

# A GENERALIZED INFORMATION BOTTLENECK THEORY OF DEEP LEARNING

005 **Anonymous authors**

006 Paper under double-blind review

## ABSTRACT

011 The Information Bottleneck (IB) principle offers a compelling theoretical  
 012 framework to understand how neural networks (NNs) learn. However, its  
 013 practical utility has been constrained by unresolved theoretical ambiguities  
 014 and significant challenges in accurate estimation. In this paper, we present a  
 015 *Generalized Information Bottleneck (GIB)* framework that reformulates the  
 016 original IB principle through the lens of synergy, i.e., the information obtain-  
 017 able only through joint processing of features. We provide theoretical and  
 018 empirical evidence demonstrating that synergistic functions achieve superior  
 019 generalization compared to their non-synergistic counterparts. Building on  
 020 these foundations we re-formulate the IB using a computable definition of  
 021 synergy based on the average interaction information (II) of each feature  
 022 with those remaining. We demonstrate that the original IB objective is upper  
 023 bounded by our GIB in the case of perfect estimation, ensuring compatibility  
 024 with existing IB theory while addressing its limitations. Our experimen-  
 025 tal results demonstrate that GIB consistently exhibits compression phases  
 026 across a wide range of architectures (including those with *ReLU* activations  
 027 where the standard IB fails), while yielding interpretable dynamics in both  
 028 CNNs and Transformers and aligning more closely with our understanding  
 029 of adversarial robustness.

## 1 INTRODUCTION

030 Deep learning has achieved remarkable practical success, yet our theoretical understanding  
 031 of how neural networks learn effective representations remains incomplete (Shwartz-Ziv  
 032 & Tishby, 2017). Information theory offers a principled framework for analyzing deep  
 033 learning, as information-theoretic quantities are invariant to invertible transformations and  
 034 provide interpretable units of measurement (Cover & Thomas, 1991). The Information  
 035 Bottleneck (IB) principle, introduced by Tishby et al. (1999), has emerged as a particu-  
 036 larly influential framework for understanding neural network learning dynamics, providing  
 037 insights into diverse phenomena including  
 038 adversarial robustness (Ma et al., 2021), the  
 039 effects of dropout regularization (Achille &  
 040 Soatto, 2018), and generalization bounds  
 041 (Kawaguchi et al., 2023). Through this inter-  
 042 pretation, the activations of our network  
 043 can be viewed as a hidden state representa-  
 044 tion  $\mathcal{T}$ , which converges to a set of statis-  
 045 tics captured by two competing terms. The  
 046 first term, referred to as the *prediction term*,  
 047 quantifies the mutual information (MI) be-  
 048 tween the hidden representation  $\mathcal{T}$  and the  
 049 target  $Y$ , denoted as  $I(Y; \mathcal{T})^1$ . It is straight-  
 050 forward to see that achieving training (and  
 051 consequently test) accuracy above random  
 052

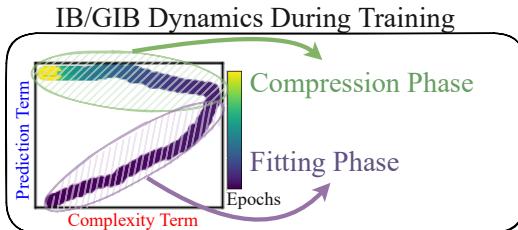


Figure 1: This schematic illustrates information plane dynamics during training, with trajectories color-coded from early epochs (light colors) to late epochs (dark purple), showing distinct fitting and compression phases.

<sup>1</sup>For the mathematical notation used throughout this paper, see Appendix A.

054 guessing requires a network whose learned representation is well aligned with that of the  
 055 target data. However, it is well established that optimizing solely for prediction accuracy can  
 056 lead to overfitting. Consequently, the IB framework introduces a second term: the *complexity*  
 057 *term*  $I(\mathcal{X}; \mathcal{T})$ , which quantifies the mutual information between the input data  $\mathcal{X}$  and the  
 058 hidden representation. Optimizing this less intuitive term can be interpreted as an effort to  
 059 minimize redundant and irrelevant information from the input that is encoded in the latent  
 060 space.

061 Overall, the IB framework posits that deep neural networks learn by solving the following  
 062 Lagrangian optimization problem:

$$\mathcal{L}_{\text{IB}} = \max_{p(\mathcal{T}|\mathcal{X})} \left[ \underbrace{I(\mathcal{T}; \mathcal{Y})}_{\text{prediction term}} - \underbrace{\beta^{-1} I(\mathcal{X}; \mathcal{T})}_{\text{complexity term}} \right] \quad (1)$$

063 Shwartz-Ziv & Tishby (2017) suggested that the success of deep learning can be attributed  
 064 to the ability of NNs to perform the aforementioned optimization problem in two distinct  
 065 phases. First, a *fitting phase*, where both of the two introduced terms increase, and second  
 066 a *compression phase*, where the complexity term decreases in size (refer to Figure 1 for a  
 067 visual illustration of these processes). It was argued that this second compression phase  
 068 was unique to deep models and helped explain their generalizability. In some cases, the  
 069 flow of information through the latent space has been shown to align precisely with the IB's  
 070 theoretical predictions (Shwartz-Ziv & Tishby, 2017).

071 While the IB framework initially seemed to provide a complete explanation of how neural  
 072 networks balance compression and predictive accuracy, Saxe et al. (2019) presented counterexamples  
 073 that challenge this view. In particular, the authors showed that compression  
 074 phases depend critically on the choice of activation function: networks with *tanh* activations  
 075 exhibited compression across all layers, whereas *ReLU*-based networks did not. Despite the  
 076 absence of a compression phase, the *ReLU* networks still generalized well. According to  
 077 Goldfeld et al. (2019), this occurs because the complexity term in deterministic networks  
 078 is theoretically constant or infinite, rendering compression impossible. Consequently, the  
 079 compression observed in *tanh* networks was not a genuine information-theoretic effect, but  
 080 rather the result of injected randomness (Saxe et al., 2019; Shwartz-Ziv & Tishby, 2017;  
 081 Geiger, 2022).

082 In this paper, we address these issues by introducing a generalized formulation of the IB  
 083 framework that is grounded in synergy. Synergy, a concept from multivariate information  
 084 theory, captures the extra predictive power that arises when inputs are considered together  
 085 rather than in isolation (Williams & Beer, 2010). To motivate this perspective, we begin  
 086 by asking: why synergy? We then present both theoretical arguments and empirical results  
 087 showing that synergistic functions lead to improved generalization.

088 Having established that synergistic functions generalize better, we construct the GIB by  
 089 reformulating the IB through the lens of synergy. First, we introduce a point-wise mutual  
 090 information (PMI)-based reweighting scheme that ensures we measure synergy specifically  
 091 for correct predictions rather than arbitrary outputs. We then combine this reweighting  
 092 with our feature-wise synergy decomposition, which uses the interaction information (II) to  
 093 quantify information available only through joint processing of all features. Finally, we cast  
 094 this as a Lagrangian optimization problem, yielding our GIB objective that measures how  
 095 synergistically the input features combine to describe correct outputs.

096 After deriving the GIB, we prove that, under perfect estimation, it can be lower bounded by  
 097 the IB. Importantly, our formulation overcomes key theoretical limitations of standard IB,  
 098 including the issue of infinite complexity terms. We demonstrate that GIB exhibits clear  
 099 compression phases and interpretable learning dynamics across a wide range of scenarios  
 100 where standard IB fails. In Figure 2, we revisit the experiments presented in Saxe et al. (2019)  
 101 and show that the GIB displays compression phases for five different activation functions  
 102 while the IB is limited to one<sup>2</sup>. Beyond these synthetic examples, we observe consistent

103 <sup>2</sup>**Reading Information Planes Plots.** Throughout, we visualize information dynamics using  
 104 information plane plots. In these plots, the  $x$ -axis represents the complexity term and the  $y$ -axis

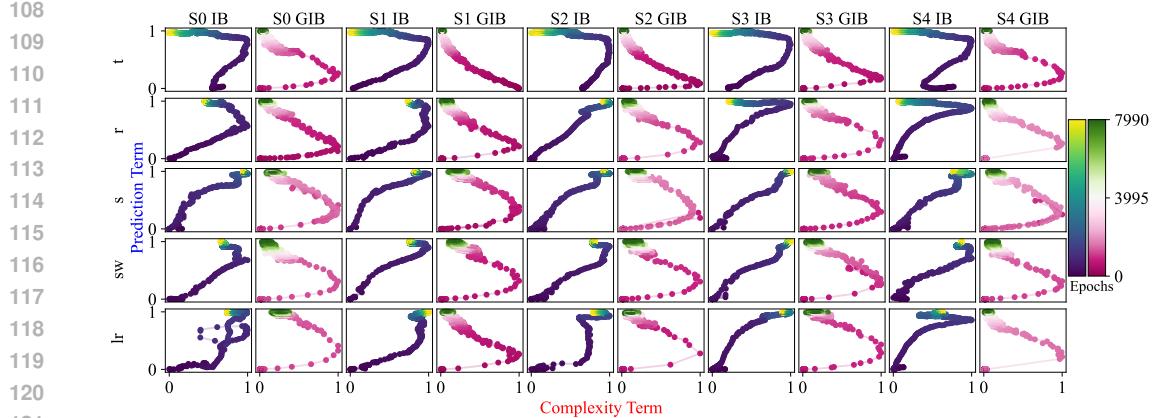


Figure 2: Information plane dynamics across multiple activation functions, extending Shwartz-Ziv & Tishby (2017) and Saxe et al. (2019) beyond *tanh* and *ReLU* to include *softplus*, *swish*, and *leaky ReLU*. Standard IB (blue) shows compression only for *tanh*; GIB (pink) shows compression for all activation functions. Each column represents one seed.

information dynamics in practical deep learning settings including ResNets on CIFAR-10 and BERT fine-tuning. Furthermore, the complexity term in our framework provides meaningful insights into model behavior under adversarial attacks, correctly tracking vulnerability where standard IB fails. Code for full reproducibility of these results will be made publicly available upon publication.

## 2 RELATED WORK

**MI Estimation.** MI estimation in the IB framework remains an active area of research and debate. Numerous estimators exist (e.g., *k*-nearest-neighbor (kNN) methods (Kozachenko & Leonenko, 1987; Kraskov et al., 2004), and kernel-density approaches (Kandasamy et al., 2015; Han et al., 2017)), and trainable neural estimators (Belghazi et al., 2018)), yet many information-theoretic studies of deep networks (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2019) discretize neuron outputs (“binning”) to approximate MI. Binning is simple and fast, but even moderate coarse-graining can introduce substantial estimation error (Goldfeld et al., 2019). Despite these limitations, we use binning because MI estimates are needed throughout training (e.g., at each epoch); running kNN, KDE, or variational estimators at this frequency would be prohibitively slow and numerically unstable in high-dimensional settings. Our goal is to track relative trends in MI rather than obtain exact differential values, and binning provides a tractable, reproducible proxy that makes per-training-step MI monitoring feasible.

**Other Generalizations of the Information Bottleneck.** There exist many extensions of the IB framework. For instance, the variational information bottleneck (VIB) introduces stochastic neural parameterizations to scale IB to deep networks (Alemi et al., 2017), while Information dropout applies a neuron-wise IB-like penalty via multiplicative noise to improve generalization (Achille & Soatto, 2018). The HSIC bottleneck (Wang et al., 2021) replaces mutual information with the Hilbert–Schmidt independence criterion to regularize intermediate representations for adversarial robustness. More recent work has extended the IB to multivariate and deep settings. Matrix-based Rényi’s  $\alpha$ -order entropy functionals (Yu

represents the prediction term. For standard IB, these are  $I(\mathcal{X}; \mathcal{T})$  and  $I(\mathcal{T}; Y)$ , respectively. For the full formulation of the GIB, see Section 3. Trajectories are color-coded by training epoch, progressing from early training (dark pink/blue) to late training (light green/yellow). Blue trajectories correspond to the standard IB dynamics, whereas pink trajectories depict the dynamics under our GIB formulation. Movement leftward indicates compression (reduction of redundant information), while movement upward indicates improved prediction. For the IB we only report the information plane of the final layer as this is where compression dynamics are most readily observed. Meanwhile, the GIB is formulated based on inputs and therefore only produces one information plane per training. For clarity of presentation, we normalize the complexity term and prediction term results between 0 and 1.

et al., 2020; 2021a) provide estimators of multivariate entropies and dependence. These have been used to develop deterministic IB objectives in deep networks (Yu et al., 2021b), and to design gated IB objectives for sequential environments (Alesiani et al., 2023). The multivariate IB of Friedman et al. (2001) captures statistical dependencies between multiple bottleneck variables through graphical models. Most relevant to our work, recent advances have also connected IB and Partial Information Decomposition (PID), with Kolchinsky (2024) building on earlier IB work such as Kolchinsky et al. (2019), showing that PID redundancy can be isolated via an IB-style optimization. Although these methods extend IB to multivariate settings and even show that certain PID quantities, such as redundancy, can be characterized via bottlenecks, none explicitly incorporate a synergy-specific information term into the IB formulation. Our approach achieves this, and we will demonstrate that it not only results in more consistent compression behavior but also offers theoretical advantages.

**Synergy.** Synergy characterizes the additional information obtained by evaluating variables collectively rather than individually, quantifying how features interact to reduce uncertainty about a target. The characteristics of this relationship can be illustrated by means of the XOR function. Consider two binary string variables,  $X_1$  and  $X_2$ , with  $Z$  being their XOR output. In this scenario,  $X_1$  and  $Z$ , as well as  $X_2$  and  $Z$ , are uncorrelated ( $I(X_1; Z) = I(X_2; Z) = 0$ ), but together,  $X_1$  and  $X_2$  fully describe  $Z$  ( $I(X_1, X_2; Z) = H(Z)$ ) (Guyon & Elisseeff, 2003; Williams & Beer, 2010). While the concept is intuitive, its formalization has proven challenging, leading to multiple proposed measures. Early work by Williams & Beer (2010) introduced PID, which decomposes MI into unique, redundant, and synergistic components. However, the number of terms in this decomposition equals the  $n - 1$ 'th Dedekind number, where  $n$  is the number of features. This number is impractically large: a system with nine variables would require approximately  $5 \times 10^{22}$  terms, while for ten variables, the Dedekind number remains unknown. Moreover, estimating these terms is subject to convergence issues and size limitations (Makkeh et al., 2018; 2019; Pakman et al., 2021). While Varley & Hoel (2022) reduced the number of investigable quantities by averaging contributions of layers in the PID lattice, with different layers representing different levels of redundancy or synergy, calculations remained too complex for applications involving more than a few features. Alternative measures such as O-information (Rosas et al., 2019), correlational importance (Nirenberg & Latham, 2003), and synergistic MI (Griffith & Koch, 2014) can estimate the synergy or redundancy of large sets of variables, but fail to reveal whether a specific feature interacts synergistically or redundantly. We resolve these issues by averaging the interaction information of each feature with those remaining:

$$\text{Syn}(\mathcal{X} \rightarrow Y) = I(\mathcal{X}; Y) - \frac{1}{N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Y) + I(X^i; Y)) \quad (2)$$

where  $\mathcal{X}^{-i} = \mathcal{X} \setminus \{X^i\}$ . This formulation captures how features collectively reduce uncertainty about the target  $Y$ , while maintaining computational feasibility by avoiding the exponential explosion of subset calculations (Westphal et al., 2025).

### 3 THE GENERALIZED INFORMATION BOTTLENECK

We now introduce the GIB, by first showing that, given two functions with identical mutual information (MI) with noisy training data, the function exhibiting higher synergy achieves tighter generalization bounds. This result motivates the principle that synergistic functions generalize better than non-synergistic ones, and thus learning should favor synergy. At the same time, we emphasize that functions must also be correct. To capture this, we introduce a distribution that prioritizes accurate predictions. The final formulation of the GIB therefore maximizes the synergistic contribution of the inputs in describing this distribution.

#### 3.1 SYNERGY AND GENERALIZATION

In this subsection, we formally establish the connection between synergy and generalization. We begin by presenting theoretical results supported by experiments on synthetic data, and then extend the discussion to empirical findings on ResNets.

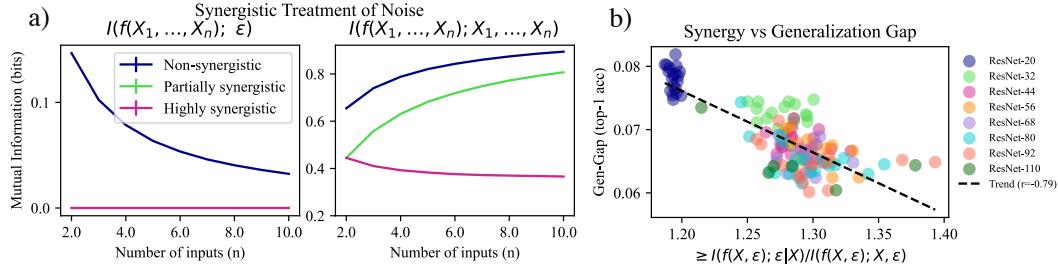


Figure 3: Synergistic processing of noise enhances generalization. (a) Controlled synthetic experiment demonstrating how synergy affects information flow (see Appendix E.1 for details). Three functions of increasing synergy process binary inputs with noise: non-synergistic (blue), partially synergistic (green), and highly synergistic (magenta). Left:  $I(f(X, \varepsilon); \varepsilon)$ : more synergistic functions have lower dependence on noise. Right:  $I(f(X, \varepsilon); X)$  - we observe that synergistic functions have lower MI with the input. (b) Empirical validation on CIFAR-10 using ResNets of varying depths (see Appendix E.2). We quantify synergistic interactions between inputs and noise as  $I(f(X, \varepsilon); \varepsilon|X)/I(f(X, \varepsilon); X, \varepsilon)$ . Higher synergy correlates with smaller generalization gaps.

### 3.1.1 THEORETICAL EVIDENCE

Let us first suppose we have some noise  $\varepsilon$  that can be considered independent of our input data  $\mathcal{X}$ . Now consider two functions  $s^-$  and  $s^+$ . If  $s^+$  combines the independent components of its arguments in a more synergistic manner than  $s^-$ , by definition we have:

$$\begin{aligned} I(s^+(\mathcal{X}, \varepsilon); \mathcal{X}, \varepsilon) - I(s^+(\mathcal{X}, \varepsilon); \varepsilon) - I(s^+(\mathcal{X}, \varepsilon); \mathcal{X}) > \\ I(s^-(\mathcal{X}, \varepsilon); \mathcal{X}, \varepsilon) - I(s^-(\mathcal{X}, \varepsilon); \varepsilon) - I(s^-(\mathcal{X}, \varepsilon); \mathcal{X}). \end{aligned} \quad (3)$$

If we now assume that  $I(s^+(\mathcal{X}, \varepsilon); \mathcal{X}, \varepsilon) = I(s^-(\mathcal{X}, \varepsilon); \mathcal{X}, \varepsilon)$  (which can crudely be thought of as approximately equal train accuracies) then it must be true that:

$$I(s^+(\mathcal{X}, \varepsilon); \varepsilon) + I(s^+(\mathcal{X}, \varepsilon); \mathcal{X}) < I(s^-(\mathcal{X}, \varepsilon); \varepsilon) + I(s^-(\mathcal{X}, \varepsilon); \mathcal{X}) \quad (4)$$

where  $I(s(\mathcal{X}, \varepsilon); \varepsilon)$  represents the MI between the output of a function and the noise, while  $I(s(\mathcal{X}, \varepsilon); \mathcal{X})$  describes the information shared between the output and uncorrupted input. In Figure 3(a) we analyze the implications of Equation 4 via synthetic data. We show that more synergistic functions for the same complexity of input and output have lower values of both  $I(s(\mathcal{X}, \varepsilon); \varepsilon)$  and  $I(s(\mathcal{X}, \varepsilon); \mathcal{X})$ . This is favorable as both of these terms are known to be inversely related to generalization capabilities, as discussed below.

**How  $I(s(\mathcal{X}, \varepsilon); \mathcal{X})$  Impedes Generalizability.** This quantity can be re-written as the complexity term of the IB, reducing its value has repeatedly been shown to be related to compression and generalization (Tishby et al., 1999; Shwartz-Ziv & Tishby, 2017). High values of this term ensure a latent representation that has memorized irrelevant and redundant information in the input. Furthermore, recent work has formally related this quantity to generalization bounds (Kawaguchi et al., 2023).

**How  $I(s(\mathcal{X}, \varepsilon); \varepsilon)$  Impedes Generalizability.** The relationship between noise sensitivity and generalization is fundamentally tied to function smoothness. Most generalization bounds require that the learned function be *Lipschitz smooth*, meaning there exists a constant  $L$  such that  $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$  for all inputs. This constraint ensures the function's output changes at most proportionally to input perturbations. When a function has high mutual information with noise  $I(s(\mathcal{X}, \varepsilon); \varepsilon)$ , it indicates the output varies significantly with small noise perturbations, implying a large Lipschitz constant. As shown by Bartlett et al. (2017) and Neyshabur et al. (2017), generalization bounds scale with the Lipschitz constant of neural networks, which can be bounded by the product of layer-wise spectral norms. Therefore, functions with lower  $I(s(\mathcal{X}, \varepsilon); \varepsilon)$  exhibit smaller Lipschitz constants and tighter generalization bounds, explaining why synergistic functions that minimize noise sensitivity achieve superior generalization.

270  
271

## 3.1.2 EMPIRICAL EVIDENCE

272  
273  
274  
275  
276  
277  
278  
279  
280

To empirically validate our theoretical findings, we conducted experiments examining how synergistic processing of noise affects generalization in deep NNs. We trained ResNet models of varying depths (20, 32, 44, 56, 68, 80, 92, and 110 layers) on CIFAR-10 with standard data augmentations. To quantify synergy with augmentation noise, we developed a novel teacher-student framework: a teacher model trained with augmentations (random crops and horizontal flips) teaches a student model to predict its outputs from non-augmented inputs. The cross-entropy loss achieved by the student provides a maximal upper bound for the proportion of information between inputs and outputs that cannot be explained without considering the interaction of noise and features, formally:  $I(f(X, \varepsilon); \varepsilon|X)/I(f(X, \varepsilon); X, \varepsilon)$ .

281  
282  
283  
284  
285  
286  
287  
288  
289  
290

Our results, shown in Figure 3(b), reveal a strong negative correlation (Pearson  $r = -0.79$ ,  $p < 0.001$ ) between this synergy measure and generalization performance across all model configurations. Models with higher synergy (those whose predictions depend more on the interaction between image content and augmentation patterns) consistently achieve smaller generalization gaps. This confirms our theoretical prediction: synergistic processing of augmentation noise, rather than treating it as independent corruption, enables models to extract more robust features that generalize better to clean test images. Full experimental details are provided in Appendix E.2. Considering that to synergistically process noise, we must synergistically process the features, we design our GIB principle based on measures of feature synergy.

291  
292

## 3.2 FORMULATING THE GIB PRINCIPLE

293  
294  
295  
296  
297  
298

During the last section, we argued that synergistic functions generalize better than their non-synergistic counterparts. Consequently, we argue that when learning, a deep network should aim to maximize the synergy of the inputs to produce the outputs. However, this is not a strict enough condition, because there are many different synergistic functions, most of which are irrelevant to the task at hand. We instead want to measure how synergistically our inputs combine to give the *correct* outputs.

299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312

To facilitate this, we take the following two steps. First, our prediction term will solely measure the MI between our predictions and targets. Second, our complexity term will be a function of a new distribution that describes the co-occurrences of  $Z$  with  $Y$ . The exact definition of  $Q$  is based on PMI-based reweighting, i.e., weighting samples by the likelihood ratio between the joint distribution and the product of marginals:  $Q(Z, Y) = \frac{P(Z, Y)}{P(Z)P(Y)}$ . This reweighting scheme emphasizes patterns where  $Z$  and  $Y$  co-occur more frequently than would be expected under independence, effectively highlighting the meaningful dependencies between our learned representations and the target outputs. PMI has proven effective in capturing meaningful associations in numerous ML contexts: it underlies word2vec’s implicit matrix factorization (Levy & Goldberg, 2014), drives contrastive learning objectives (van den Oord et al., 2018), and measures feature relevance in interpretable ML (Bouma, 2009). By sampling from the distribution  $Q(Z, Y) = \frac{P(Z, Y)}{P(Z)P(Y)}$  we obtain the random variable  $Q$ . Combining these steps with how we earlier defined synergy, we get the following formulation of the GIB:

313  
314  
315  
316  
317  
318

$$\mathcal{L}_{\text{GIB}} = \max_{p(Z|X)} \left[ \underbrace{I(Z; Y)}_{\text{prediction term}} - \underbrace{\frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Q) + I(X^i; Q))}_{\text{complexity term}} \right] \quad (5)$$

319  
320  
321  
322  
323

The prediction term (blue)  $I(Z; Y)$  measures the mutual information between the model outputs  $Z$  and the labels  $Y$ , capturing how well the predictions align with the true targets. The complexity term (red)  $\frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Q) + I(X^i; Q))$  inversely quantifies the average information obtainable from individual features or their complements about the PMI-reweighted distribution  $Q(Z, Y)$ , which emphasizes correct predictions. By maximizing their difference, GIB measures information dynamics that emerge only from collective feature

324 interactions, which our analysis also indicates leads to improved generalization. On the other  
 325 hand, measuring synergy can be computationally demanding, as we discuss in Appendix G.  
 326

## 327 4 RELATING THE GIB TO THE IB

329 In this section, we first prove that under a simple assumption (i.e., perfect estimation) the IB  
 330 is a lower bound of our GIB. Finally, we discuss how the GIB solves longstanding IB issues.  
 331

332 **Theorem 1.** *If we assume perfect training accuracy and therefore  $Q = Z = Y$ , then the  
 333 original IB objective is upper bounded by our GIB:*

$$334 I(\mathcal{T}; Y) - \beta I(\mathcal{X}; \mathcal{T}) \leq I(Z; Y) - \frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Q) + I(X^i; Q)) \quad (6)$$

337 The proof is provided in Appendix B. This result demonstrates that the GIB provides  
 338 an upper bound on the IB objective. Consequently, as we optimize the traditional IB to  
 339 find sufficient statistics, we simultaneously optimize our GIB objective, ensuring that our  
 340 approach remains compatible with the theoretical foundations of the IB. For instance, in  
 341 Appendix C we prove the GIB discovers sufficient statistics.

342 This new formulation overcomes two main limitations of the original IB. First, the partition  
 343 across subsets of features combined with the PMI definition of  $Q(Z, Y)$  protects the compres-  
 344 sion term from becoming infinite. In Appendix D we prove that the GIB is only infinite under  
 345 interpretable circumstances. Second, and more fundamentally, our formulation explicitly  
 346 considers over-reliance on individual features, as explained in Section 3.1. IB optimizes  
 347 the total information flow between inputs and outputs through the latent representation  $\mathcal{T}$   
 348 without considering how features interact. In contrast, GIB explicitly models how inputs  
 349 combine to form the latent representation, distinguishing between different types of feature  
 350 interactions. This is evident in the complexity terms: IB’s  $\beta^{-1} I(\mathcal{X}; \mathcal{T})$  aggregates all in-  
 351 formation equally, while GIB’s synergistic decomposition  $\frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Q) + I(X^i; Q))$   
 352 penalizes the information contained in individual features. Consequently, IB compresses  
 353 indiscriminately, whereas GIB selectively preserves long-range feature combinations; the  
 354 synergistic patterns we have shown lead to better generalization.

## 355 5 EXPERIMENTAL CASE STUDIES

358 This section presents experimental evidence demonstrating GIB’s advantages over standard  
 359 IB across diverse settings. We show that GIB provides more consistent and interpretable  
 360 information dynamics throughout training, successfully capturing compression phases where  
 361 standard IB fails (refer to Footnote 2 on reading information planes). Additionally, we  
 362 demonstrate that GIB’s complexity term serves as a direct indicator of adversarial vulnera-  
 363 bility, providing quantitative insights into model robustness that standard IB cannot capture.  
 364 As stated in Section 2, all MIs will be estimated using binning. However, in Appendix F, we  
 365 reproduce our results with a different method of MI estimation.

### 366 5.1 INFORMATION DYNAMICS OF MLPs LEARNING SIMPLE FUNCTIONS

368 We examine NNs learning five mathematical functions of increasing complexity: addition,  
 369 multiplication, and three symmetric polynomials labelled  $f_1$ ,  $f_2$  and  $f_3$  (polynomials in  
 370 which all arguments are subjected to the same operations). Full experimental details are in  
 371 Appendix E.3.

372 The information plane dynamics in Figure 4 show clear differences between the standard  
 373 IB and GIB formulations. For the GIB, we observe compression phases, characterized  
 374 by leftward movement during training, across all five functions and random seeds. The  
 375 trajectories initially move upward and rightward as networks fit the training data, then shift  
 376 leftward as training progresses. The standard IB displays more variable behavior, without  
 377 real indication of compression, despite the strong generalization capabilities of these networks.

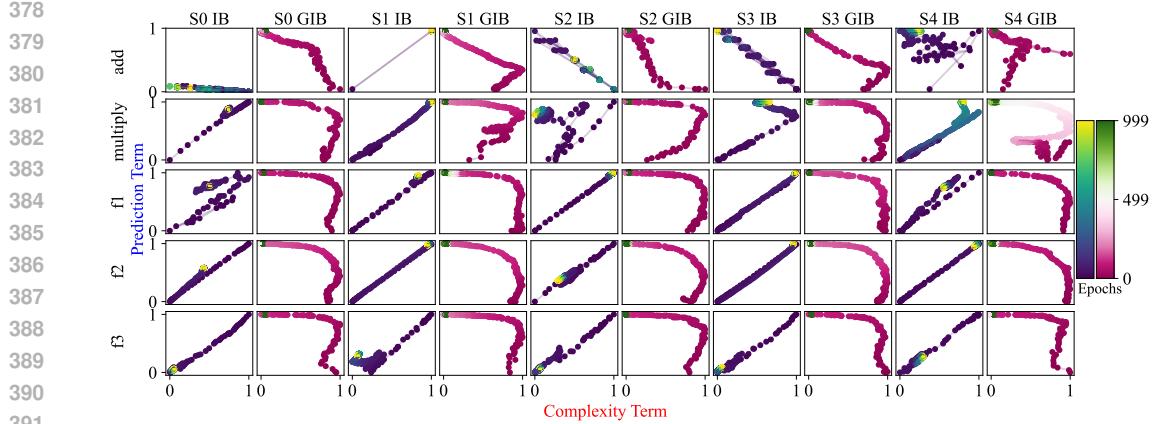


Figure 4: Information plane dynamics for NNs learning simple mathematical functions. Comparison of standard IB versus GIB across five functions (rows) and five random seeds. Functions include basic arithmetic and symmetric polynomials. GIB consistently shows compression phases (leftward movement), while standard IB exhibits varied behaviors. See Appendix E.3 for experimental details.

## 5.2 INFORMATION DYNAMICS OF RESNETS

We analyze information dynamics in residual networks (ResNets) of varying depths (20, 56, 80, 110 layers) trained on CIFAR-10. For the standard IB, we compute MI using the 10-dimensional output layer directly. For GIB, due to the need to compute feature-wise decompositions on the high-dimensional input space (3072 dimensions), we first apply Kernel PCA to reduce the pixel space to 50 principal components before computing MI, as explained in more detail in Appendix E.4 (Turk & Pentland, 1991). For more details as to why we chose 50 components, see Appendix H.

Figure 5 displays information plane trajectories for ResNets trained on CIFAR-10. The GIB formulation shows consistent compression and fitting for all network depths and random seeds, though the dynamics vary with architecture size. In smaller networks (ResNet-20), trajectories show a general trend of increasing prediction term while the complexity term decreases throughout training. Larger networks (ResNet-56 and above) begin to show phase structure.

The standard IB presents markedly different dynamics. Rather than showing clear phases, IB trajectories remain largely clustered with minimal compression across epochs. The absence of compression phases in standard IB holds across all tested architectures, confirming previous observations that *ReLU* networks fail to exhibit expected IB behavior (Saxe et al., 2018).

## 5.3 INFORMATION DYNAMICS OF TRANSFORMERS CLASSIFYING NEWS HEADLINES

We examine BERT-base fine-tuned on AG News text classification, comparing standard fine-tuning with a novel “unlearning” initialization strategy. In this case, unlearning involves training on random labels to remove biases from the model. For the standard IB, we again set  $\mathcal{T}$  as the final layer representation for use in MI calculations. For GIB, we apply our feature-wise synergy decomposition to the full set of inputs. Full experimental details are in Appendix E.5.

The standard fine-tuning approach shown in row 1 of Figure 6 produces highly variable trajectories that begin near the center of the information plane, indicating that pre-trained BERT representations contain substantial pre-training biases. This prompted us to unlearn, where we train on random labels. The unlearning intervention dramatically alters these dynamics. After 3 epochs of random label training, models consistently start from the bottom-right corner of the information plane, as shown in row 2. From this reset position, both IB and GIB show more coherent learning trajectories during subsequent fine-tuning. This

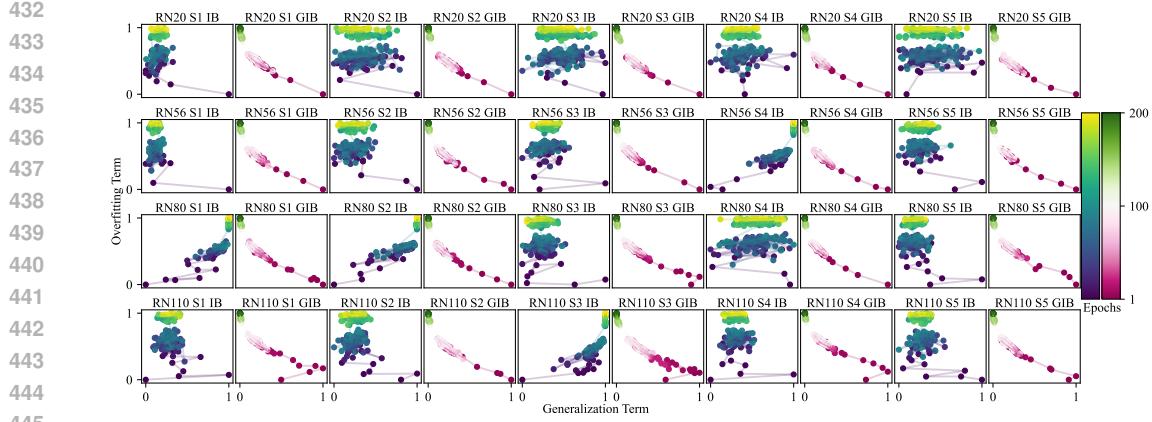


Figure 5: Information plane dynamics for ResNets of varying depths trained on CIFAR-10. Comparison across four network depths and five random seeds. GIB consistently exhibits compression phases, while standard IB shows limited or no compression. See Appendix E.4 for details.

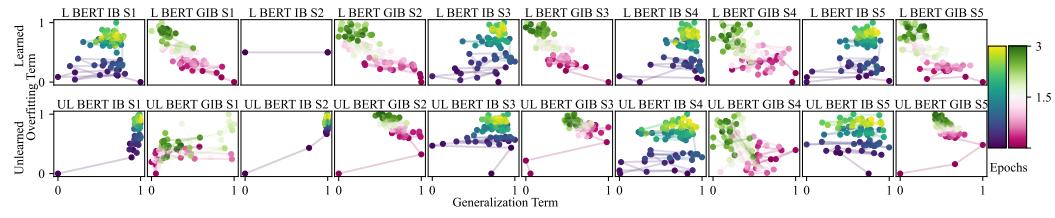


Figure 6: Information plane dynamics for BERT fine-tuned on AG News. Comparison of standard fine-tuning (top) versus unlearning + fine-tuning (bottom). The unlearning procedure repositions models to a more favorable initialization point for subsequent learning. See Appendix E.5 for details.

reveals how studying these information-planes can be used for diagnostics and interpretation.

#### 5.4 ADVERSARIAL ROBUSTNESS

We investigate how adversarial perturbations affect information dynamics by training NNs with *tanh* activations on MNIST under Fast Gradient Sign Method (FGSM) attacks of varying strength. Full details are in Appendix E.6. Figure 7(a) illustrates the effect of adversarial attacks on learning dynamics. Networks trained under weak attacks ( $\epsilon = 0.01$ ) exhibit normal convergence, whereas strong attacks ( $\epsilon = 1.0$ ) substantially hinder the learning process. The information-theoretic analysis in Figure 7(b) exposes a critical difference between standard IB and our GIB formulation. The GIB’s complexity term faithfully reflects the degradation in generalization: values remain high for  $\epsilon = 1.0$  (poor generalization), while decreasing rapidly when proper training occurs. In contrast, the standard IB’s complexity term shows minimal differentiation between attack strengths.

## 6 CONCLUSION

In this work, we have introduced the Generalized Information Bottleneck (GIB), a principled reformulation of the IB that explicitly accounts for synergistic interactions between features. Our theoretical and empirical analysis demonstrated that synergistic functions achieve better generalization, providing a fundamental justification for why deep networks should learn representations that combine inputs synergistically rather than processing them independently. The GIB framework addresses several longstanding limitations of the original IB. First, we proposed a PMI-based reweighting scheme  $Q(Z, Y)$  that protects the compression term from becoming infinite during training. Second, we have introduced a feature-wise synergy decomposition, explicitly penalizing representations that rely too heavily on individual

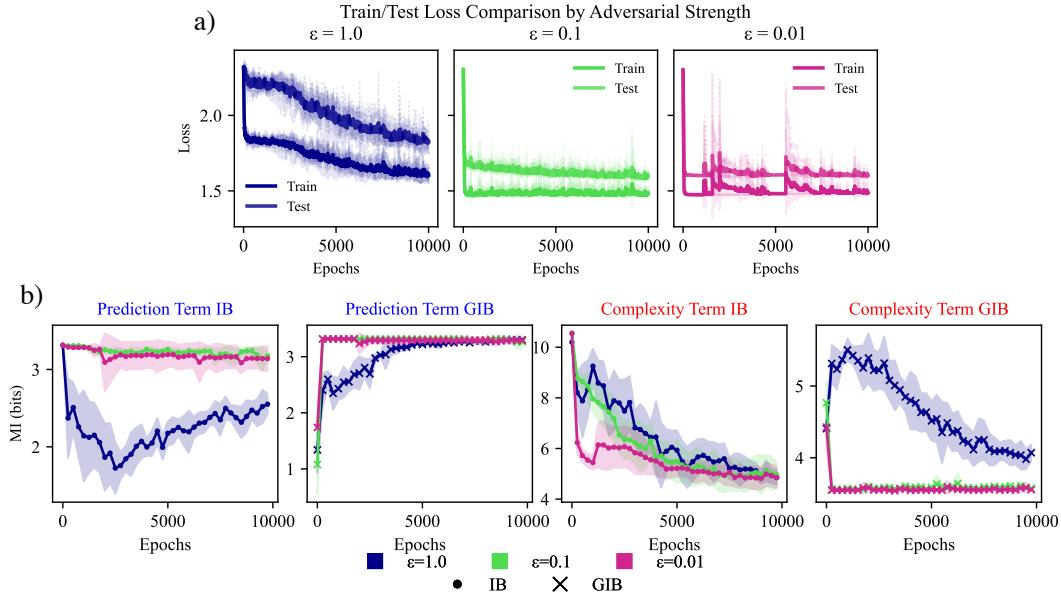


Figure 7: Information dynamics under adversarial attacks on MNIST. (a) Training dynamics for three FGSM attack strengths. (b) Information-theoretic analysis comparing IB versus GIB. GIB’s complexity term correctly reflects degradation in generalization, while standard IB fails to differentiate between attack strengths. See Appendix E.6 for details.

features or simple feature combinations. This ensures that our formulation highlights when networks learn patterns that emerge solely through the collective processing of multiple inputs, a distinction that is crucial for robust generalization. Our experimental results across diverse architectures demonstrate that GIB provides a more complete picture of how deep networks process information. The GIB framework opens new possibilities for both understanding and improving deep learning systems.

## REFERENCES

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2897–2905, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations (ICLR’17)*, 2017.

Francesco Alesiani, Shujian Yu, and Xi Yu. Gated information bottleneck for generalization in sequential environments. *Knowledge and Information Systems*, 65(2):683–705, 2023.

Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS’17)*, 2017.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning (ICML’18)*, 2018.

Anthony J Bell. Co-information lattice. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Source Separation*, pp. 921–926, 2003.

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the 21st Biennial Gesellschaft für Semantische Verarbeitung Natürlicher Sprache Conference (GSCL’09)*, 2009.

540 Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience,  
 541 1991.

542

543 Nir Friedman, Ori Mosenzon, Noam Slonim, and Naftali Tishby. Multivariate information  
 544 bottleneck. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence  
 545 (UAI'01)*, 2001.

546 Bernhard C. Geiger. On information plane analyses of neural network classifiers – a review.  
 547 *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):7039–7051, 2022.

548

549 Ziv Goldfeld, Ewout van den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian  
 550 Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks.  
 551 *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

552

553 Virgil Griffith and Christof Koch. Quantifying synergistic mutual information. In Mikhail  
 554 Prokopenko (ed.), *Guided Self-Organization: Inception*, pp. 159–190. Springer, 2014.

555

556 Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal  
 557 of Machine Learning Research*, 3:1157–1182, 2003.

558

559 Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Differential entropy estimation via score  
 560 matching. In *International Conference on Artificial Intelligence and Statistics (AISTAT'17)*,  
 2017.

561

562 Robin A. A. Ince, Bruno L. Giordano, Christoph Kayser, Guillaume A. Rousselet, Joachim  
 563 Gross, and Philippe G. Schyns. A statistical framework for neuroimaging data analysis  
 564 based on mutual information estimated via a Gaussian copula. *Human Brain Mapping*, 38  
 (3):1541–1573, 2017.

565

566 Kirthevasan Kandasamy, Akshay Krishnamurthy, and Barnabas Poczos. Nonparametric  
 567 von Mises estimators for entropies, divergences and mutual informations. In *Advances in  
 568 Neural Information Processing Systems (NeurIPS'15)*, 2015.

569

570 Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck  
 571 help deep learning? In *Proceedings of the 40th International Conference on Machine  
 572 Learning (ICML'23)*, 2023.

573

574 Artemy Kolchinsky. Partial information decomposition: Redundancy as information bottle-  
 575 neck. *Entropy*, 26(7):1–20, 2024.

576

577 Artemy Kolchinsky, Brendan D. Tracey, and David H. Wolpert. Nonlinear information  
 578 bottleneck. *Entropy*, 21(12):1181, 2019.

579

580 L. F. Kozachenko and N. N. Leonenko. The sample estimate of the entropy of a random  
 581 vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

582

583 Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information.  
 584 *Physical Review E*, 69(6):066138, 2004.

585

586 Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In  
 587 *Advances in Neural Information Processing Systems (NeurIPS'14)*, pp. 2177–2185, 2014.

588

589 Chao Ma, Chen Zhang, and Liwei Wang. Revisiting Hilbert-Schmidt information bottle-  
 590 neck for adversarial robustness. In *Advances in Neural Information Processing Systems  
 591 (NeurIPS'21)*, 2021.

592

593 Abdullah Makkeh, Dirk Oliver Theis, and Raul Vicente. BROJA-2PID: A robust estimator  
 594 for bivariate partial information decomposition. *Entropy*, 20(4):271, 2018.

595

596 Abdullah Makkeh, Armin J. Gutknecht, and Michael Wibral. MAXENT3D\_PID: An  
 597 estimator for the maximum-entropy trivariate partial information decomposition. *Entropy*,  
 598 21(9):862, 2019.

594 Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring  
 595 generalization in deep learning. In *Advances in Neural Information Processing Systems*  
 596 (*NeurIPS'17*), 2017.

597 Sheila Nirenberg and Peter E. Latham. Decoding neuronal spike trains: How important are  
 598 correlations? *Proceedings of the National Academy of Sciences of the United States of*  
 599 *America*, 100(12):7348–7353, 2003.

600 Ari Pakman, Amir Nejatbakhsh, Julien Terglane, Cengiz Pehlevan, and Liam Paninski. Esti-  
 601 mating the unique information of continuous variables. In *Advances in Neural Information*  
 602 *Processing Systems (NeurIPS'21)*, 2021.

603 Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of*  
 604 *Mathematical Statistics*, 33(3):1065–1076, 1962.

605 Fernando E. Rosas, Pedro A. M. Mediano, Michael Gastpar, and Henrik J. Jensen. Quantify-  
 606 ing high-order interdependencies via multivariate extensions of the mutual information.  
 607 *Physical Review E*, 100(3):032305, 2019.

608 Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Bren-  
 609 dan D. Tracey, and David D. Cox. On the information bottleneck theory of deep learning.  
 610 *Journal of Statistical Mechanics: Theory and Experiment*, 2018.

611 Andrew M. Saxe, Yamini Bansal, Joel Dapello, and Madhu Advani. On the information  
 612 bottleneck theory of deep learning. *Proceedings of the 7th International Conference on*  
 613 *Learning Representations (ICLR'19)*, 2019.

614 Elad Schneidman, William Bialek, and Michael J Berry. Synergy, redundancy, and indepen-  
 615 dence in population codes. *Journal of Neuroscience*, 23(37):11539–11553, 2003.

616 David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*.  
 617 Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 1992.

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

620 Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method.  
 621 In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and*  
 622 *Computing*, 1999.

623 Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Proceedings of*  
 624 *the 4th IEEE Computer Society Conference on Computer Vision and Pattern Recognition*  
 625 (*CVPR'91*), 1991.

626 Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive  
 627 predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

628 Thomas F. Varley and Erik Hoel. Emergence as the conversion of information: A unifying  
 629 theory. *Philosophical Transactions of the Royal Society A*, 380(2227):20210150, 2022.

630 Zifeng Wang, Tong Jian, Aria Masoomi, Stratis Ioannidis, and Jennifer G. Dy. Revisiting  
 631 hilbert-schmidt information bottleneck for adversarial robustness. In *Advances in Neural*  
 632 *Information Processing Systems*, volume 34, pp. 586–597, 2021.

633 Charles Westphal, Stephen Hailes, and Mirco Musolesi. Partial information decomposition  
 634 for data interpretability and feature selection. In *Proceedings of the 28th International*  
 635 *Conference on Artificial Intelligence and Statistics (AISTATS'25)*, 2025.

636 Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information.  
 637 In *arXiv preprint arXiv:1004.2515*, 2010.

638 Shujian Yu, Luis Gonzalo Sanchez Giraldo, Robert Jenssen, and Jose C. Principe. Multivariate  
 639 Extension of Matrix-Based Rényi's  $\alpha$ -Order Entropy Functional. *IEEE Transactions on*  
 640 *Pattern Analysis and Machine Intelligence*, 42(11):2960–2966, 2020.

648 Shujian Yu, Francesco Alesiani, Xi Yu, Robert Jenssen, and Jose C. Principe. Measuring  
649 dependence with matrix-based entropy functional. In *Proceedings of the 35th AAAI*  
650 *Conference on Artificial Intelligence (AAAI'21)*, 2021a.  
651  
652 Xi Yu, Shujian Yu, and Jose C. Principe. Deep deterministic information bottleneck with  
653 matrix-based entropy functional. In *Proceedings of the 46th IEEE International Conference*  
654 *on Acoustics, Speech and Signal Processing (ICASSP'21)*, 2021b.  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

702 A NOTATION TABLE  
703  
704  
705  
706  
707Table 1 contains all the mathematical notation used in this paper.  
708  
709  
710  
711  
712713 Table 1: Summary of Notational Conventions.  
714

Symbol	Description
$\mathcal{X}$	Set of input random variables (features)
$X^i$	$i$ -th individual input feature
$\mathcal{X}^{-i}$	All features except the $i$ -th feature: $\mathcal{X} \setminus \{X^i\}$
$Y$	Target random variable (labels)
$Z$	Output/prediction random variable
$\mathcal{T}$	Hidden representation/latent space
$\varepsilon$	Noise random variable
$N$	Number of input features
$\beta$	Trade-off parameter in IB formulation
$I(\cdot; \cdot)$	Mutual information
$H(\cdot)$	Entropy
$H(\cdot   \cdot)$	Conditional entropy
$P(\cdot)$	Probability distribution
$Q(Z, Y)$	PMI-based reweighted distribution
$\hat{Q}$	Variable sampled from $Q(Z, Y)$
$\text{Syn}(\cdot)$	Synergy measure
$s^+, s^-$	More/less synergistic functions
IB	Information Bottleneck
GIB	Generalized Information Bottleneck
MI	Mutual Information
NN	Neural Network
PMI	Point-wise Mutual Information
PID	Partial Information Decomposition
FGSM	Fast Gradient Sign Method

738  
739  
740  
741  
742  
743  
744 B PROOF OF THEOREM 1  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755In this section we prove Theorem 1. To do this, we assume that we have perfect training performance and therefore  $Q(Y, Z) = Y = Z$ . We also assume the predictor is deterministic given its input (as in a standard feed-forward network), hence  $H(Z|\mathcal{X}) = 0$  and therefore

$$I(\mathcal{X}; Z) = H(Z). \quad (7)$$

756 *Proof.* Given this, our GIB formulation becomes:  
 757

$$758 \quad \mathcal{L}_{\text{GIB}} = I(Z; Y) - \frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Q) + I(X^i; Q)) \quad (8)$$

$$761 \quad = H(Y) - \frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Z) + I(X^i; Z)) \quad (\text{since } Z = Y \text{ and } Q = Z = Y) \quad (9)$$

$$764 \quad \geq I(\mathcal{X}; Y) - \frac{1}{2\beta N} \sum_{i=1}^N (I(\mathcal{X}^{-i}; Z) + I(X^i; Z)) \quad (\text{since } I(\mathcal{X}; Y) \leq H(Y)) \quad (10)$$

$$767 \quad \geq I(\mathcal{X}; Y) - \frac{1}{2\beta N} \sum_{i=1}^N (H(Z) + H(Z)) \quad (\text{by } I(A; B) \leq H(B)) \quad (11)$$

$$770 \quad = I(\mathcal{X}; Y) - \frac{1}{2\beta N} \cdot N \cdot 2H(Z) \quad (12)$$

$$772 \quad = I(\mathcal{X}; Y) - \frac{1}{\beta} H(Z) \quad (13)$$

$$774 \quad = I(\mathcal{X}; Y) - \frac{1}{\beta} I(\mathcal{X}; Z) \quad (\text{by Eq. 7, } H(Z|\mathcal{X}) = 0) \quad (14)$$

$$776 \quad \geq I(\mathcal{X}; Y) - \frac{1}{\beta} I(\mathcal{X}; \mathcal{T}) \quad (\text{by data processing, } \mathcal{X} \rightarrow \mathcal{T} \rightarrow Z) \quad (15)$$

$$778 \quad \geq I(\mathcal{T}; Y) - \frac{1}{\beta} I(\mathcal{X}; \mathcal{T}) \quad (\text{by data processing, } \mathcal{X} \rightarrow \mathcal{T} \rightarrow Y). \quad (16)$$

780 We obtain:  
 781

$$782 \quad I(\mathcal{T}; Y) - \frac{1}{\beta} I(\mathcal{X}; \mathcal{T}) \leq \mathcal{L}_{\text{GIB}}$$

783 which shows that the original IB objective is upper bounded by the proposed GIB under the  
 784 stated assumption.  $\square$   
 785

## 786 C GIB AND SUFFICIENT STATISTICS

788 In this section, we show that when the first term in the GIB objective is  $I(Z; Y)$ , the GIB  
 789 recovers sufficient statistics in the limit  $\beta \rightarrow \infty$ .  
 790

791 **Theorem 2.** For  $\beta \in (0, \infty]$ , consider the GIB functional

$$793 \quad \mathcal{J}_\beta(P_{Z|X}) := I(Z; Y) - \frac{1}{\beta 2N} \sum_{i=1}^N (I(X^{-i}; Q) + I(X^i; Q)).$$

795 Then at  $\beta = \infty$ ,

$$797 \quad \sup_{P_{Z|X}} \mathcal{J}_\infty = \sup_{P_{Z|X}} I(Z; Y) \leq H(Y).$$

798 Moreover, equality holds (i.e.,  $\sup_{P_{Z|X}} I(Z; Y) = H(Y)$ ) if and only if  $Y$  is a deterministic  
 799 function of  $Z$ , equivalently  $H(Y | Z) = 0$ . In this case,  $Z$  is a sufficient statistic.  
 800

801 *Proof.* At  $\beta = \infty$ , the penalty term vanishes and therefore  $\mathcal{J}_\infty = I(Z; Y)$ . By the elementary  
 802 information inequality  $I(Z; Y) \leq H(Y)$  for all encoders  $P_{Z|X}$ , we have  
 803

$$804 \quad \sup_{P_{Z|X}} \mathcal{J}_\infty = \sup_{P_{Z|X}} I(Z; Y) \leq H(Y).$$

806 If for some encoder we have  $H(Y | Z) = 0$ , i.e.  $Y$  is a deterministic function of  $Z$ , then  
 807

$$808 \quad I(Z; Y) = H(Y) - H(Y | Z) = H(Y),$$

809 and the upper bound is achieved.

810 Conversely, if  $I(Z; Y) = H(Y)$  for some encoder, then

$$811 \quad H(Y | Z) = H(Y) - I(Z; Y) = 0,$$

812 so  $Y$  must be a deterministic function of  $Z$ .

814 Thus, at  $\beta = \infty$ , maximising  $\mathcal{J}_\infty$  recovers precisely the set of encoders for which  $Z$  is a  
815 sufficient statistic for  $Y$ .  $\square$

## 817 D WHEN GIB ENCOUNTERS INFINITY

819 In this section, we analyze the conditions under which our GIB formulation yields infinite  
820 values and show that, unlike standard IB, these infinities have meaningful interpretations.

821 **Theorem 3.** *Let  $\mathcal{X} = (X^1, \dots, X^N)$  be a feature vector and let  $Y$  be a continuous random  
822 variable. Assume that the representation  $Z$  achieves perfect prediction of  $Y$  in the sense that  
823  $I(Z; Y) = \infty$ . Assume further that no individual feature  $X^i$  nor its complement  $\mathcal{X}^{-i}$  alone  
824 yields an infinite mutual information with the PMI variable  $Q$ , in the sense that*

$$825 \quad \frac{1}{N} \sum_{i=1}^N \left( I(\mathcal{X}^{-i}; Q) + I(X^i; Q) \right) < \infty,$$

826 where  $Q$  is the point-wise mutual information random variable. Then  $\mathcal{L}_{\text{GIB}} = \infty$ .

827 *Proof.* By definition,

$$828 \quad \mathcal{L}_{\text{GIB}} = I(Z; Y) - \frac{1}{2\beta N} \sum_{i=1}^N \left( I(\mathcal{X}^{-i}; Q) + I(X^i; Q) \right).$$

829 By assumption,  $I(Z; Y) = \infty$ , while the sum is finite. Hence, in the extended real line,

$$830 \quad \mathcal{L}_{\text{GIB}} = \infty - (\text{finite}) = \infty. \quad \square$$

831 **Interpretation of Infinities in GIB.** In contrast to the standard IB, where infinities arise  
832 as technical artifacts of deterministic mappings between continuous variables, the infinities  
833 in the GIB functional admit a more structural interpretation.

834 When  $\mathcal{L}_{\text{GIB}} = \infty$ , we are in a regime of *perfect synergy*: the output  $Y$  can be recovered from  
835 the full feature set (via  $Z$ ), yet no individual coordinate or coordinate subset carries enough  
836 information to reconstruct the relevant PMI signal. In other words, all features are jointly  
837 essential. This corresponds to the ideal synergistic regime.

838 Conversely, if some feature  $X^i$  or complement  $\mathcal{X}^{-i}$  can alone perfectly determine  $Y$ , then  
839 the corresponding term  $I(X^i; Q)$  or  $I(\mathcal{X}^{-i}; Q)$  becomes infinite, making the entire penalty  
840 infinite. In this case, the GIB takes the form

$$841 \quad \infty - \infty,$$

842 which is *indeterminate*. This suggests that prediction is not synergistic, as the output can  
843 be reconstructed from a sufficiently informative subset of the features.

844 Thus, while both IB and GIB encounter infinities in the continuous setting, the GIB admits a  
845 meaningful interpretation: an infinite value corresponds precisely to perfect synergy, whereas  
846 the indeterminate case reflects the absence of synergy.

## 855 E EXPERIMENTAL SETTINGS

### 856 E.1 SYNTHETIC SYNERGY EXPERIMENT

857 **Data Generation.** For each input dimension  $n \in \{2, \dots, 10\}$ , we generate  $N = 10^6$  samples.  
858 Each sample consists of a binary input vector  $X \in \{0, 1\}^n$  with i.i.d. Bernoulli(0.5) entries.  
859 We apply a “force-to-1” noise model: with probability  $p_{\text{flip}} = 1/3$ , we randomly select one  
860 coordinate  $i \sim \text{Uniform}\{1, \dots, n\}$  and set  $X'_i = 1$ , leaving all other coordinates unchanged.  
861 With probability 2/3, no modification is made ( $X' = X$ ). The noise pattern is encoded as  
862  $\varepsilon \in \{0, 1, \dots, n\}$ , where 0 indicates no flip and  $i > 0$  indicates coordinate  $i$  was forced to 1.

864     **Functions.** We examine three deterministic functions of increasing synergy applied to the  
 865     noisy input  $X'$ :

866

- 867     • Non-synergistic:  $f_1(X') = X'_1$  (output depends only on first input)
- 868     • Partially synergistic:  $f_2(X') = X'_1 \oplus X'_2$  (XOR of first two inputs)
- 869     • Highly synergistic:  $f_3(X') = \bigoplus_{i=1}^n X'_i$  (XOR of all inputs)

870

871     **MI Estimation.** Since all variables are discrete, we compute exact MI using empirical  
 872     probability distributions with base-2 logarithms.

873

## 874     E.2 CIFAR-10 SYNERGY WITH AUGMENTATION

875

876     **Architecture and Training.** We train ResNet models of depths  $\{20, 32, 44, 56, 68, 80,$   
 877      $92, 110\}$  on CIFAR-10. Each architecture follows the standard ResNet design for CIFAR  
 878     with initial  $3 \times 3$  convolution, three residual stages, global average pooling, and a final 10-way  
 879     linear classifier. Models are trained with SGD (learning rate 0.1, momentum 0.9, weight  
 880     decay  $5 \times 10^{-4}$ ) for 200 epochs with batch size 256. Learning rate is reduced by a factor of 0.1  
 881     at epochs 100 and 150 using MultiStepLR scheduler. Standard data augmentation consists  
 882     of random crops ( $32 \times 32$  with padding 4) and horizontal flips applied during training.

883     **Teacher-Student Framework.** To quantify synergy with augmentation noise, we employ  
 884     a two-stage approach. First, a teacher network is trained as described above on augmented  
 885     data. After training, we collect the teacher's softmax outputs on the augmented training set.  
 886     We then train a student network of identical architecture to predict these teacher outputs  
 887     from non-augmented inputs. The student is trained for 200 epochs using the same SGD  
 888     configuration ( $lr=0.1$ ,  $momentum=0.9$ ,  $weight\ decay=5 \times 10^{-4}$ ) with MultiStepLR milestones  
 889     at epochs 100 and 150. The student minimizes cross-entropy loss between its predictions  
 890     and the teacher's softmax targets.

891

892     **Synergy Measurement.** We compute the marginal entropy of teacher predictions as  
 893      $H(Y) = -\mathbb{E}[(p_{\text{teacher}} \log p_{\text{teacher}})]$  where the expectation is over all augmented training  
 894     samples. The conditional entropy is estimated as the final cross-entropy loss achieved by  
 895     the converged student model. The synergy ratio  $I(f(X, \varepsilon); \varepsilon|X)/I(f(X, \varepsilon); X, \varepsilon)$  (which can  
 896     be re-written as the final loss of the student divided by the total entropy) quantifies the  
 897     proportion of the teacher's output entropy that cannot be predicted from clean images alone,  
 898     requiring knowledge of the augmentation pattern.

899

## 900     E.3 SIMPLE FUNCTIONS

901     **Network Architecture.** All networks consist of a single hidden layer with specified units,  
 902     followed by a linear output layer. No bias terms, regularization, or normalization are used.  
 903     Weights are initialized using PyTorch's default settings, namely Kaiming uniform for the  
 904     hidden layers and uniform initialization for the output layer.

905     **Target Functions and Architectures.** The target functions and architectures considered  
 906     in our evaluation are the following:

907

- 908     • Addition:  $f(a, b) = a + b$ , 2 inputs  $\rightarrow$  4 hidden units (identity activation)  $\rightarrow$  1  
 909     output;
- 910     • Multiplication:  $f(a, b) = a \times b$ , 2 inputs  $\rightarrow$  3 hidden units (square activation:  $x^2$ )  
 911      $\rightarrow$  1 output;
- 912     • Symmetric polynomial 1 (f1):  $f(a, b, c) = ab + bc + ca$ , 3 inputs  $\rightarrow$  16 hidden units  
 913     (square activation:  $x^2$ )  $\rightarrow$  1 output;
- 914     • Symmetric polynomial 2 (f2):  $f(a, b, c) = a^2 + b^2 + c^2$ , 3 inputs  $\rightarrow$  8 hidden units  
 915     (square activation:  $x^2$ )  $\rightarrow$  1 output;
- 916     • Symmetric polynomial 3 (f3):  $f(a, b, c, d) = ab + bc + cd + da$ , 4 inputs  $\rightarrow$  16 hidden  
 917     units (square activation:  $x^2$ )  $\rightarrow$  1 output.

918 **Training Details.** Networks are trained with standard gradient descent (no momentum)  
 919 with learning rate 0.01 for 1000 epochs, minimizing mean squared error (MSE) loss. Training  
 920 data consists of 1500 samples uniformly sampled from  $[-10, 10]^n$  for all functions except addition,  
 921 which uses  $[0, 10]^2$ . Test data uses 1500 samples from the extended range  $[-1000, 1000]^n$   
 922 to evaluate extrapolation. MI is computed every 10 epochs using histogram binning with 40  
 923 bins. We get our binning estimation technique from Saxe et al. (2018).

924

#### 925 E.4 RESNETS ON CIFAR-10

926

927 **Architecture.** We use standard ResNet architectures for CIFAR-10 with depths  $\{20, 56, 80, 110\}$ ,  
 928 implemented with BasicBlocks. Each network has an initial 16-filter  $3 \times 3$   
 929 convolution, followed by three stages with  $\{16, 32, 64\}$  filters respectively, global average  
 930 pooling, and a 10-way linear classifier.

931 **Training.** Models are trained on CIFAR-10 (50k train, 10k test) with SGD (momentum  
 932 0.9, weight decay  $5 \times 10^{-4}$ , initial learning rate 0.1) for 200 epochs with batch size 128.  
 933 Learning rate decays by 0.1 at epochs 100 and 150. Standard augmentation includes random  
 934 crops ( $32 \times 32$ , padding 4) and horizontal flips. No additional preprocessing is applied beyond  
 935 standard CIFAR normalization.

936 **MI Estimation.** For standard IB, we compute MI between the 10-dimensional logit vector  
 937  $\mathcal{T}$  (pre-softmax outputs) and targets  $Y$ . For GIB’s input decomposition, we first apply  
 938 Kernel PCA with RBF kernel (gamma=1/3072) to reduce the 3072-dimensional flattened  
 939 images to 50 principal components. MI is computed at each epoch using the first 5000  
 940 training samples with histogram binning (30 bins). For IB:  $I(\mathcal{T}; Y)$  using the 10-dimensional  
 941 logits. For GIB: synergy decomposition using the 50 PCA components as features.

942

#### 943 E.5 BERT ON AG NEWS

944

945 **Model Configuration.** BERT-base-uncased (12 layers, 768 hidden dimensions, 12 attention  
 946 heads) fine-tuned for 4-way AG News classification (World, Sports, Business, Sci/Tech).  
 947 The dataset contains 120,000 training and 7,600 test examples. Maximum sequence length is  
 948 128 tokens with padding.

949

950 **Training Protocols.** The training protocols used in our evaluation are the following:

951

- 952 • **Standard Fine-tuning:** Direct fine-tuning from pre-trained BERT weights for 3  
 953 epochs.
- 954 • **Unlearning + Fine-tuning:** 3 epochs of training with randomly shuffled labels  
 955 (maintaining class balance), followed by 3 epochs of standard fine-tuning.

956

957 **Optimization.** Both protocols use AdamW optimizer with learning rate  $2 \times 10^{-5}$  and  
 958 weight decay 0.01, batch size 32. No learning rate warmup or scheduling is applied. Training  
 959 uses cross-entropy loss over the 4 classes.

960

961 **MI Computation.** MI is computed 24 times per epoch (approximately every 200 batches)  
 962 using 5000 training samples. For standard IB, we use the 4-dimensional logit vector  $\mathcal{T}$  from  
 963 the classification head. For GIB, we use the raw 128-dimensional token ID sequences as  
 964 input features  $\mathcal{X}$  (no PCA is applied). MI estimation uses histogram binning with 30 bins.

965

#### 966 E.6 ADVERSARIAL ROBUSTNESS

967

968 **Architecture.** 4-layer fully-connected network:  $784 \rightarrow 1024 \rightarrow 20 \rightarrow 20 \rightarrow 20 \rightarrow 10$ , with  
 969  $tanh$  activations after each hidden layer and softmax output.

970 **Adversarial Training.** FGSM attacks are applied to every training example in each batch:  
 971  $x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(x), y))$  where  $\epsilon \in \{0.01, 0.1, 1.0\}$ . Perturbed inputs are clipped to

972 [0,1]. The training loss is the average of clean and adversarial losses:  $\mathcal{L} = (\mathcal{L}_{clean} + \mathcal{L}_{adv})/2$ .  
 973 No validation set or early stopping is used.  
 974

975 **Training Details.** Networks are trained for 10,000 epochs using Adam optimizer with  
 976 learning rate  $10^{-3}$ . MI is computed every 250 epochs between inputs and the final 20-  
 977 dimensional hidden layer activations using histogram binning (30 bins).  
 978

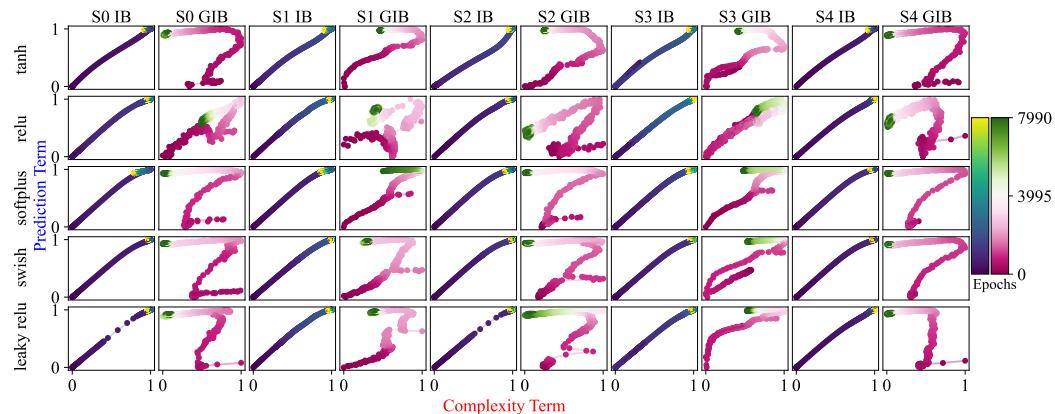
## 979 F ALTERNATIVE MI ESTIMATION USING GCMI AND KDE

982 To validate the robustness of our findings, we repeated our experiments using two alternative  
 983 MI estimation methods: Gaussian Copula Mutual Information (GCMI) and Kernel Density  
 984 Estimation (KDE). These methods offer different trade-offs between computational efficiency  
 985 and estimation accuracy compared to our primary histogram binning approach.

### 986 F.1 KERNEL DENSITY ESTIMATION (KDE)

988 KDE (Parzen, 1962) estimates probability densities using kernel functions centered at each  
 989 data point, then computes MI from these continuous density estimates. We use Gaussian  
 990 kernels with bandwidth selected via Scott’s rule (Scott, 1992). While computationally more  
 991 intensive than GCMI ( $\mathcal{O}(n^2)$  for  $n$  samples), KDE provides non-parametric estimates that  
 992 can capture arbitrary distribution shapes without assuming specific parametric forms. This  
 993 flexibility makes KDE particularly suitable for complex, multi-modal distributions that might  
 994 arise in neural network representations.

#### 996 F.1.1 ACTIVATION FUNCTION COMPARISON



1013 Figure 8: Information plane dynamics across activation functions using KDE estimation.  
 1014 The non-parametric nature of KDE reveals fine-grained dynamics in the information plane  
 1015 trajectories.  
 1016

1017 Figure 8 displays information plane dynamics using KDE estimation. The GIB formulation  
 1018 exhibits clear compression phases for all functions. Although, this can also be accompanied  
 1019 by a stage of decompression. Standard IB shows more erratic behavior, generally moves  
 1020 upward and leftward but in less distinct phases.

### 1021 F.1.2 SIMPLE MATHEMATICAL FUNCTIONS

1023 Figure 9 shows KDE-based MI estimation for simple function learning. The GIB formulation  
 1024 exhibits clear compression phases for all functions. Although, this can also be accompanied  
 1025 by a stage of decompression. Standard IB shows more erratic behavior, generally moves  
 upward and leftward but in less distinct phases.

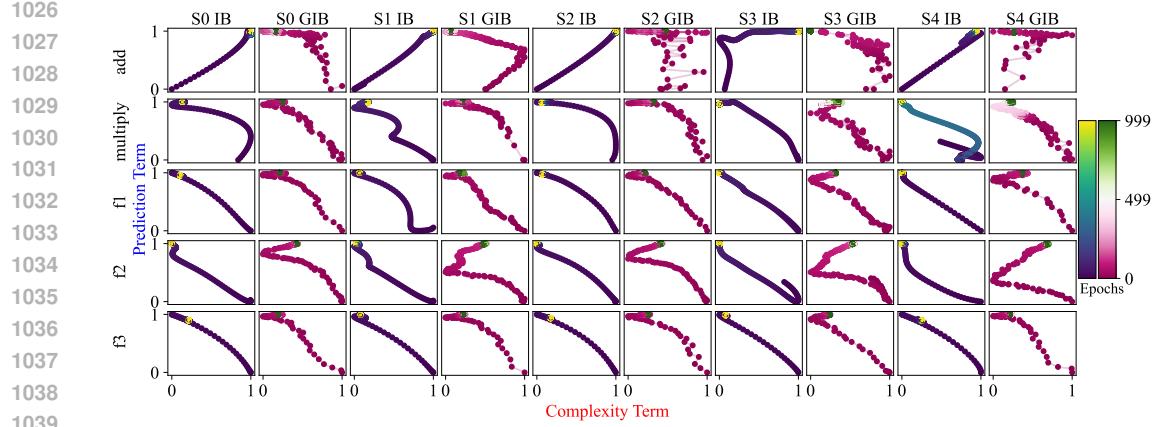


Figure 9: Information plane dynamics for simple mathematical functions using KDE estimation. The method provides sharp phase transitions despite increased trajectory variance.

#### F.1.3 RESNET INFORMATION DYNAMICS

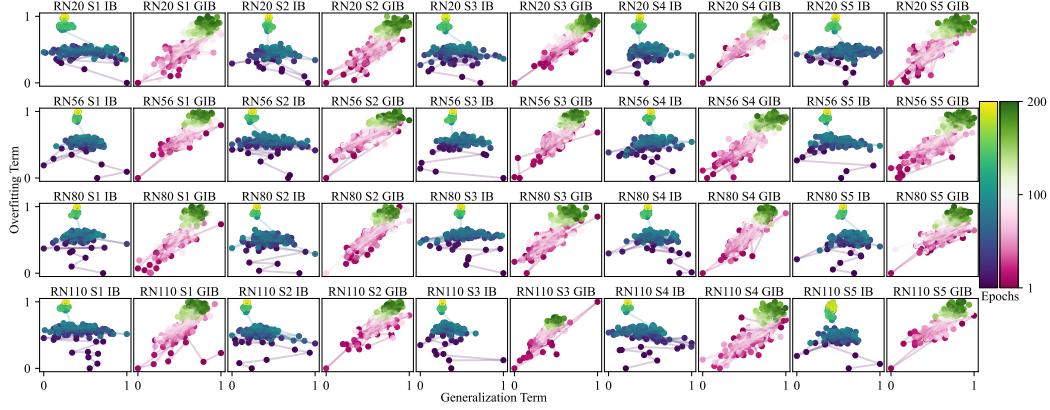


Figure 10: Information plane dynamics for ResNets on CIFAR-10 using KDE estimation.

Figure 10 reveals the first significant divergence in interpretation between the GIB results obtained using binning versus KDE. Under KDE estimation, neither GIB nor IB exhibits interpretable information bottleneck dynamics.

#### F.1.4 BERT FINE-TUNING DYNAMICS

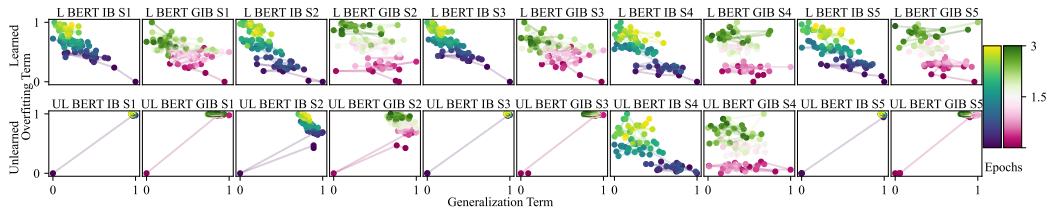


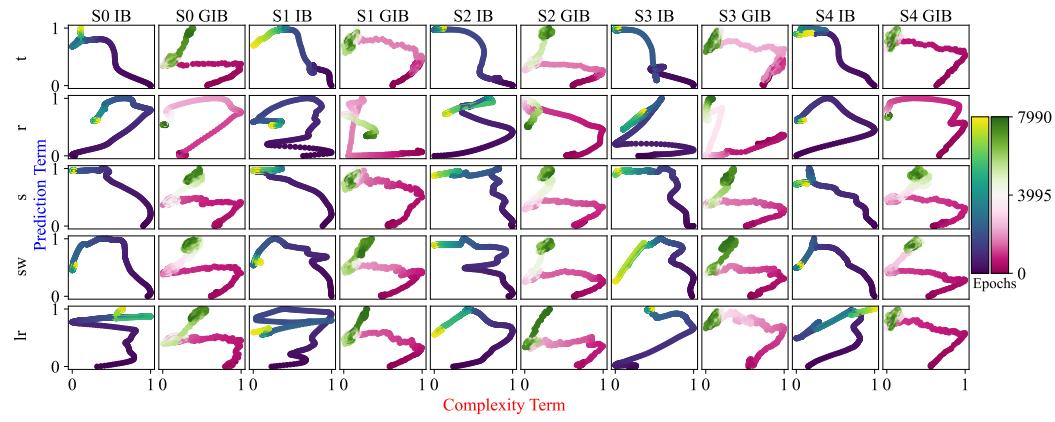
Figure 11: Information plane dynamics for BERT fine-tuning using KDE estimation.

1080  
 1081 In Figure 11, our results realign with the interpretation presented in the main text. Following  
 1082 the unlearning phase, we observe a rapid initial fitting step succeeded by a prolonged, gradual  
 1083 (and less pronounced) compression phase. These dynamics are evident for both IB and GIB.  
 1084  
 1085

## F.2 GAUSSIAN COPULA MUTUAL INFORMATION (GCMI)

1086 GCMI (Ince et al., 2017) estimates MI by first transforming variables to have standard  
 1087 Gaussian marginals using the Gaussian copula, then computing MI under the Gaussian  
 1088 assumption. This approach is particularly effective for continuous variables with complex,  
 1089 potentially non-linear relationships. The method is computationally efficient ( $O(n \log n)$  for  
 1090  $n$  samples) and provides robust estimates even for high-dimensional data. Unlike histogram  
 1091 binning, GCMI does not require discretization parameters and automatically adapts to the  
 1092 data distribution.

### F.2.1 ACTIVATION FUNCTION COMPARISON



1111 Figure 12: Information plane dynamics across activation functions using GCMI estimation.  
 1112 Standard IB (blue) and GIB (pink) trajectories for networks trained on synthetic tasks  
 1113 S0-S4.

1114  
 1115 Figure 12 presents information plane dynamics using GCMI estimation across five activation  
 1116 functions. Under this estimator, both methods frequently exhibit compression phases.  
 1117 However, GIB often displays a subsequent decompression phase, characterized by a rightward  
 1118 shift in the later stages of training.

### F.2.2 SIMPLE MATHEMATICAL FUNCTIONS

1121 Figure 13 presents results for networks learning arithmetic and polynomial functions using  
 1122 GCMI estimation. While the dynamics of both methods are not easily interpretable, the GIB  
 1123 formulation demonstrates upward leftward movement, indicating the occurrence of fitting  
 1124 and compression but they are occurring at once rather than in distinct phases. In contrast,  
 1125 standard IB shows variable behavior with limited evidence of compression.

### F.2.3 RESNET INFORMATION DYNAMICS

1127 In Figure 14, we observe that neither the IB nor the GIB yields interpretable results.

### F.2.4 BERT FINE-TUNING DYNAMICS

1128 Figure 15 likewise shows that the dynamics produced by both the IB and GIB are not  
 1129 interpretable.

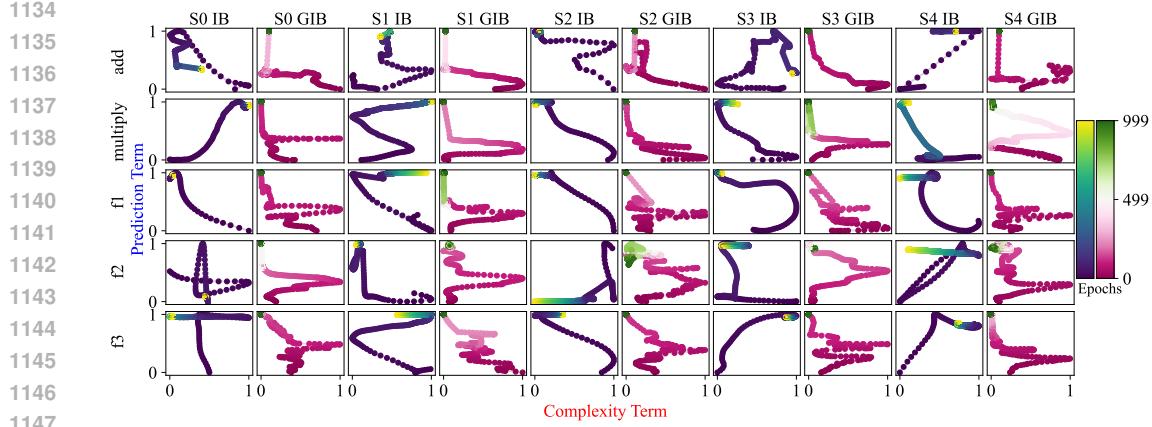


Figure 13: Information plane dynamics for simple mathematical functions using GCMI estimation.

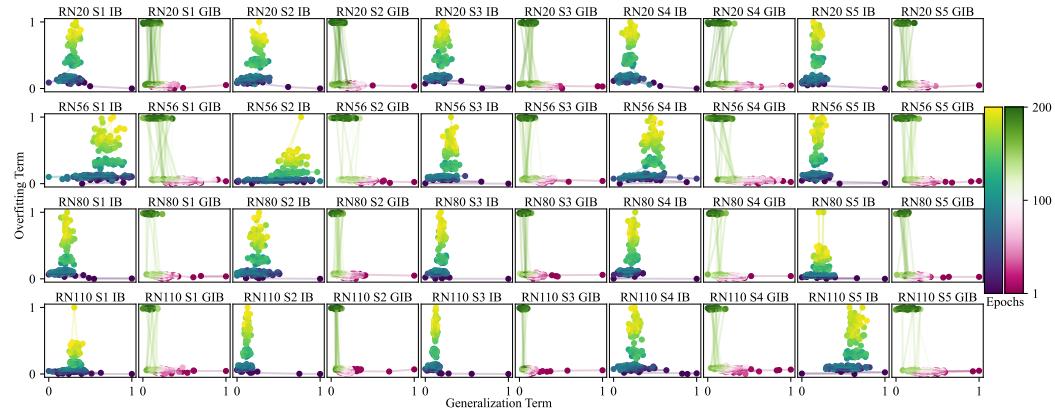


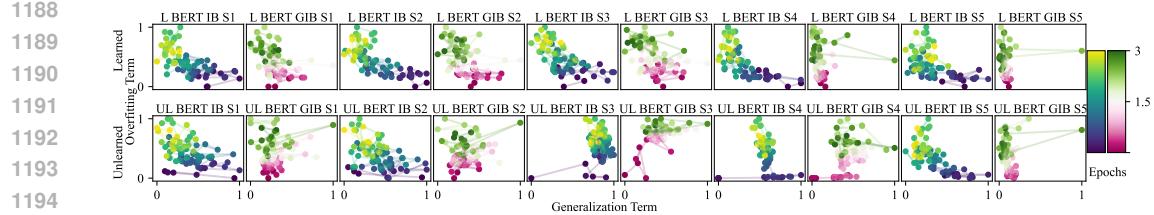
Figure 14: Information plane dynamics for ResNets on CIFAR-10 using GCMI estimation.

### F.3 SUMMARY OF GCMI AND KDE RESULTS

In this section, KDE provides strong validation for our main findings. For the KDE method, our GIB formulation consistently shows compression phases where standard IB fails, particularly for different activation functions and simple arithmetic. Meanwhile, the effect is less pronounced for GCMI, but compression remains more likely than for the standard IB. This consistency across three fundamentally different MI estimation approaches (binning, GCMI, and KDE) strongly supports our theoretical framework. The fact that synergy-based decomposition reveals consistent information dynamics across estimation methods suggests that GIB captures a fundamental aspect of how neural networks process information during learning.

## G COMPUTATIONAL COMPLEXITY

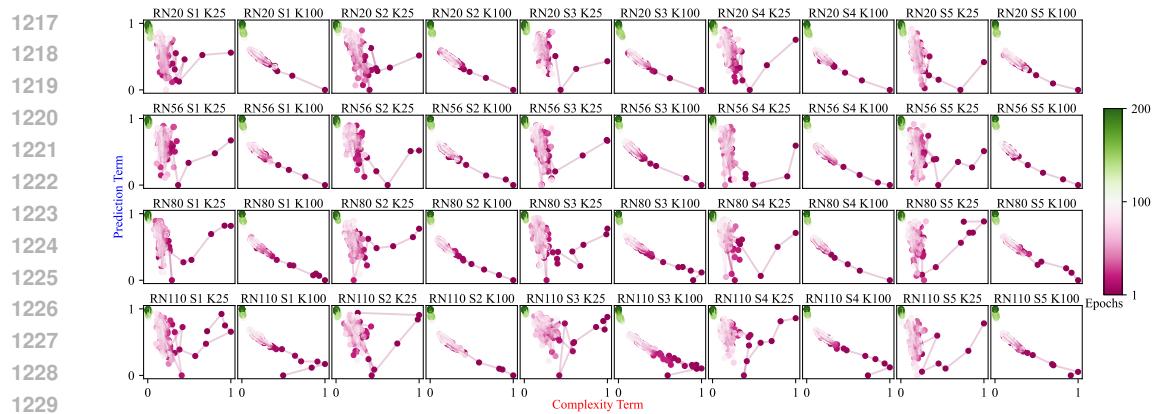
The computational requirements of GIB and IB differ significantly in their scaling behavior. If we define our unit of computation as a single MI estimation, GIB requires  $2N + 1$  calculations: more specifically, one for the prediction term  $I(Z; Y)$  and  $2N$  for the complexity term (computing  $I(\mathcal{X}^{-i}; Q)$  and  $I(X^i; Q)$  for each feature). Critically, these calculations occur at the input layer where dimensionality is typically highest, for CIFAR-10, this means 3072

1197 Figure 15: Information plane dynamics for BERT fine-tuning using GCMI estimation.  
1198  
1199  
1200

1201 features. However, GIB’s computational cost is independent of network depth, since it only  
1202 considers input-output relationships. In contrast, standard IB requires  $2L$  MI calculations for  
1203  $L$  layers, computing  $I(\mathcal{X}; \mathcal{T}_i)$  and  $I(\mathcal{T}_i; Y)$  at each layer. While one might compute IB only  
1204 for the final layer where dynamics are often most pronounced, this prevents the use of the IB  
1205 as a tool for understanding learning dynamics throughout the network. Additionally, GIB  
1206 benefits from a key advantage: we can apply PCA to high-dimensional inputs (as we do for  
1207 CIFAR-10 in Section 5.2) because features at the input layer share a common representation  
1208 space (Turk & Pentland, 1991). Conversely, combining representations across layers for IB is  
1209 less conventional.

1210  
1211  
1212 

## H EFFECT OF PCA DIMENSIONALITY ON GIB DYNAMICS

  
1213  
1214  
1215  
12161227 Figure 16: GIB information plane dynamics for ResNets with varying PCA dimensionality.  
1228 Each subplot shows results for KPCA with 25 (left) versus 100 (right) components (C’s). As  
1229 dimensionality increases, compression phases become more pronounced.  
1230  
1231

1232 Figure 16 demonstrates the impact of PCA dimensionality on observed GIB dynamics. With  
1233 only 25 principal components, the dynamics are erratic and noisy. However, as we increase to  
1234 50 components (shown in main results) and then to 100 components, the dynamics become  
1235 less noisy. This progression suggests that capturing synergistic information requires sufficient  
1236 dimensionality to represent the complex feature interactions present in the original input  
1237 space.

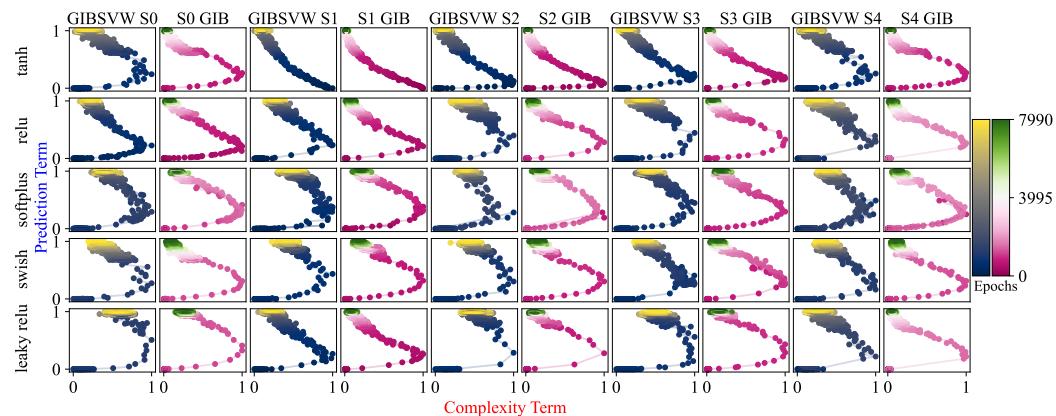
1242 I SUM VERSUS WHOLE SYNERGY FORMULATION  
12431244 I.1 SUM-VERSUS-WHOLE SYNERGY  
1245

1246 In this section, we examine an alternative formulation of synergy based on sum versus whole  
1247 synergy rather than our feature-wise approach. Due to the increased noise in this estimation  
1248 method, all MI values are averaged over 50 iterations to obtain stable measurements.

1249 This alternative definition of GIB is based on the sum-versus-whole formulation of synergy,  
1250 which compares information available from the complete feature set against the sum of  
1251 information from individual components (Schneidman et al., 2003). The basic form is  
1252  $\text{SyngIB}(\mathcal{X} \rightarrow Y) = I(\mathcal{X}; Y) - \sum_{i=1}^N (I(X^i; Y))$ , which considers only individual features.  
1253 This captures the intuitive notion of synergy, for example, XOR has zero information from  
1254 individual inputs but perfect information from their combination, yielding maximal synergy  
1255 (Bell, 2003). Unlike exponentially complex PID-based measures (Williams & Beer, 2010), this  
1256 formulation requires only  $O(N)$  mutual information calculations, making it computationally  
1257 feasible for tracking synergistic learning dynamics in high-dimensional neural networks.  
1258 Combining this with our representation of the PMI-weighted combination of  $Z$  and  $Y$  and  
1259 rewriting as a Lagrangian optimization we get the following:

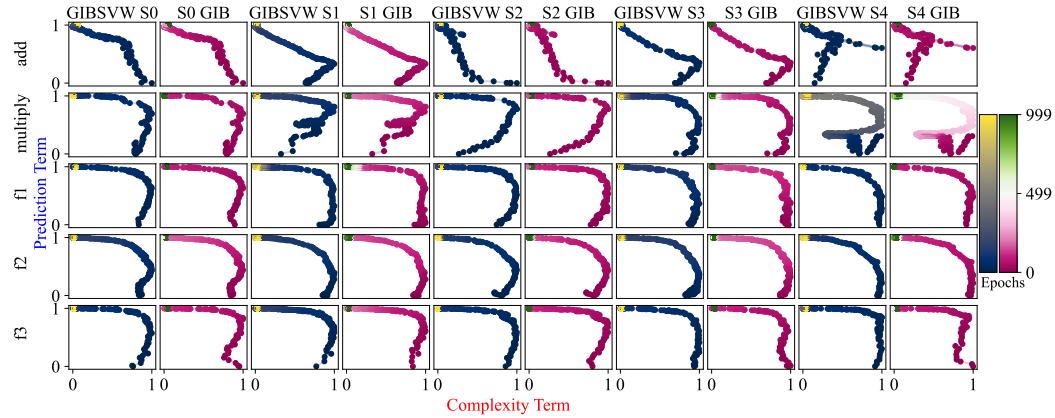
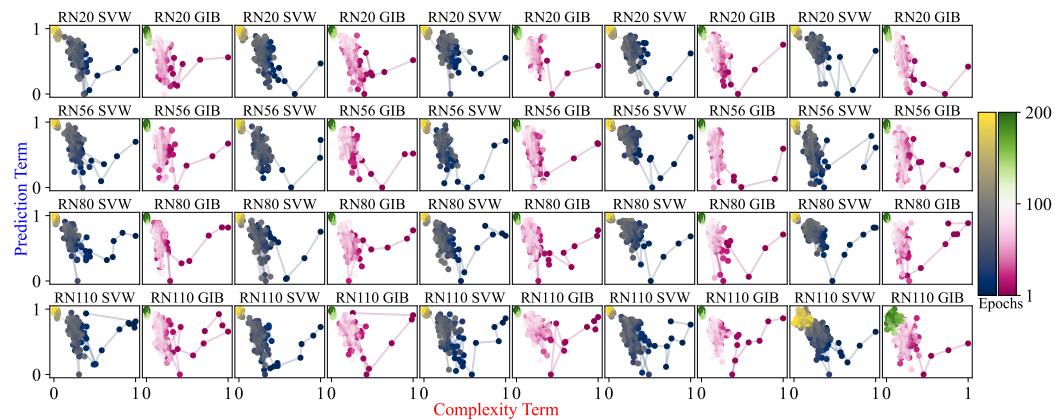
$$1260 \quad \mathcal{L}_{\text{SVW}} = \max_{p(Z|X)} \left[ \underbrace{I(Z; Y)}_{\text{prediction term}} - \underbrace{\beta^{-1} \sum_{i=1}^N I(X^i; Q)}_{\text{complexity term}} \right] \quad (17)$$

1261 In the following section, we compare the outcomes of tracking this optimization with those  
1262 obtained from the approach introduced in the main paper.

1263 I.2 ACTIVATION FUNCTION COMPARISON  
1264

1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
Figure 17: Information plane dynamics comparing the alternative synergy bottleneck (SVW, blue) with our GIB (pink) across multiple activation functions. While SVW shows compression phases more frequently than standard IB, it exhibits less consistent compression than our feature-wise GIB formulation. MI values averaged over 50 iterations.

Figure 17 shows that the alternative synergy formulation (SVW) improves upon standard IB by exhibiting compression phases in several cases where IB fails. However, the compression is less pronounced and less consistent across activation functions compared to our feature-wise GIB. This suggests that while any synergy-based decomposition provides benefits over treating the latent space as a black box, the specific choice of synergy might impact the observability of information dynamics.

1296 I.3 SIMPLE MATHEMATICAL FUNCTIONS  
12971313 Figure 18: Learning dynamics for simple mathematical functions. The alternative synergy  
1314 bottleneck (SVW) consistently shows compression phases across all functions, significantly  
1315 outperforming standard IB.  
13161317 For NNs learning simple mathematical functions (Figure 18), the alternative synergy formu-  
1318 lation consistently exhibits compression phases across all tasks. This represents a substantial  
1319 improvement over standard IB, which shows no compression for these functions.  
13201321 I.4 RESNET INFORMATION DYNAMICS  
13221338 Figure 19: ResNet information dynamics on CIFAR-10. The alternative synergy bottleneck  
1339 (SVW) shows distinct compression and non-compression phases, providing clearer learning  
1340 dynamics than standard approaches while exhibiting more variability than our feature-wise  
1341 GIB.  
13421343 Figure 19 shows that the alternative synergy formulation reveals distinct phases in ResNet  
1344 training.  
13451346 I.5 BERT FINE-TUNING DYNAMICS  
13471348 The first notable limitation of the alternative synergy formulation appears in transformer  
1349 fine-tuning (Figure 20). The SVW method fails to exhibit compression phases for BERT on  
the AG News classification task, even after our unlearning intervention. In contrast, our GIB

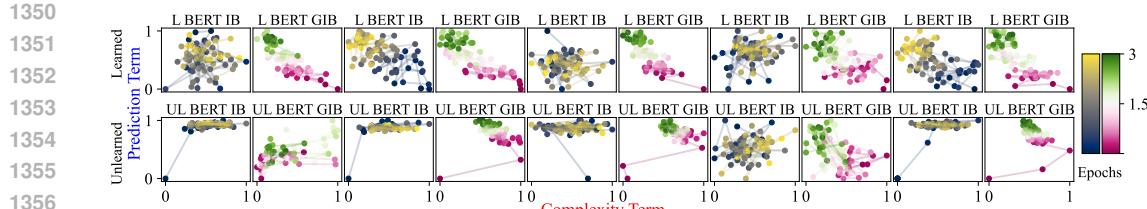


Figure 20: BERT fine-tuning with the alternative synergy bottleneck. Unlike our GIB formulation, SVW fails to show compression phases for both standard fine-tuning and the unlearning protocol, suggesting inherent limitations in capturing synergistic dynamics in transformer architectures.

formulation clearly reveals compression dynamics. This discrepancy suggests that variance-weighted synergy measures may struggle to capture the high-dimensional, attention-based computations characteristic of transformers.

## I.6 SUMMARY

The alternative synergy formulation serves as a useful baseline, showing that synergy-based approaches generally outperform standard IB.

1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403