastrophic forgetting, providing complementary empirical evidence for the role of low-order interactions in mitigating catastrophic forgetting.

4 Conclusion and Discussion

In this paper, we make the first attempt to use interactions to detailedly explain catastrophic forgetting in class incremental learning, by quantifying the forgotten and preserved interactions of different complexities *w.r.t.* previous classes, providing a common interaction-based explanation for the effectiveness of different CIL methods, and proposing a loss with theoretical guarantees to investigate the role of low-order interactions in resisting catastrophic forgetting.

Our work provides avenues for future exploration, such as applying our explanation to analyze other CIL problems (*e.g.*, the stability-plasticity trade-off), which is theoretically feasible. We hope our explanation can serve as a theoretical foundation and shed new light on CIL.

AI USE STATEMENT

In this work, we use generative AI tools exclusively for language polishing. We have not used generative AI tools for information retrieval, discovery, or research ideation.

ETHICS AND REPRODUCIBILITY STATEMENT

This paper identifies and quantifies what is forgotten in class incremental learning, and explains the effectiveness of different CIL methods, from the perspective of interactions. There are no ethical issues with this paper.

Besides, we have provided proofs for the theoretical results in Appendix C, and provided experimental details in Section 3.1, Section 3.2, Section 3.3, and Appendix I. The code will be released when the paper is accepted.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Xu Cheng, Lei Cheng, Zhaoran Peng, Yang Xu, Tian Han, and Quanshi Zhang. Layerwise change of knowledge in neural networks. In *ICML*, 2024.

Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223, 2020.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang. Unifying fourteen post-hoc attribution methods with taylor interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):4625–4640, 2024. doi: 10.1109/TPAMI.2024.3358410.

John C Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 831–839, 2019.

Linlan Huang, Xusheng Cao, Haori Lu, Yifan Meng, Fei Yang, and Xialei Liu. Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3777–3786, 2025.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Artemy Kolchinsky, Brendan D. Tracey, and Steven Van Kuyk. Caveats for information bottleneck in deterministic scenarios. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rke4HiAcY7.

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

Boqi Li, Youjun Wang, and Weiwei Liu. Towards understanding catastrophic forgetting in two-layer convolutional neural networks. In *Forty-second International Conference on Machine Learning*, 2025.

Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? In *International conference on machine learning*, pp. 20452–20469. PMLR, 2023.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7816–7825, 2023.

Dongrui Liu, Huiqi Deng, Xu Cheng, Qihan Ren, Kangrui Wang, and Quanshi Zhang. Towards the difficulty for a deep neural network to learn concepts of different complexities. *Advances in Neural Information Processing Systems*, 36:41283–41304, 2023.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Weiguo Pian, Shentong Mo, Yunhui Guo, and Yapeng Tian. Audio-visual class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7799–7811, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, et al. Towards a unified game-theoretic view of adversarial perturbations and robustness. *Advances in Neural Information Processing Systems*, 34:3797–3810, 2021.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20280–20289, 2023a.

Jie Ren, Xinhao Zheng, Jiyu Liu, Andrew Lizarraga, Ying Nian Wu, Liang Lin, and Quanshi Zhang. Monitoring primitive interactions during the training of dnns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20183–20191, 2025.

Qihan Ren, Huiqi Deng, Yunuo Chen, Siyu Lou, and Quanshi Zhang. Bayesian neural networks avoid encoding complex and perturbation-sensitive concepts. In *International Conference on Machine Learning*, pp. 28889–28913, 2023b.

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where we have arrived in proving the emergence of sparse interaction primitives in dnns. In *The Twelfth International Conference on Learning Representations*, 2024a.

Qihan Ren, Junpeng Zhang, Yang Xu, Yue Xin, Dongrui Liu, and Quanshi Zhang. Towards the dynamics of a DNN learning symbolic interactions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=dIHXwKjXRE.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry_WPG-A-.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Hao Sun and Da-Wei Zhou. C3box: A clip-based class-incremental learning toolbox. *arXiv preprint arXiv:2601.20852*, 2026.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 247–263, 2018.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pp. 398–414. Springer, 2022.

Maorong Wang, Nicolas Michel, Ling Xiao, and Toshihiko Yamasaki. Improving plasticity in online continual learning via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23460–23469, 2024.

Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. In *ICLR*, 2021.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3014–3023, 2021.

Quanshi Zhang, Xin Wang, Jie Ren, Xu Cheng, Shuyun Lin, Yisen Wang, and Xiangming Zhu. Proving common mechanisms shared by twelve methods of boosting adversarial transferability. *arXiv preprint arXiv:2207.11694*, 2022.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13208–13217, 2020.

Bowen Zheng, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Task-agnostic guided feature expansion for class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10099–10109, 2025.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101, 2023a. doi: https://doi.org/10.1007/s11432-022-3600-y.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=S07feAlQHgM.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024a.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Explaining generalization power of a dnn using interactive concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17105–17113, 2024b.

Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. Ds-al: A dual-stream analytic learning for exemplar-free class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17237–17244, 2024.

A RELATED WORK

Class incremental learning. Previous research related to class incremental learning primarily focused on proposing different effective methods to address the challenge of catastrophic forgetting, which could be further categorized into the following three types by Liang & Li (2023). Specifically, the *regularization-based methods* (Li & Hoiem, 2017; Zhao et al., 2020) mainly added a regularization constraint to the network's weight updates to prevent important network parameters of previous steps from changing too much. The *memory-based methods* (Zhou et al., 2023b; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019) usually constructed a memory buffer to preserve knowledge of previous steps. The *expansion-based methods* (Yan et al., 2021; Wang et al., 2022; Rusu et al., 2016) dynamically expanded the network architecture for each new step to mitigate catastrophic forgetting. Despite their effectiveness, whether these methods share a common mechanism for mitigating catastrophic forgetting still remains unclear. Thus, this paper aims to use interactions to explain this shared mechanism.

However, less attention has been given to explaining catastrophic forgetting, and existing works commonly utilized accuracy-based metrics for explanation. Specifically, Chaudhry et al. (2018) calculated the difference between the highest past accuracy on a certain previous task and the accuracy on the same task evaluated by the current DNN to evaluate the forgetting of the DNN, which was further modified by Wang et al. (2024) to measure the proportion of performance the DNN forgot. Lopez-Paz & Ranzato (2017) computed the average change in accuracy on a previous task after training on a new task to measure the DNN's forgetting in the continual learning process. However, these accuracy-based metrics failed to reveal the intrinsic reason behind catastrophic forgetting, because they could not explain and identify which knowledge *w.r.t.* previous classes was forgotten and preserved in incremental learning process. Thus, they could only offer a coarse-grained explanation, leaving the fine-grained dynamics of knowledge preservation and erasure unexplored. To this end, we quantify the forgotten interactions of different complexities to provide deep insights into the core factors driving catastrophic forgetting.

Notably, the recent work (Li et al., 2025) has provided a theoretical analysis of catastrophic forgetting in a two-layer CNN, using a multi-view data model to disentangle different types of features and further tracking how these features changes during binary-classification task incremental learning. They discover that the existence of catastrophic forgetting is due to the larger signal of the task-specific feature compared to the general feature, which echoes our finding that more low-order interactions are forgotten during CIL in Section 3.1. Beyond the above similar conclusion, we further make the **first** attempt to provide a common interaction-based understanding of different CIL methods, *i.e.,* reducing the forgetting of low-order interactions, and propose a simple-yet-efficient method to verify the primary role of low-order interactions in mitigating catastrophic forgetting.

Interaction-based explanations of DNNs. Although DNNs have achieved remarkable performances on different tasks, their underlying decision-making process still remains opaque and uninterpretable to humans, potentially introducing risks. Thus, explainable AI has received increasing attention in recent years, and post-hoc explanations of DNNs is a typical direction of it. However, the disappointing view of the faithfulness of post-hoc explanations of DNNs has existed for years (Adebayo et al., 2018; Rudin, 2019). Fortunately, Ren et al. (2024a); Li & Zhang (2023); Ren et al. (2023a) have proposed the interaction metric based on game theory as a new perspective to analyze the inference logic of a DNN, and the faithfulness of taking interactions as primitive inference patterns of the DNN has been mathematically ensured by a series of theorems, which also served as the the theoretical foundation of this paper.

Thus, a line of research has employed interactions to formulate and define concepts encoded by a DNN (Ren et al., 2024a; 2023a), to mathematically explain the representation capacity of DNNs (Zhou et al., 2024b; Liu et al., 2023; Ren et al., 2021; 2024b), to unify the common mechanism behind various attribution methods (Deng et al., 2024), and explain the shared mechanism behind different transferability-boosting methods (Wang et al., 2021; Zhang et al., 2022). Thus, this paper further develops the above interaction theory system by using interactions to provide detailed explanations for catastrophic forgetting in class incremental learning, *i.e.,* quantifying the explicit forgotten knowledge, explaining the shared mechanism of classical CIL methods, and summarizing the effective factor for mitigating catastrophic forgetting.

Quantifying the knowledge encoded in DNNs. Explaining and quantifying the precise amount of knowledge in a DNN still remains a significant challenge in the field of explainable AI. A series of prior works used the mutual information between input variables and network outputs/intermediate-layer features to quantify the knowledge (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018; Higgins et al., 2017), but accurately measuring the mutual information was difficult (Kolchinsky et al., 2019).

Besides, other studies often employed human-annotated semantic concepts (Bau et al., 2017; Kim et al., 2018) or automatically learned concepts (Chen et al., 2019) to explain the knowledge encoded in the DNN, but these works lacked a mathematically guaranteed boundary that precisely defined the scope of each concept or knowledge. Thus, these studies could not quantify the exact amount of forgotten/preserved knowledge during incremental learning procedure. In comparison, the theoretically verifiable interactions allow us to represent knowledge as primitive inference patterns. In this way, we can explicitly quantify how many interactions *w.r.t.* previous steps are forgotten and preserved during class incremental learning, so as to provide detailed explanations for catastrophic forgetting.

B EXPLANATION OF EQ. (1)

Ren et al. (2023a); Harsanyi (1963) define interaction effect $I(S|\boldsymbol{x})$ as the **additional effect contributed exclusively to the network output by the AND relationship between input variables in the set S, subtracting the additional effects $I(U|\boldsymbol{x})$ that are already created by subsets $U \subsetneq S$.**

$$I(S|\boldsymbol{x}) \overset{\text{def}}{=} f(\boldsymbol{x}_S) - \sum\nolimits_{U\subsetneq S} I(U|\boldsymbol{x}) = \sum\nolimits_{U\subseteq S}(-1)^{|S|-|U|}\cdot f(\boldsymbol{x}_U).$$

To better understand the the term $(-1)^{|S|-|U|}$ in this equation, let us consider two simple examples. First, let the set $S=\{i\}$ contain only a single input variable, and its interaction effect $I(S|\boldsymbol{x})$ can be quantified as

$$I(\{i\}|\boldsymbol{x}) \overset{\text{def}}{=} f(\boldsymbol{x}_{\{i\}}) - I(\emptyset|\boldsymbol{x}) = f(\boldsymbol{x}_{\{i\}}) - f(\boldsymbol{x}_\emptyset) = \sum\nolimits_{U\subseteq S=\{i\}}(-1)^{|S|-|U|}\cdot f(\boldsymbol{x}_U), \tag{10}$$

where $\boldsymbol{x}_{\{i\}}$ denotes the masked sample crafted by keeping i unchanged and masking all other variables $N\setminus i$ to baseline values, and $\boldsymbol{x}_\emptyset$ denotes the sample in which all input variables are masked.

Second, when the set $S=\{i,j\}$ contains 2 input variables, its interaction effect $I(S|\boldsymbol{x})$ is quantified by subtracting the additional effects that are created by subsets $\{i\}$ and $\{j\}$.

$$\begin{aligned} I(\{i,j\}|\boldsymbol{x}) \overset{\text{def}}{=} & f(\boldsymbol{x}_{\{i,j\}}) - I(\{j\}|\boldsymbol{x}) - I(\{i\}|\boldsymbol{x}) - I(\emptyset|\boldsymbol{x}) \\ = & f(\boldsymbol{x}_{\{i,j\}}) - (f(\boldsymbol{x}_{\{i\}}) - f(\boldsymbol{x}_\emptyset)) - (f(\boldsymbol{x}_{\{j\}}) - f(\boldsymbol{x}_\emptyset)) - f(\boldsymbol{x}_\emptyset) \\ = & f(\boldsymbol{x}_{\{i,j\}}) - f(\boldsymbol{x}_{\{i\}}) - f(\boldsymbol{x}_{\{j\}}) + f(\boldsymbol{x}_\emptyset) \\ = & \sum\nolimits_{U\subseteq S=\{i,j\}}(-1)^{|S|-|U|}\cdot f(\boldsymbol{x}_U) \end{aligned} \tag{11}$$

Thus, when the set S contains more than 2 input variable, we can similarly recursively obtain the interaction effect $I(S|\boldsymbol{x}) \overset{\text{def}}{=} f(\boldsymbol{x}_S) - \sum_{U\subsetneq S} I(U|\boldsymbol{x}) = \sum_{U\subseteq S}(-1)^{|S|-|U|}\cdot f(\boldsymbol{x}_U)$.

C PROOF OF THEOREM 3

Theorem 3. *The change of the network output $\Delta f(m_1, m_2)$ is proven to be decomposed into interaction effects of different orders, where $\binom{m_2 n}{m}$ and $\binom{m_1 n}{m}$ denote the combinatorial number.*

$$\Delta f(m_1, m_2) = \sum\nolimits_{m=0}^{n} w^{(m)} \cdot \mathbb{E}_{S\subseteq N, |S|=m}[I(S|\boldsymbol{x})],$$
$$w^{(m)} = \begin{cases} \binom{m_2 n}{m} - \binom{m_1 n}{m}, & m \le m_1 n, \\ \binom{m_2 n}{m}, & m_1 n < m \le m_2 n, \\ 0, & m_2 n < m \le n. \end{cases}$$

Proof. The difference in network outputs between different randomly masked samples is represented as:

$$\begin{aligned} \Delta f(m_1, m_2) &= \mathbb{E}_{\substack{S_1, S_2: \emptyset \subseteq S_1 \subseteq S_2 \subseteq N \\ |S_1| = m_1 n, |S_2| = m_2 n}} [f(\boldsymbol{x}_{S_2}) - f(\boldsymbol{x}_{S_1})], \\ &= \mathbb{E}_{\substack{S_2 \subseteq N, \\ |S_2| = m_2 n}} [f(\boldsymbol{x}_{S_2})] - \mathbb{E}_{\substack{S_1 \subseteq N, \\ |S_1| = m_1 n}} [f(\boldsymbol{x}_{S_1})] \end{aligned} \tag{12}$$

where subsets S_1 and S_2 are randomly sampled from the universal set N, $0 \le m_1 \le m_2 < 1$.

Then, according to the Theorem 2, the first term can be re-written as follows, where $S_1 \subseteq N, |S_1| = m_1 n$.

$$\begin{aligned} \mathbb{E}[f(\boldsymbol{x}_{S_1})] &= \mathbb{E}_{S_1}[\sum\nolimits_{S\subseteq S_1} I(S|\boldsymbol{x})] \\ &= \mathbb{E}_{S_1}[\sum\nolimits_{m=0}^{m_1 n} \sum\nolimits_{S\subseteq S_1, |S|=m} I(S|\boldsymbol{x})] \\ &= \sum\nolimits_{m=0}^{m_1 n} \mathbb{E}_{S_1}[\binom{m_1 n}{m} \mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]] \\ &= \sum\nolimits_{m=0}^{m_1 n} \binom{m_1 n}{m} \mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]], \end{aligned} \tag{13}$$

Similarly, we can obtain,

$$\mathbb{E}[f(\boldsymbol{x}_{S_2})] = \sum\nolimits_{m=0}^{m_2 n} \binom{m_2 n}{m} \mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2, |S|=m}[I(S|\boldsymbol{x})]], \tag{14}$$

where $S_2 \subseteq N$, and $|S_2| = m_2 n$. Note that $\mathbb{E}_{S_1}\mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]$ is averaged over all subsets $S_1 \subseteq N$ and $\mathbb{E}_{S_2}\mathbb{E}_{S\subseteq S_2, |S|=m}[I(S|\boldsymbol{x})]$ is averaged over subsets $S_2 \subseteq N$. Then, we can prove Eq. (15), when $m < m_1 n < m_2 n$.

$$\begin{aligned} \mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]] &= \mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2, |S|=m}[I(S|\boldsymbol{x})]] \\ &= \mathbb{E}_{S\subseteq N, |S|=m}[I(S|\boldsymbol{x})]. \end{aligned} \tag{15}$$

Specifically, let us first consider $\mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]]$, which can be re-written as follows.

$$\begin{aligned} &\mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1, |S|=m}[I(S|\boldsymbol{x})]] = \sum_{S_1\subseteq N, |S_1|=m_1 n} \frac{1}{\binom{n}{m_1 n}} [\frac{1}{\binom{m_1 n}{m}} \sum_{S\subseteq S_1, |S|=m} I(S|\boldsymbol{x})] \\ &= \sum_{S\subseteq N, |S|=m} I(S|\boldsymbol{x}) [\sum_{S_1 \supseteq S, |S_1|=m_1 n} \frac{1}{\binom{n}{m_1 n}} \cdot \frac{1}{\binom{m_1 n}{m}}] \quad \%\text{swaping the order of summation} \\ &= \sum_{S\subseteq N, |S|=m} I(S|\boldsymbol{x}) \frac{\binom{n-m}{m_1 n - m}}{\binom{n}{m_1 n} \cdot \binom{m_1 n}{m}} \\ &= \sum_{S\subseteq N, |S|=m} I(S|\boldsymbol{x}) \frac{\binom{n-m}{m_1 n - m}}{\binom{n}{m} \cdot \binom{n-m}{m_1 n - m}} \qquad \%\text{using the combinatorial identity} \\ &= \sum_{S\subseteq N, |S|=m} I(S|\boldsymbol{x}) \frac{1}{\binom{n}{m}} \\ &= \mathbb{E}_{S\subseteq N, |S|=m}[I(S|\boldsymbol{x})]. \end{aligned} \tag{16}$$

Similarly, $\mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2,|S|=m}[I(S|\boldsymbol{x})]]$ can be re-written as follows.

$$\begin{aligned}
\mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2,|S|=m}[I(S|\boldsymbol{x})]] &= \sum_{S_2\subseteq N,|S_2|=m_2n} \frac{1}{\binom{n}{m_2n}}[\frac{1}{\binom{m_2n}{m}} \sum_{S\subseteq S_2,|S|=m} I(S|\boldsymbol{x})] \\
&= \sum_{S\subseteq N,|S|=m} I(S|\boldsymbol{x})[\sum_{S_2\supseteq S,|S_2|=m_2n} \frac{1}{\binom{n}{m_2n}} \cdot \frac{1}{\binom{m_2n}{m}}] \\
&= \sum_{S\subseteq N,|S|=m} I(S|\boldsymbol{x}) \frac{\binom{n-m}{m_2n-m}}{\binom{n}{m_2n}\cdot\binom{m_2n}{m}} \\
&= \sum_{S\subseteq N,|S|=m} I(S|\boldsymbol{x}) \frac{\binom{n-m}{m_2n-m}}{\binom{n}{m}\cdot\binom{n-m}{m_2n-m}} \\
&= \sum_{S\subseteq N,|S|=m} I(S|\boldsymbol{x}) \frac{1}{\binom{n}{m}} \\
&= \mathbb{E}_{S\subseteq N,|S|=m}[I(S|\boldsymbol{x})].
\end{aligned} \tag{17}$$

Thus, we can obtain $\mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1,|S|=m}[I(S|\boldsymbol{x})]] = \mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2,|S|=m}[I(S|\boldsymbol{x})]] = \mathbb{E}_{S\subseteq N,|S|=m}[I(S|\boldsymbol{x})]$, when $m < m_1n < m_2n$. In this way, the output change $\Delta f(m_1, m_2)$ can be rewritten as follows.

$$\begin{aligned}
\Delta f(m_1,m_2) &= \mathbb{E}_{S_1,S_2:\emptyset\subseteq S_1\subset S_2\subseteq N}[f(\boldsymbol{x}_{S_2}) - f(\boldsymbol{x}_{S_1})] \\
&= \mathbb{E}_{S_2\subseteq N,|S_2|=m_2n}[f(\boldsymbol{x}_{S_2})] - \mathbb{E}_{S_1\subseteq N,|S_1|=m_1n}[f(\boldsymbol{x}_{S_1})] \\
&= \sum\nolimits_{m=0}^{m_2n} \binom{m_2n}{m} \mathbb{E}_{S_2}[\mathbb{E}_{S\subseteq S_2,|S|=m}[I(S|\boldsymbol{x})]] - \sum\nolimits_{m=0}^{m_1n} \binom{m_1n}{m} \mathbb{E}_{S_1}[\mathbb{E}_{S\subseteq S_1,|S|=m}[I(S|\boldsymbol{x})]] \\
&= \sum\nolimits_{m=0}^{n} w^{(m)} \mathbb{E}_{S\subseteq N,|S|=m}[I(S|\boldsymbol{x})],
\end{aligned} \tag{18}$$

where

$$w^{(m)} = \begin{cases} \binom{m_2n}{m} - \binom{m_1n}{m}, & m \le m_1n, \\ \binom{m_2n}{m}, & m_1n < m \le m_2n, \\ 0, & m_2n < m \le n. \end{cases} \tag{19}$$

Thus, Theorem 3 is proven. □

D MORE RESULTS ON VERIFYING THAT THE CIL METHOD MAKES THE DNN FORGET LESS INTERACTIONS OF PREVIOUS CLASSES

D.1 VERIFICATION ON AUDIO-VISUAL CIL

In this section, we conducted experiments on another scenario of CIL, *i.e.* audio-visual CIL, to verify the generality of our explanation. Specifically, we followed settings in (Pian et al., 2023) to train two versions of autoencoders (VideoMAE for video and AudioMAE for audio) on the audio-visual dataset AVE (Tian et al., 2018) for 4 incremental steps, including a baseline model and a CIL model trained with iCaRL (Rebuffi et al., 2017). Fig. 6 shows the difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ of different orders m between the CIL model and the baseline model, which also verified our conclusion that the CIL method makes the DNN forget less interactions for previous classes to mitigate catastrophic forgetting, *i.e.,* $\mathbb{E}_m[\Delta\mathcal{I}^m_{\text{forget}}] > 0$, supporting the generality of our conclusion.

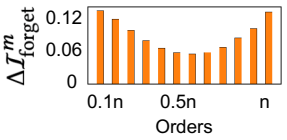


Figure 6: The difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ *w.r.t.* previous classes between the CIL model and the baseline model, both trained on the AVE dataset for audio-visual class-incremental learning.

D.2 VERIFICATION ON CLIP-BASED CIL

We further conducted experiments on CLIP-based class incremental learning to verify the generality of the above shared phenomenon, where we followed (Sun & Zhou, 2026) to adopt ZS-CLIP (Radford et al., 2021) as the baseline model and MG-CLIP (Huang et al., 2025) as the CIL method, and compared their difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ under the CIFAR100-B0 5/10/20-step and CIFAR100-B50 5/10-step settings, respectively. Fig. 7 shows $\mathbb{E}_m[\Delta\mathcal{I}^m_{\text{forget}}] > 0$, which indicated that MG-CLIP reduced the forgetting of interactions *w.r.t.* previous classes, particularly low-order interactions. This verified the generality of our shared phenomenon.

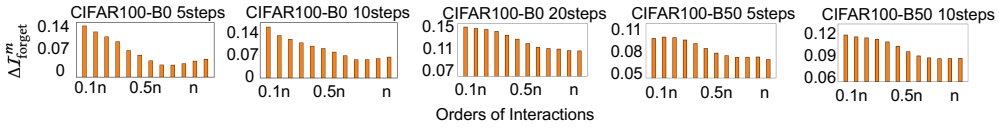


Figure 7: The difference in forgotten interactions $\Delta\mathcal{I}^m_{\text{forget}}$ *w.r.t.* previous classes between the CIL model and the baseline model in CLIP-based class incremental learning.

E EXPLAINING WHY MEMO MAKES THE DNN FORGET MORE HIGH-ORDER INTERACTIONS

In this section, we explain the observation that MEMO (Zhou et al., 2023b) makes its corresponding CIL model forget more high-order interactions in Fig. 4 and Fig. 3, compared to the baseline model. We consider that the CIL model trained by MEMO forgets more high-order interaction *w.r.t.* previous steps, in order to learn new interactions *w.r.t.* current classes.

To this end, we first use interactions to quantify the learning of new interactions of each complexity for the inference of current classes. Specifically, given an input sample $\boldsymbol{x}_t \in \mathcal{D}_t$ and the DNN f_t incrementally learned from steps 1 to t, the strength of m-order newly learned interaction $\mathcal{I}^{(t),m}_{\text{new}}(S|\boldsymbol{x}_t)$ is defined as the interaction encoded in f_t but not encoded in the previous DNN f_{t-1}.

$$\begin{aligned}&\forall S \subseteq N, |S| = m, \quad \mathcal{I}^{(t),m}_{\text{new}}(S|\boldsymbol{x}_t) = \mathcal{I}^m(S|\boldsymbol{x}_t, f_t) - \mathcal{I}^{(t),m}_{\text{share}}(S|\boldsymbol{x}_t),\\ &\mathcal{I}^{(t),m}_{\text{share}}(S|\boldsymbol{x}_t) = \Gamma^t_{t-1}(S|\boldsymbol{x}_t) \cdot \min(\mathcal{I}^m(S|\boldsymbol{x}_t, f_t), \mathcal{I}^m(S|\boldsymbol{x}_t, f_{t-1})),\end{aligned} \tag{20}$$

where $\mathcal{I}^m(S|\boldsymbol{x}_t, f_t) \triangleq |I^m(S|\boldsymbol{x}_t, f_t)|$ denotes the strength of the m-order interaction $I^m(S|\boldsymbol{x}_t, f_t)$ extracted from the sample $\boldsymbol{x}_t \in \mathcal{D}_t$. $\mathcal{I}^{(t),m}_{\text{share}}(S|\boldsymbol{x}_t)$ represents the m-order interaction shared by f_{t-1} and f_t. $\Gamma^t_{t-1}(S|\boldsymbol{x}_t) = \mathbb{1}((I^m(S|\boldsymbol{x}_t, f_t) \cdot I^m(S|\boldsymbol{x}_t, f_{t-1})) > 0)$ measures whether the m-order interaction $I^m(S|\boldsymbol{x}_t, f_t)$ encoded by the current DNN f_t has the same effect as $I^m(S|\boldsymbol{x}_t, f_{t-1})$ encoded by the previous DNN f_{t-1}, where $\mathbb{1}(\cdot)$ is the indicator function. If $I^m(S|\boldsymbol{x}_t, f_t)$ and $I^m(S|\boldsymbol{x}_t, f_{t-1})$ have opposite effects, then the shared interaction $\mathcal{I}^{(t),m}_{\text{share}}(S|\boldsymbol{x}_t) = 0$. Otherwise, the shared interaction is quantified as $\mathcal{I}^{(t),m}_{\text{share}}(S|\boldsymbol{x}_t) = \min(\mathcal{I}^m(S|\boldsymbol{x}_t, f_t), \mathcal{I}^m(S|\boldsymbol{x}_t, f_{t-1}))$.

Then, we compare the difference in newly learned interactions $\Delta\mathcal{I}^m_{\text{learn}}$ between the baseline model and the CIL model trained by MEMO, in order to check whether the CIL model trained by MEMO learns more new high-order interactions than the baseline model.

$$\begin{aligned}\Delta\mathcal{I}^m_{\text{learn}} &= \mathcal{I}^m_{\text{learn,CIL}} - \mathcal{I}^m_{\text{learn, base}}\\ \mathcal{I}^m_{\text{learn,base}} &= \frac{1}{T-2}\sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{x}_t \in \mathcal{D}_t} \mathbb{E}_{\substack{S\subseteq N,\\ |S|=m}} [\mathcal{I}^{(t),m}_{\text{new, base}}(S|\boldsymbol{x}_t)]\\ \mathcal{I}^m_{\text{learn,CIL}} &= \frac{1}{T-2}\sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{x}_t \in \mathcal{D}_t} \mathbb{E}_{\substack{S\subseteq N,\\ |S|=m}} [\mathcal{I}^{(t),m}_{\text{learn,CIL}}(S|\boldsymbol{x}_t)],\end{aligned} \tag{21}$$

where $\mathcal{I}^{(t),m}_{\text{learn,CIL}}(S|\boldsymbol{x}_t)$ denotes the new interactions *w.r.t.* the current step t learned by the CIL model, which is obtained by applying $f_t = f_{t,\text{CIL}}$ to compute Eq. (20). Accordingly, $\mathcal{I}^{(t),m}_{\text{learn,base}}(S|\boldsymbol{x}_t)$ represents the new interactions learned by the baseline model, which is obtained by applying $f_t = f_{t,\text{base}}$ to calculate Eq. (20). Thus, a positive value of $\Delta I^m_{\text{learn}}$ indicates that the CIL model learns more m-order interactions *w.r.t.* new classes.

Fig. 8 shows that the CIL model trained by MEMO usually learns more high-order interactions *w.r.t.* current classes, compared to the baseline model. This may partially explain MEMO makes its corresponding CIL model forget more high-order interactions *w.r.t.* previous classes.

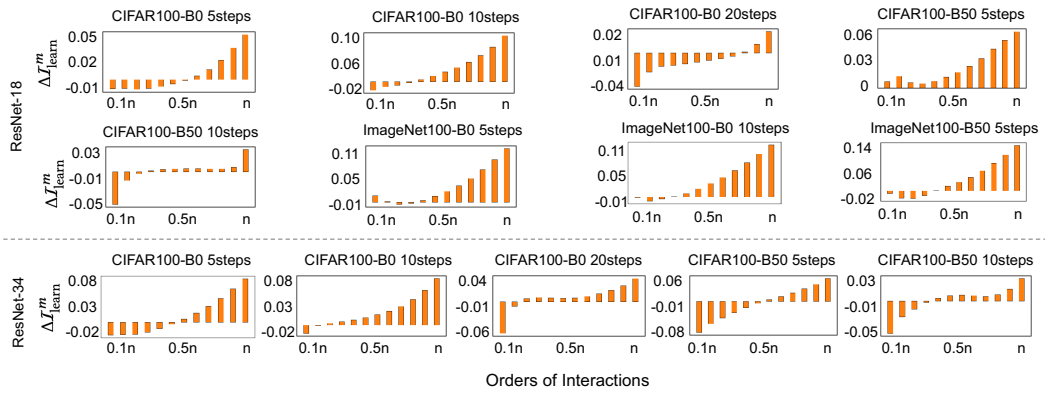


Figure 8: The difference in newly learned interactions $\Delta\mathcal{I}^m_{\text{learn}}$ between the baseline model and the CIL model trained by MEMO.

F PRELIMINARY ATTEMPT OF USING LOW-ORDER INTERACTION

In this section, we further verified that low-order interactions served as effective factors in class incremental learning from a complementary perspective, *i.e.,* examining whether encouraging the DNN to rely more on low-order interactions could effectively mitigate catastrophic forgetting and improve the CIL performance. Specifically, we minimized the loss function $L_{\text{encourage}}(m_1, m_2)$ to force the DNN to mainly use interactions encoded in $\Delta f(m_1, m_2)$ for incremental learning, thereby boosting the learning of $[m_1 n, m_2 n]$-order interactions.

$$L_{\text{encourage}}(m_1, m_2) = L_{\text{classification}} + \alpha \cdot L_{\text{inter}}(m_1, m_2). \tag{22}$$

To this end, we trained ResNet-18 model on the CIFAR-100 dataset under different class splits in Section 3.1 for CIL. For each DNN, we trained three versions, including a baseline model learned without $L_{\text{inter}}(m_1, m_2)$ by setting $\alpha = 0$, and two DNNs trained with $\alpha = 1.0$ and respectively setting $m_1 = 0, m_2 = 0.3$ and $m_1 = 0.7, m_2 = 1.0$ to encourage the DNN to mainly rely on low-order and high-order interactions for inference.

Apart from the baseline model, we also incorporated $L_{\text{encourage}}(m_1, m_2)$ into the training of the CIL model to verify the role of low-order interactions. Specifically, given a CIL method, we also trained three versions, including a normally trained CIL model without $L_{\text{inter}}(m_1, m_2)$, and two CIL models respectively trained with $L_{\text{inter}}(0, 0.3n)$ and $L_{\text{inter}}(0.7n, n)$ to encourage the encoding of low-order and high-order interactions.

Table 3 reports the final test accuracy after the whole incremental process, abbreviated as **Last**, and the averaged test accuracy over all incremental steps, abbreviated as **Avg**, where each reported result was averaged over multiple runs. We discovered that encouraging the DNN to encode more low-order interactions could effectively mitigate catastrophic forgetting in both the baseline and CIL model, *i.e.,* improving both the averaged test accuracy and the final test accuracy, without introducing any additional trainable parameters or modules. In comparison, forcing the DNN to preserve more high-order interactions brought almost no improvement. For example, under the CIFAR100-B0 5-step setting, encouraging low-order interactions improved the Last/Avg performance of the baseline model from $17.37/38.59$ to $28.20/43.90$. In contrast, encouraging high-order interactions resulted in only marginal changes, yielding $18.05/38.61$. Similarly, encouraging low-order interactions in the CIL model trained by TagFex improved both Last/Avg from $71.30/78.60$ to $75.14/82.93$ in the CIFAR100-B0 5-step setting, whereas encouraging high-order interactions leads to only limited changes, with Last/Avg reaching $71.86/79.42$. Thus, the above phenomenon further supported that low-order interactions constituted beneficial and practically exploitable factors for mitigating catastrophic forgetting.

Table 3: Effects of encouraging low-order and high-order interactions on CIL performance. Last denotes the final test accuracy after the whole incremental process, and Avg is referred to as the averaged test accuracy over all incremental steps.

Model	CIFAR100-B0 5 steps		CIFAR100-B0 10 steps		CIFAR100-B0 20 steps		CIFAR100-B50 5 steps		CIFAR100-B50 10 steps	
	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg
Baseline	17.37	38.59	9.38	25.24	4.81	17.07	9.36	23.21	5.13	13.48
Baseline+ $L(0, 0.3n)$	**28.20**	**43.90**	**14.73**	**28.41**	**7.94**	**21.53**	**15.27**	**26.08**	**7.64**	**16.47**
Baseline+ $L(0.7n, n)$	18.05	38.61	9.41	25.53	5.15	17.32	9.84	23.95	5.42	13.71
DER	58.65	69.89	65.85	75.71	62.53	74.04	68.06	72.95	67.37	72.50
DER+ $L(0, 0.3n)$	**63.20**	**72.69**	**69.73**	**78.84**	**65.91**	**78.63**	**71.84**	**76.37**	**70.91**	**76.28**
DER+ $L(0.7n, n)$	59.01	69.99	65.52	75.58	62.87	74.11	68.19	72.88	67.08	72.64
FOSTER	63.35	72.46	60.15	71.80	56.50	69.56	63.63	70.09	60.71	68.05
FOSTER+ $L(0, 0.3n)$	**67.82**	**74.62**	**64.27**	**74.53**	**60.49**	**74.08**	**67.55**	**76.31**	**64.18**	**71.84**
FOSTER+ $L(0.7n, n)$	63.71	72.69	61.83	71.87	56.94	69.62	63.94	70.22	60.89	68.93
TagFex	71.30	78.60	68.15	78.32	64.90	76.80	70.28	75.84	69.60	75.40
TagFex+ $L(0, 0.3n)$	**75.14**	**82.93**	**71.72**	**80.15**	**68.27**	**79.91**	**73.85**	**79.63**	**73.18**	**78.69**
TagFex+ $L(0.7n, n)$	71.86	79.42	68.49	78.50	66.57	77.69	70.93	76.27	69.87	75.51
DS-AL	59.40	56.56	52.60	44.47	27.80	35.35	64.44	64.38	64.70	63.77
DS-AL+ $L(0, 0.3n)$	**64.72**	**59.85**	**57.83**	**47.28**	**32.95**	**39.16**	**67.53**	**67.71**	**67.36**	**67.54**
DS-AL+ $L(0.7n, n)$	59.97	58.43	52.93	45.19	28.05	36.33	66.18	65.29	65.26	63.98

G MORE RESULTS ON EXPLORING THE ROLE OF LOW-ORDER INTERACTIONS

In this section, we conducted experiments to explore the role of low-order interactions in the resistance of catastrophic forgetting. To this end, we trained ResNet-18 and ResNet-34 models on CIFAR-100 datasets under different class splits in Section 3.1 for class incremental learning. For each DNN, we trained three versions, including one using a specific CIL method without any interaction penalization (*i.e.,* $\alpha = 0$, the CIL model introduced in Section 3.2), and two others with the same CIL method but applying interaction penalization ($\alpha = 1.0$), by setting $m_1 = 0, m_2 = 0.3$ and $m_1 = 0.7, m_2 = 1.0$ in $L(m_1, m_2)$, respectively. For simplicity, we named the CIL model trained to penalize $[0, 0.3n]$-order ($[0.7n, n]$-order) interactions ***high-order (low-order) CIL model***, as it mainly encoded high-order (low-order) interactions. Then, we compared the stability difference $\Delta FM_{\text{CIL, low}}$ between the low-order CIL model $f_{\text{low, CIL}}$ and the CIL model f_{CIL}, as well as $\Delta FM_{\text{CIL, high}}$ between the high-order CIL model $f_{\text{CIL, high}}$ and the CIL model f_{CIL}, *i.e.,* $\Delta FM_{\text{CIL, low}} = FM(f_{\text{CIL, low}}) - FM(f_{\text{CIL}})$ and $\Delta FM_{\text{CIL, high}} = FM(f_{\text{CIL, high}}) - FM(f_{\text{CIL}})$.

Table 4 illustrated that $\Delta FM_{\text{CIL, high}}$ was consistently larger than $\Delta FM_{\text{CIL, low}}$, which indicated that the compared to the CIL, model, the stability of high-order CIL models was much worse than that of the low-order CIL model. Thus, preventing the DNN from encoding low-order interactions could significantly harm its stability in resisting catastrophic forgetting, which suggested that low-order interactions might, to some extent, be an effective factor for mitigating catastrophic forgetting.

Table 4: The stability difference $\Delta FM_{\text{CIL, low}}$ between the low-order CIL model and the CIL model, as well as $\Delta FM_{\text{CIL, high}}$ between the high-order CIL model and the CIL model.

Model	Metric	CIFAR100-B0 5 steps	CIFAR100-B0 10 steps	CIFAR100-B0 20 steps	CIFAR100-B50 5 steps
ResNet-18	$\Delta FM_{\text{MEMO, low}}$	0.08	0.07	0.08	0.02
	$\Delta FM_{\text{Memo,high}}$	0.17	0.16	0.20	0.13
ResNet-34	$\Delta FM_{\text{MEMO,low}}$	0.18	0.1	0.10	0.10
	$\Delta FM_{\text{MEMO,high}}$	0.30	0.26	0.23	0.15
ResNet-18	$\Delta FM_{\text{LWF, low}}$	0.20	0.11	0.06	0.15
	$\Delta FM_{\text{LWF,high}}$	0.32	0.25	0.29	0.33
ResNet-34	$\Delta FM_{\text{LWF, low}}$	0.07	0.08	0.09	0.04
	$\Delta FM_{\text{LWF, high}}$	0.16	0.21	0.23	0.18
ResNet-18	$\Delta FM_{\text{DER, low}}$	0.15	0.09	0.07	0.06
	$\Delta FM_{\text{DER, high}}$	0.32	0.28	0.19	0.24
ResNet-34	$\Delta FM_{\text{DER, low}}$	0.07	0.12	0.15	0.07
	$\Delta FM_{\text{DER, high}}$	0.18	0.26	0.42	0.34
ResNet-18	$\Delta FM_{\text{FOSTER, low}}$	0.02	0.05	0.05	0.09
	$\Delta FM_{\text{FOSTER, high}}$	0.21	0.25	0.21	0.30
ResNet-34	$\Delta FM_{\text{FOSTER, low}}$	0.04	0.07	0.14	0.08
	$\Delta FM_{\text{FOSTER, high}}$	0.13	0.29	0.23	0.26

H SPARSITY OF INTERACTIONS

Given an input sample $\boldsymbol{x}$ with n input variables, among all 2^n possible interactions, Ren et al. (2024a) have proven that the number of interactions with salient effects on the network output is $\mathcal{O}(n^{\kappa}/\tau)$ under three common mathematical conditions. First, the DNN does not encode interactions of extremely high orders. Second, the average classification confidence monotonically decreases as more input variables are masked, which is computed on $\{\boldsymbol{x}_T||T| = n - k\}$ by masking different random sets of k input variables. Third, the decreasing speed of the average confidence is polynomial. κ is empirically within the range of $[1.9, 2.2]$ Ren et al. (2024b). This indicates that the number of salient interactions is much less than 2^n, *i.e.,* the interactions are sparse.

To this end, we conducted experiments on incrementally trained DNNs introduced in Section 3.1 to examine whether interactions satisfy the sparsity property in real applications. Fig. 9 shows all the interaction effects encoded by the incrementally trained DNN on different input samples. We discovered that only a few interactions have salient interaction effects $|I(S|\boldsymbol{x})|$ on the network output, while the effects of all other interactions are negligible, *w.r.t.* $|I(S|\boldsymbol{x})| \approx 0$. This finding was consistent with the conclusion found by Ren et al. (2024a), *i.e.,* interactions encoded by a DNN were usually very sparse.

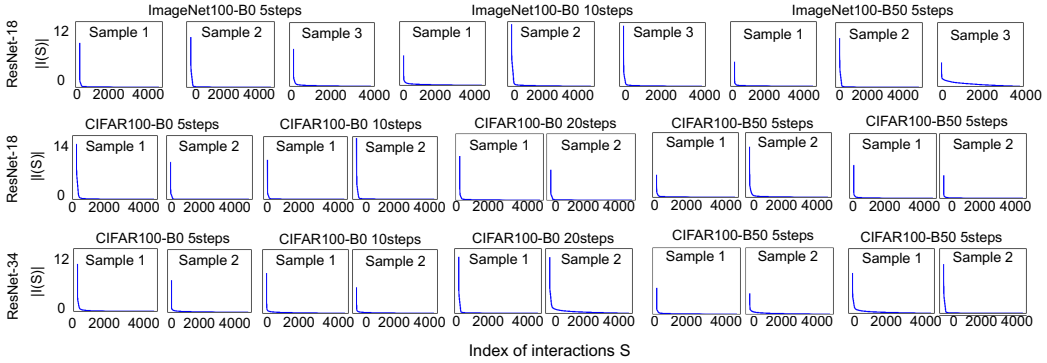


Figure 9: Interactions in descending order of the interaction strength.

I EXPERIMENTAL DETAILS

In this paper, we follow settings in (Li & Zhang, 2023; Ren et al., 2024a; Liu et al., 2023) to compute interactions. Considering the computational cost of calculating and extracting all 2^n interactions is intolerable in real implementation, we follow the method in (Li & Zhang, 2023; Ren et al., 2024a) to annotate and select a set of image patches as input variables to reduce the computation cost. Specifically, given an image in the CIFAR-100 dataset, we follow (Li & Zhang, 2023; Ren et al., 2023a) to divide it into small patches of size 4×4, resulting in a total of 8×8 image patches, and further select $n = 12$ patches from 6×6 image patches located in the center of the image as input variables to calculated interactions, because Li & Zhang (2023); Ren et al. (2024a) consider the DNN mainly used foreground information to make inference. Similarly, for each image in the ImageNet-100 dataset, we follow (Liu et al., 2023) to divide it into small patches of size 28×28, resulting in a total of 8×8 image patches, and further select $n = 12$ patches from 6×6 image patches located in the center of the image as input variables to calculated interactions. Thus, we calculate totally $2^{12} = 1024$ interactions for each input image, which costs only 2.55 seconds per image in CIFAR-100 on ResNet-18 and 2.91 seconds on ResNet-34, and 12.37 seconds per image in ImageNet-100 on ResNet-18, using the single NVIDIA 4090 GPU.

Moreover, we conduct experiments to verify that our conclusion is stable, under **different patch sizes, different patch-selection strategies, different task orders, different numbers of selected input variables, and input variables selected from different image regions**. Specifically, we (1) change the patch size from 4×4 to 8×8, (2) use the classic attribution method Integrated Gradients (Sundararajan et al., 2017) to identify the most salient image patches as input variables to compute interactions, (3) randomly shuffle the task orders, (4) vary the number of selected input variables from 12 to 14, and (5) randomly select 12 patches from the entire image, rather than from the center of the image. For this analysis, we use FOSTER as the CIL method, and train both the CIL model and baseline model under the setting of CIFAR100-B0 5 steps for class incremental learning. We compute the Jaccard similarity metric to check whether our conclusion remained stable under different interaction-computation settings, where $\Delta\mathcal{I}^m_{\text{forget},a}$ was computed under the default setting, while $\Delta\mathcal{I}^m_{\text{forget},b}$ was calculated under each modified setting.

$$\textit{Jaccard Similarity} = \frac{\sum_m \min(\Delta\mathcal{I}^m_{\text{forget},a}, \Delta\mathcal{I}^m_{\text{forget},b})}{\sum_m \max(\Delta\mathcal{I}^m_{\text{forget},a}, \Delta\mathcal{I}^m_{\text{forget},b})} \tag{23}$$

Fig. 10 and Fig. 11 show similar trends under different interaction-computation settings with high Jaccard similarity, which verified the stability of our conclusion that CIL method made the DNN forget less interactions *w.r.t.* previous classes.

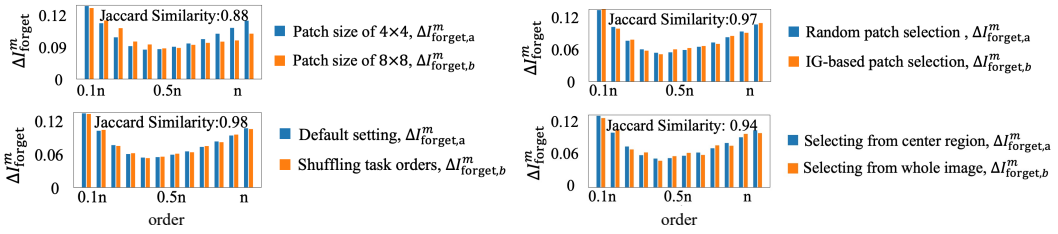


Figure 10: Ablation study of interaction computation *w.r.t.* different patch sizes, patch-selection strategies, task orders, and input variables selected from different image regions.

Besides, to generate the masked sample $\boldsymbol{x}_T$, we follow the widely used setting in (Dabkowski & Gal, 2017; Li & Zhang, 2023; Ren et al., 2024a; Liu et al., 2023) to set the baseline value of each variable b_i as the mean value of this variable across all samples in image classification to mask each input variable in $N \setminus T$. Meanwhile, given each incrementally trained DNN f, we follow (Li & Zhang, 2023; Ren et al., 2024a; Liu et al., 2023) to set $f(\boldsymbol{x})$ in Eq. (2) of the main paper as the confidence of predicting $\boldsymbol{x}$ to the ground-truth category y^{truth}, $f(\boldsymbol{x}) = \log \frac{p(y=y^{\text{truth}}|\boldsymbol{x})}{1-p(y=y^{\text{truth}}|\boldsymbol{x})}$, to compute each interaction $I(S|\boldsymbol{x})$. According to Theorem 2, we normalize each interaction $I(S|\boldsymbol{x})$ encoded by the DNN f as $I(S|\boldsymbol{x}) \leftarrow I(S|\boldsymbol{x})/\mathbb{E}_{\boldsymbol{x}}[|f(\boldsymbol{x}) - f(\boldsymbol{x}_\emptyset)|]$ to enable fair comparisons between different

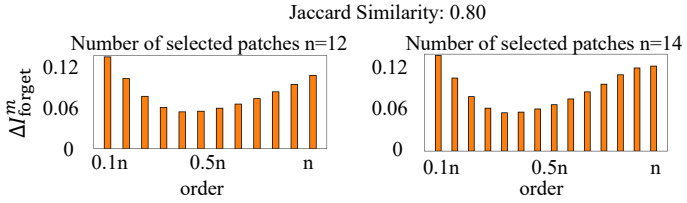


Figure 11: Ablation study of interaction computation *w.r.t.* different numbers of selected input variables.

DNNs. We follow the widely-used setting in Ancona et al. (2019); Ren et al. (2024a) to set the baseline value b_i as the mean value of this variable i over all samples in image classification.

Additionally, we follow training settings in (Zhou et al., 2024a) to use SGD with an initial learning rate of 0.1 and momentum of 0.9 to train DNNs introduced in Section 3 for class incremental learning based on PyTorch and PyCIL[6] (Zhou et al., 2023a). In Section 3.3, we trained DNNs to encode interactions of specific orders by setting set $m_1 = 0, m_2 = 0.3$ and $\alpha = 1.0$ in $L(m_1, m_2)$ to train high-order DNNs, and set $m_1 = 0.7, m_2 = 1.0$ and $\alpha = 1.0$ to train low-order DNNs. This training process was roughly summarized in Algorithm 1 for a better understanding. We will release the code if the paper is accepted.

Algorithm 1 Training the DNN to encode interactions of specific orders.

Input: Training dataset $\mathcal{D}_{\text{train}}$, interaction orders m_1 and m_2, coefficient α, epoch number E
Output: a trained DNN f

Initialize parameters of f
for $e = 1$ to E **do**
  Initialize $L_{\text{inter}}(m_1, m_2) = 0$ and $L_{\text{classification}} = 0$
  for $(\boldsymbol{x}, y) \in \mathcal{D}_{\text{train}}$ **do**
    Compute the network output change $\Delta f(m_1, m_2, \boldsymbol{x})$
    Compute the loss $L_{\text{inter}}(m_1, m_2, \boldsymbol{x}) = -\sum_{c=1}^{C}[p(\hat{y} = c|\Delta f(m_1, m_2, \boldsymbol{x})_c) \cdot \log(p(\hat{y} = c|\Delta f(m_1, m_2, \boldsymbol{x})_c))]$
    Compute the classification loss $L_{\text{classification}}(\boldsymbol{x}) = \text{CrossEntropy}(\boldsymbol{x}, y)$
    Compute $L_{\text{inter}}(m_1, m_2) += L_{\text{inter}}(m_1, m_2, \boldsymbol{x}), L_{\text{classification}} += L_{\text{classification}}(\boldsymbol{x})$
  end for
  Compute $L_{\text{inter}}(m_1, m_2) /= |\mathcal{D}_{\text{train}}|, L_{\text{classification}} /= |\mathcal{D}_{\text{train}}|$
  Compute the loss $L(m_1, m_2) = L_{\text{classification}} - \alpha \cdot L_{\text{inter}}(m_1, m_2)$
  Compute the gradient of the loss $L(m_1, m_2)$ to update parameters of f
end for

[6]PyCIL is an open-source python tool box to implement CIL.