# Track 1:
# Achieving Domain-Independent Certified Robustness via *Knowledge Continuity*

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We present *knowledge continuity*, a novel definition inspired by Lipschitz continuity which aims to certify the robustness of neural networks across input domains (such as continuous and discrete domains in vision and language, respectively). Most existing approaches that seek to certify robustness, especially Lipschitz continuity, lie within the continuous domain with norm and distribution-dependent guarantees. In contrast, our proposed definition yields certification guarantees that depend only on the loss function and the intermediate learned metric spaces of the neural network. These bounds are independent of domain modality, norms, and distribution. We further demonstrate that the expressiveness of a model class is not at odds with its knowledge continuity. This implies that achieving robustness by maximizing knowledge continuity should not theoretically hinder inferential performance. Finally, we present several applications of knowledge continuity such as regularization and show that knowledge continuity can also localize vulnerable components of a neural network.

## 1 Introduction

Deep neural networks (DNNs) have demonstrated remarkable generalization capabilities. Their robustness, however, has been considerably more difficult to achieve. Robustness refers to the preservation of model performance under natural or adversarial alterations of the input [14]. DNNs' lack of robustness, highlighted by seminal works such as [19, 53] and recently [6, 4], poses significant challenges to their adoption in critical applications, underscoring concerns for AI safety and trustworthiness [15, 23, 7, 6].

Though issues of robustness emerged from computer vision applications, they have since spanned multiple domains [1, 29, 59, 62, 6]. This research trajectory has not only prompted significant advancements in robustness improvements through architectural, procedural, and dataset augmentations, but also unveiled the sophistication of adversarial attacks—the process through which counterexamples to robustness are generated [1, 29, 59, 62, 6]. In particular, a great deal of work has gone into certified robustness which seeks to provide theoretical robustness guarantees. Certification is desirable as it generally transcends any particular task, dataset, or model.

As a result, *Lipschitz continuity* has emerged, promising certified robustness by bounding the derivative of a neural network's output with respect to its input. In this way, Lipschitz continuity directly captures the volatility of a model's performance, getting at the heart of robustness. Such an approach has proven its merit in computer vision, facilitating robustness under norm and distributional assump-

Submitted to AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA. Do not distribute.

tions [22, 50, 65, 63]. Its inherent ease and interpretability has lead to widespread adoption as a means to measure and regulate robustness among practitioners as well [58, 10, 16, 55, 47].

Despite these successes in computer vision, there are fundamental obstacles when one tries to apply Lipschitz continuity into discrete or non-metrizable domains such as natural language. Firstly, characterizing distance in this input (and output) space is highly nontrivial, as language does not have a naturally-endowed distance metric. Additionally, distance in this input (and output) space cannot be task-invariant, as context could dramatically change the meaning of a sentence [41]. Lastly, key architectures such as the Transformer [57] are provably *not* Lipschitz continuous [30]. Most of these challenges are not unique to language and form the tip of the iceberg that represents the strong divide of robustness between discrete/non-metrizable and continuous domains [17, 38]. For a detailed summary of the related literature, see Appendix A.

To address these issues, we propose a new conceptual framework which we call *knowledge continuity*. At its core, we adopt the following axiom:

> *Robustness is the stability of a model's performance with respect to its perceived knowledge of input-output relations.*

Concretely, our framework is grounded on the premise that robustness is better achieved by focusing on the probabilistic variation of a model's loss with respect to its hidden representations, rather than forcing arbitrary metrics on its inputs and outputs. Our approach results in certification guarantees independent of domain modality, norms, and distribution. We demonstrate that the expressiveness of a model class is not at odds with its knowledge continuity. In other words, achieving robustness by improving knowledge continuity should not theoretically hinder inferential performance. We show that in continuous settings (i.e. computer vision) knowledge continuity generalizes Lipschitz continuity and inherits its tight robustness bounds. Finally, we present an array of practical applications using knowledge continuity both as an indicator to predict and characterize robustness as well as an additional term in the loss function to train robust classifiers.

Although our results apply to all discrete/non-metrizable and continuous spaces, throughout the paper we invoke examples from natural language as it culminates the aforementioned challenges. Further, the ubiquity of large language models make their robustness a timely focus.

# 2 Knowledge Continuity

In this section, we provide a formulation of *knowlege continuity* and explore its theoretical properties. Refer to Appendix B for all of the necessary background and notation.

We start by defining a model's perceived knowledge through a rigorous treatment of its hidden representation spaces. By considering the distance between inputs in some representation space in conjunction with changes in loss, we result in a measure of *volatility* analogous to Lipschitz continuity. Bounding this volatility in expectation then directly leads to our notion of knowledge continuity. With these tools, we demonstrate a host of theoretical properties of knowledge continuity including its certification of robustness, guarantees of expressiveness, and connections to Lipschitz continuity in continuous settings. We summarize our theoretical contributions as follows:

- We ***define*** the perceived knowledge of a model as well as volatility and knowledge continuity within a model's representation space (see Def. 1, 2, 3, 4, respectively).
- We ***prove*** that knowledge continuity implies *probabilistic* certified robustness under perturbations in the representation space and constraining knowledge continuity should not hinder the expressiveness of the class of neural networks (see Thm. 2.1 and Prop. 2.2, 2.3, respectively).
- We ***prove*** that in some cases knowledge continuity is equivalent (in expectation) to Lipschitz continuity. This shows that our axiomization of robustness aligns with existing results when perturbation with respect to the input is well-defined (see Prop. 2.4, 2.6).

## 2.1 Defining Perceived Knowledge

Knowledge is generally accepted as a relational concept, as it arises from the connections we make between ideas and experiences [21]. Herein, we capture the perceived knowledge of a model by focusing on the relations it assigns to input-input pairs. Specifically, these relations are exposed by decomposing a function $f : \mathcal{X} \to \mathcal{Y}$ into projections to intermediate metric spaces. Formally,

**Definition 1** (Metric Decomposition)**.** *We say that $f$ admits a metric decomposition if there exists metric spaces $(\mathcal{Z}_1, d_1), \ldots, (\mathcal{Z}_n, d_n)$ with metrics $d_k$ for $k \in [n]$ such that*

1. *$(\mathcal{Z}_k, d_k)$ is endowed with its Borel $\sigma$-algebra.*
2. *There exists measurable mappings $h_0, h_1, \ldots, h_n$ where $h_0 : \mathcal{X} \to \mathcal{Z}_1$, $h_k : \mathcal{Z}_k \to \mathcal{Z}_{k+1}$ for $k \in [n-1]$, and $h_n : \mathcal{Z}_n \to \mathcal{Y}$.*
3. *$f = h_n \circ h_{n-1} \circ \ldots \circ h_1 \circ h_0$.*

To the best of our knowledge, all deep learning architectures admit metric decompositions, since their activations are generally real-valued. So, for all subsequent functions from $\mathcal{X}$ to $\mathcal{Y}$, unless otherwise specified, we assume they are measurable and possess a metric decomposition. Further, we denote $f^k = h_k \circ h_{k-1} \circ \ldots \circ h_1 \circ h_0$ and adopt the convention of calling $h_k$ the $k^{\text{th}}$ hidden layer. In Appendix C, we present several metric decompositions for a variety of architectures.

For any metric-decomposible function, an immediate consequence of our definition is that its metric decomposition may not be unique. However, in the context of neural networks, this is a desirable property. Seminal works from an array of deep learning subfields such as semi-supervised learning [49], manifold learning [43], and interpretability [8] place great emphasis on the quality of learned representation spaces by examining the induced-topology of their metrics. This often does not affect the typical performance of the estimator, but has strong robustness implications [27]. Our results, which are dependent on particular metric decompositions, capture this trend. In Section 2.4, we discuss in detail the effects of various metric decompositions on our theoretical results.

### 2.2 Defining Knowledge Continuity

We first introduce what it means for a model's performance to be volatile at a data point with respect to some learned representation of that model.

**Definition 2** ($k$-Volatility)**.** *Let $f : \mathcal{X} \to \mathcal{Y}$ and $\mathcal{L}$ be any loss function. The $k$-volatility of a point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ which we denote as $\sigma_f^k(x, y)$ is given by*

$$\sigma_f^k(x, y) := \mathbb{E}_{\substack{(x',y') \sim \mathcal{D}_{\mathcal{X},\mathcal{Y}} \\ f(x) \neq f(x')}} \left[ \frac{\Delta \mathcal{L}_f^{(x,y)}(x', y')}{d_k(f^k(x), f^k(x'))} \right]. \tag{2.1}$$

By performing some algebra on the definition, we see that it decomposes nicely into two distinct terms: *sparsity* of the representation and *variation* in loss.

$$\begin{aligned} \sigma_f^k(x, y) &= \mathbb{E}_{(x',y') \sim \mathcal{D}_{\mathcal{X},\mathcal{Y}}} \left[ \frac{|\mathcal{L}(f(x), y) - \mathcal{L}(f(x'), y')|}{d_k(f^k(x), f^k(x'))} \right], \\ &= \mathcal{L}(f(x), y)\, \mathbb{E}_{(x',y')} \Bigg[ \underbrace{\frac{1}{d_k(f^k(x), f^k(x'))}}_{\text{sparsity}} \cdot \underbrace{\left| 1 - \frac{\mathcal{L}(f(x'), y')}{\mathcal{L}(f(x), y)} \right|}_{\text{variation in loss}} \Bigg], \end{aligned} \tag{2.2}$$

Our notion of volatility essentially measures the derivative of performance with respect to perturbations to a model's perceived knowledge. In particular, Eq. 2.2 reveals that there are two interactions in play which we illustrate in Fig. 1. Informally, we say that $(x, y)$ is highly volatile if there is a large discrepancy in performance between it and points that are perceived to be conceptually similar. Therefore, highly volatile points capture inaccurate input-input knowledge relations. Additionally, $(x, y)$ experiences low volatility if the space around it is sparse with respect to $\mathcal{D}_{\mathcal{X},\mathcal{Y}}$. In other words, any set of perturbations applied in $\mathcal{Z}_k$ would push $(x, y)$ far away, with high probability. This makes $(x, y)$ an isolated concept with little knowledge relationships associated with it.

Similar to Lipschitz continuity, the boundedness of the $k$-volatility of $f$ across the data distribution is crucial and we denote this class of functions as *knowledge continuous*.

**Definition 3** ($\varepsilon$-Knowledge Continuity at a Point)**.** *We say that $f$ is $\varepsilon$-knowledge continuous at $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with respect to a function $f$, loss function $\mathcal{L}$, and hidden layer $k$ if $\sigma_f^k(x, y) < \varepsilon$.*

Conversely, we say that $(x, y)$ is $\varepsilon$-knowledge discontinuous if the previous inequality does not hold. Further, $(x, y)$ is simply knowledge discontinuous if $\sigma_f^k(x, y)$ is unbounded. Now, we extend this definition globally by considering the $k$-volatility between all pairs of points.

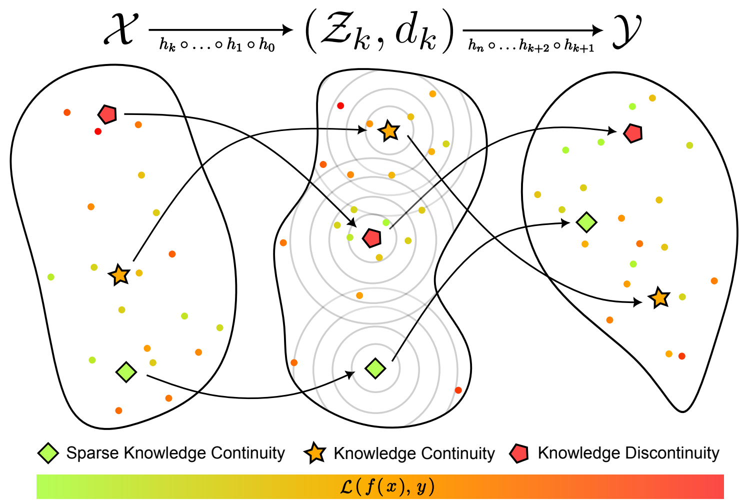


Figure 1: Various types of knowledge (dis)continuities. $f : \mathcal{X} \to \mathcal{Y}$ is a measurable map, and $(\mathcal{Z}_k, d_k)$ is one of its hidden representation. ◆ denotes knowledge continuity from sparsity: an isolated concept with no knowledge relations close to it. So, any perturbation moves ◆ far away with high probability. Smooth changes in loss around ★ implies knowledge continuity. Finally, ⬟ lacks continuity due to drastic changes in loss nearby.

**Definition 4** ($\varepsilon$-Knowledge Continuity)**.** *We say that $f$ is $\varepsilon$-knowledge continuous with respect to a loss function $\mathcal{L}$ and hidden layer $k$ if*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\sigma_f^k(x,y)] < \varepsilon. \tag{2.3}$$

Though the functional forms of Lipschitz continuity and knowledge continuity are similar, there are important differences that allow us to prove more general results. Firstly, unlike Lipschitz continuity which is an analytical property of the model $f$, knowledge continuity is a statistical one. In this way, non-typical data points, even if they are volatile, are ignored, whereas Lipschitz continuity treats all points equally. This is necessary in many discrete applications, as projecting a countable input space onto a non-countable metric space inevitably results in a lack of correspondence thereof. Moreover, the ground-truth function from $\mathcal{X} \to \mathcal{Y}$ may not be well-defined on *all* of $\mathcal{X}$: consider sentiment classification of an alpha-numeric UUID string or dog-cat classification of Gaussian noise. Secondly, knowledge continuity of an estimator is measured with respect to the loss function rather than its output. This property allows us to achieve the expressiveness guarantees in Section 2.4, since it places no restrictions on the function class of estimators. Lastly, knowledge continuity measures the distance between inputs with the endowed metric in its hidden layers. This flexibility allows us to define knowledge continuity even when the input domain is not a metric space.

### 2.3 Certification of Robustness

Our first main result demonstrates that $\varepsilon$-knowledge continuity implies probabilistic certified robustness in the hidden representation space. In Theorem 2.1, given some reference set $A \subset \mathcal{X} \times \mathcal{Y}$, we bound the probability that a $\delta$-sized perturbation in the representation space away from $A$ will result in an $\eta$ change in loss. In other words, knowledge continuity is able to characterize the robustness of any subset of data points with positive measure.

**Theorem 2.1.** *Let $A \subset \mathcal{X} \times \mathcal{Y}$ such that $\mathbb{P}_{\mathcal{D}_{\mathcal{X},\mathcal{Y}}}[A] > 0$ and $\delta, \eta > 0$. Let $A' = \{(x', y') \in \mathcal{X} \times \mathcal{Y} : \exists (x,y) \in A, \Delta\mathcal{L}_f^{(x,y)}(x', y') > \eta\}$. If $f : \mathcal{X} \to \mathcal{Y}$ is $\varepsilon$-knowledge continuous with respect to the hidden layer indexed by $k$ and $(\mathcal{Z}_k, d_k)$ is bounded by $B > 0$, then*

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}[A' \mid d_k(f^k(x), f^k(A)) < \delta] \leq \frac{\varepsilon\delta}{\eta\left(1 - \exp\left[-\Omega\left(\frac{\delta}{B} - \sqrt{\log\frac{1}{\mathbb{P}[A]}}\right)^2\right]\right)}. \tag{2.4}$$

See Appendix D for the proof. We can lose the assumptions of boundedness and knowledge of $\mathbb{P}[A]$ by taking limits of Eq. D.11 with respect to $B$ and $\mathbb{P}[A]$. This result is shown in Appendix D.

### 2.4 Expressiveness

Our second main result demonstrates that $\varepsilon$-knowledge continuity can be achieved without theoretically compromising the accuracy of the model. In other words, universal function approximation is an invariant property with respect to $\varepsilon$-knowledge continuity. Universal approximation puts limits on what neural networks can learn [12, 24, 37]. A major limitation of Lipschitz functions is that they are not universal function approximators of arbitrary functions (see Appendix A for a detailed discussion). However, we show that this is achievable with knowledge continuity.

First, we formally define a universal function approximator.

**Definition 5** (Universal Function Approximator)**.** *Suppose that $\mathcal{L}$ is Lebesgue-integrable in both coordinates. Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a set of measurable functions from $\mathcal{X} \to \mathcal{Y}$ such that for any $f \in \mathcal{F}$, there exists $\mu_f \ll \mathcal{D}_{\mathcal{X},\mathcal{Y}}$ such that $\mu_f(graph(f)) = 1$. Then, $\mathcal{U} \subset \mathcal{F}$ is a universal function approximator of $\mathcal{F}$ if for every $f \in \mathcal{F}$ and every $\varepsilon > 0$, there exists $\hat{f} \in \mathcal{U}$ such that*

$$\int \mathcal{L}(\hat{f}(x), y)\, d\mu_f < \varepsilon. \tag{2.5}$$

We now show that it is always possible to learn some hidden representation that is perfectly robust.

**Proposition 2.2.** *Let $\mathcal{U} \subset \mathcal{Y}^{\mathcal{X}}$ be a universal function approximator of $\mathcal{Y}^{\mathcal{X}}$ with respect to some loss function $\mathcal{L}$. Then, for any $f \in \mathcal{Y}^{\mathcal{X}}$ and sequence $\varepsilon_1, \varepsilon_2, \ldots$ such that $\varepsilon_n \to 0$ there are a sequence of $\varepsilon_n$-knowledge continuous functions in $\mathcal{U}$ such that $\int \mathcal{L}(f_n(x), y)\, d\mu_f < \varepsilon_n$, for $n \in \mathbb{N}$.*

*Proof.* Choose $f_n \in \mathcal{U}$ such that $\int \mathcal{L}(f_n(x), y)\, d\mu_f < \frac{1}{2}\varepsilon_n$. Consider the 1-layer metric decomposition of $f$, $h_1 : \mathcal{X} \to \mathcal{Z}_1$ where $\mathcal{Z}_1 = \mathcal{X}$ equipped with the trivial metric ($d_1(x, y) = 1$ if $x \neq y$ and 0 otherwise). Then, $f_n = f_n \circ h_1$. So, it follows that

$$\mathbb{E}\, \sigma^1_{f_n}(x, y) = \int \frac{\Delta\mathcal{L}^{(x,y)}_{f_n}(x', y')}{d_1(h_1(x), h_1(x'))}\, d\mu_f \leq \int \Delta\mathcal{L}^{(x,y)}_{f_n}(x', y')\, d\mu_f \leq \varepsilon_n. \tag{2.6}$$

and by the construction of $f_n$, the proof is completed. ∎

In other words, if our estimator was given "infinite representational capacity," robustness can be trivially achieved by isolating every point as its own concept (as discussed in Section 2.2). We can, however, construct a tighter model by relaxing the assumptions on the input-output metric spaces. These added constraints make it so that trivial metric decompositions are no longer possible unless the metric in $\mathcal{X}$ is also trivial. We state this formally below, note the highlighted differences between this and Prop. 2.2.

**Proposition 2.3.** *Suppose $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}}) \coloneqq (\mathcal{X}, d_{\mathcal{X}})$ are **compact** metric spaces, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ is the **set of all continuous functions** from $\mathcal{X}$ to $\mathcal{Y}$ such that $\int d_{\mathcal{X}}(x, x')^{-1} d\mu_f < \infty$ and $\mathcal{L}$ be Lipschitz continuous in both coordinates. Then, there exists a universal function approximator $\mathcal{U}$ of $\mathcal{F}$ that is knowledge continuous (i.e. $\mathbb{E}\, \sigma^k_f(x, y) < \infty$ for some $k$).*

See Appendix E for the proof.

### 2.5 Connections to Lipschitz Continuity

We now demonstrate that our axiomization of robustness presented in Section 1 aligns with the notion of robustness[1] commonly prescribed in vision [14]. This unifies the certified robustness bounds with respect to the representation space derived in Thm. 2.1 with existing work certifying robustness with respect to the input space in continuous applications such as vision.

Our first result identifies conditions under which knowledge continuity, implies Lipschitz continuity.

**Proposition 2.4.** *Suppose that $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}})$ are metric spaces. Let the first $n$ metric decompositions of $f : \mathcal{X} \to \mathcal{Y}$ be $K_i$-Lipschitz continuous, for $i \in [n]$. If $f$ is $\varepsilon$-knowledge continuous with respect to the $n^{th}$ hidden layer and $d_{\mathcal{Y}}(f(x), f(x')) \leq \eta \Delta\mathcal{L}^{(x,y)}_f(x', y)$ for all $x, x' \in \mathcal{X}$, $y \in \mathcal{Y}$, and some $\eta > 0$, then $f$ is Lipschitz continuous in expectation. That is,*

$$\mathbb{E}_{(x,y),(x',y') \sim \mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x, x')} \leq \varepsilon\eta \prod_{j=1}^{n} K_j. \tag{2.7}$$

[1] Small perturbations on the input result in small changes in performance which implies small changes in output when the loss function is Lipschitz continuous.

The proof is presented in Appendix F and follows easily through some algebriac manipulation. Next, combining this proposition with an auxiliary result from [74], we directly yield a certification on the input space.

**Corollary 2.5.** *Suppose that assumptions of Prop. 2.4 are true. And also assume that* $(\mathcal{X}, d_{\mathcal{X}}) = (\mathbb{R}^n, \ell_p)$*,* $(\mathcal{Y}, d_{\mathcal{Y}}) = (\mathbb{R}^m, \ell_p)$*, for* $1 \le p \le \infty$*. Define a classifier from* $f : \mathbb{R}^n \to \mathbb{R}^m$*,* $g$*, where* $g(x) \coloneqq \arg\max_{k\in[m]} f_k(x)$ *for any* $x \in \mathbb{R}^n$*. Then, with probability* $1 - \frac{\varepsilon\eta}{t}\prod_{j=1}^{n} K_j$*,* $g(x) = g(x+\delta)$ *for all* $\|\delta\|_p < \frac{\sqrt[p]{2}}{2t}$*margin*$(f(x))$ *and* $t > 0$*.* $f_k(x)$ *is the* $k^{th}$ *coordinate of* $f(x)$ *and margin*$(f(x))$ *denotes the difference between the largest and second-largest output logits.*

See Appendix F for the proof. Our second result identifies conditions under which Lipschitz continuity, implies knowledge continuity.

**Proposition 2.6.** *Let* $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}})$ *be a metric spaces. Let* $f : \mathcal{X} \to \mathcal{Y}$ *be* $\varepsilon$*-Lipschitz continuous and* $\mathcal{L}(f(x), y)$ *be* $\eta$*-Lipschitz continuous with respect to both coordinates. If the first* $n$ *metric decompositions of* $f$ *are* $K_i$*-Lipschitz continuous, then* $f$ *is knowledge continuous with respect to the* $n^{th}$ *hidden layer. That is,*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}\, \sigma_f^n(x,y) \le \varepsilon\eta \prod_{j=1}^{n} \frac{1}{K_j}. \tag{2.8}$$

See Appendix F for the proof. In continuous applications such as computer vision, the assumptions of both propositions are generally met (i.e. our input-output spaces are metric spaces, all hidden layers are Lipschitz, and loss functions are locally Lipschitz). Furthermore, common architectures such as fully connected networks, CNNs, RNNs, and even vision transformers are Lipschitz continuous [58, 48]. This implies that our notion of robustness is indeed an appropriate generalization that transcends domain modality since in continuous settings we can recover the strong bounds of Lipschitz continuity while expanding into new discrete and non-metrizable territory.

## 3 Practical Applications

In addition to the theoretical guarantees yielded by knowledge continuity in Section 2, we now demonstrate that knowledge continuity can be easily applied in practice.

**Using knowledge continuity to predict adversarial robustness.** For a given model, $f$, and hidden representation, $k$, we first determine the smallest $\varepsilon_k$ such that $f$ is $\varepsilon_k$-knowledge continuous. Then, we collate all $\varepsilon_k$ through a simple average. When we regress these scores from a series of model families and sizes against their empirical adversarial robustness strong correlation is observed. In particular, knowledge continuity alone is able to explain 35% of the variance in adversarial attack success rate. We present a detailed discussion of these experiments in Appendix G.

**Knowledge continuity can localize vulnerable hidden representations.** Since knowledge continuity is layer-specific, we repeat the previous experiment, but holding the index of the hidden representation constant. We plot the relationship between explained variance of adversarial robustness and layer index. We find that models belonging to different families result in dramatically different curves. We tune our regularization hyperparameters according to these curves and find they yield superior performance. These results are represented in Appendix G, H, and I.

**Regulating knowledge continuity.** Motivated by the theoretical results in Section 2, we devise algorithms to estimate the $k$-volatility of a given model during training. These algorithms are described in Appendix I along with guarantees on their convergence rate and unbiasedness. Then, we directly append this estimate of volatility to our loss function as a regularization term. By minimizing this regularized loss, we find that the adversarial robustness of the resulting model significantly improves. Moreover, our method outperforms existing works both in terms of robustness and training speed (up to 2× for TextFooler [29] and 3× for ALUM [36]). These results are presented in Appendix I, Table 1.

## 4 Conclusion

In this paper, we propose a novel definition, *knowledge continuity*, which addresses key limitations associated with Lipschitz robustness. We demonstrate that our definition certifies robustness across domain modality, distribution, and norms. We also show that knowledge continuity, in contrast to Lipschitz continuity, does not affect the universal approximation property of neural networks. We further establish conditions under which knowledge continuity and Lipschitz continuity are equivalent. Lastly, we present several practical applications that directly benefit the practitioner.

# References


[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

[2] C. Anil, J. Lucas, and R. Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[3] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[4] S. Biderman, U. PRASHANTH, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2023.

[5] S. Boucheron, G. Lugosi, and P. Massart. Concentration inequalities: A nonasymptotic theory of independence, 2013.

[6] N. Carlini, M. Nasr, C. A. Choquette-Choo, M. Jagielski, I. Gao, A. Awadalla, P. W. Koh, D. Ippolito, K. Lee, F. Tramer, et al. Are aligned neural networks adversarially aligned? *arXiv preprint arXiv:2306.15447*, 2023.

[7] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[8] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[9] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR, 2017.

[10] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing, 2019.

[11] Z. Cranko, Z. Shi, X. Zhang, R. Nock, and S. Kornblith. Generalised lipschitz regularisation equals distributional robustness. In *International Conference on Machine Learning*, pages 2178–2188. PMLR, 2021.

[12] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[13] L. Devroye, M. Lerasle, G. Lugosi, and R. I. Oliveira. Sub-gaussian mean estimators. 2016.

[14] N. Drenkow, N. Sani, I. Shpitser, and M. Unberath. A systematic review of robustness in deep learning for computer vision: Mind the gap?, 2022.

[15] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] F. Gama, J. Bruna, and A. Ribeiro. Stability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 68:5680–5695, 2020.

[18] S. Garg and G. Ramakrishnan. BAE: BERT-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *Proceedings of 3rd International Conference on Learning Representations*, 2014.

[20] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

[21] G. S. Halford, W. H. Wilson, and S. Phillips. Relational knowledge: the foundation of higher cognition. *Trends in Cognitive Sciences*, 14(11):497–505, 2010.

[22] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation, 2017.

[23] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, June 2021.

[24] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[25] H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

[26] Y. Huang, H. Zhang, Y. Shi, J. Z. Kolter, and A. Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. *Advances in Neural Information Processing Systems*, 34:22745–22757, 2021.

[27] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[28] R. Jia, A. Raghunathan, K. Göksel, and P. Liang. Certified robustness to adversarial word substitutions. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[29] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

[30] H. Kim, G. Papamakarios, and A. Mnih. The lipschitz constant of self-attention. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5562–5571. PMLR, 18–24 Jul 2021.

[31] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pages 656–672. IEEE, 2019.

[32] K. Leino, Z. Wang, and M. Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learning*, pages 6212–6222. PMLR, 2021.

[33] B. Li, C. Chen, W. Wang, and L. Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

[34] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*, 2020.

[35] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020.

[36] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*, 2020.

[37] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

[38] B. Lütjens, M. Everett, and J. P. How. Certified adversarial robustness for deep reinforcement learning. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1328–1337. PMLR, 30 Oct–01 Nov 2020.

[39] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[40] C. McDiarmid et al. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[41] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[42] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[43] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR, 2020.

[44] A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[45] Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[46] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang. Distributionally robust language modeling. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[47] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer. Training robust neural networks using lipschitz bounds. *IEEE Control Systems Letters*, 6:121–126, 2021.

[48] X. Qi, J. Wang, Y. Chen, Y. Shi, and L. Zhang. Lipsformer: Introducing lipschitz continuity to vision transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[49] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training, 2018.

[50] W. Ruan, X. Huang, and M. Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. *arXiv preprint arXiv:1805.02242*, 2018.

[51] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

[52] M. H. Stone. The generalized weierstrass approximation theorem. *Mathematics Magazine*, 21(5):237–254, 1948.

[53] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2014.

[54] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[55] Y. Tsuzuku, I. Sato, and M. Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[56] M. Usama and D. E. Chang. Towards robust neural networks with lipschitz continuity. In *Digital Forensics and Watermarking: 17th International Workshop, IWDW 2018, Jeju Island, Korea, October 22-24, 2018, Proceedings 17*, pages 373–389. Springer, 2019.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] A. Virmaux and K. Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[59] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing nlp, 2021.

[60] B. Wang, S. Wang, Y. Cheng, Z. Gan, R. Jia, B. Li, and J. Liu. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*, 2020.

[61] W. Wang, P. Tang, J. Lou, and L. Xiong. Certified robustness to word substitution attack with differential privacy. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1102–1112, Online, June 2021. Association for Computational Linguistics.

[62] A. Wei, N. Haghtalab, and J. Steinhardt. Jailbroken: How does llm safety training fail?, 2023.

[63] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon. Towards fast computation of certified robustness for ReLU networks. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5276–5285. PMLR, 10–15 Jul 2018.

[64] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pages 5276–5285. PMLR, 2018.

[65] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018.

[66] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

[67] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training, 2020.

[68] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

[69] X. Xu, L. Li, Y. Cheng, S. Mukherjee, A. H. Awadallah, and B. Li. Certifiably robust transformers with 1-lipschitz self-attention, 2023.

[70] J. Y. Yoo and Y. Qi. Towards improving adversarial training of nlp models, 2021.

[71] Y. Yoshida and T. Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[72] A. Zhang, A. Chan, Y. Tay, J. Fu, S. Wang, S. Zhang, H. Shao, S. Yao, and R. K.-W. Lee. On orthogonality constraints for transformers. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 375–382, Online, Aug. 2021. Association for Computational Linguistics.

[73] B. Zhang, D. Jiang, D. He, and L. Wang. Boosting the certified robustness of l-infinity distance nets. *arXiv preprint arXiv:2110.06850*, 2021.

[74] B. Zhang, D. Jiang, D. He, and L. Wang. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. *Advances in Neural Information Processing Systems*, 35:19398–19413, 2022.

[75] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

[76] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

# Appendix Table of Contents

## A Related Works

There have been extensive studies on developing robust neural networks with theoretical guarantees. These approaches with respect to our contribution can be organized into the following categories.

**Certified robustness with Lipschitz continuity.** The exploration of Lipschitz continuity as a cornerstone for improving model robustness has yielded significant insights, particularly in the domain of computer vision. This principle, which ensures bounded derivatives of the model's output with respect to its input, facilitates a smoother model behavior and inherently encourages robustness against adversarial perturbations. This methodology, initially suggested by [19], has since been rigorously analyzed and expanded upon. Most theoretical results in this area focus on certifying robustness with respect to the $\ell_2$-norm [9, 71, 20, 2, 32, 22, 3]. A recent push, fueled by new architectural developments, has also expanded these results into $\ell_\infty$-norm perturbations [74, 73, 75]. Further, Lipschitz continuity also serves practitioners as a computationally effective way to train more robust models [55, 65, 56, 11]. This stands in contrast to (virtual) adversarial training methods which brute-force the set of adversarial examples, then iteratively re-trains on them [42, 51, 67]. Though Lipschitz continuity has seen much success in continuous domains, it does not apply to non-metrizable domains such as language. Further, architectural limitations of prevalent models such as the Transformer [57, 30] exacerbate this problem. These challenges highlight a critical need for a new approach that can accommodate the specificities of discrete and non-metrizable domains while providing robustness guarantees.

**Achieving robustness in discrete/non-metrizable spaces.** Non-metrizable spaces, where it is non-trivial to construct a distance metric on the input/output domains, pose a unique challenge to certified robustness. Instead of focusing on point-wise perturbations, many studies have opted to examine how the output probability distribution of a model changes with respect to input distribution shifts by leveraging information bottleneck methods [54, 60, 46]. Most of these bounds lack granularity and cannot often be expressed in closed-form. In contrast to these theoretical approaches, recent efforts have refocused on directly adapting the principles underlying Lipschitz continuity to language. Virtual adversarial training methods such as [36, 70] mimic the measurement of Lipschitz continuity by comparing the textual embeddings with the KL-divergence of the output logits. Along these lines, techniques akin to those used in adversarial training in vision have also been translated to language, reflecting a shift towards robustness centered around the learned representation space [34, 18, 29]. Though these approaches have seen empirical success, they lack theoretical guarantees. As a result, their implementations and success rate is heavily task-dependent [36, 70]. There have also been attempts to mitigate the non-Lipschitzness of Transformers [72, 69] by modifying its architecture. These changes, however, add significant computational overhead.

**Other robustness approaches.** In parallel, other certified robustness approaches such as randomized smoothing [10, 33, 31] give state-of-the-art certification for $\ell_2$-based perturbations. Notable works such as [28, 61] have sought to generalize these techniques into language, but their guarantees strongly depend on the type of perturbation being performed. On the other hand, analytic approaches through convex relaxation inductively bound the output of neurons in a ReLU network across layers [66, 68, 64]. These works, however, are difficult to scale and also do not transfer easily to discrete/non-metrizable domains.

Our approach, inspired by Lipschitz continuity, distills the empirical intuition from the works of [36, 70] and provides theoretical certification guarantees independent of perturbation-type [28, 61] and domain modality. We demonstrate that knowledge continuity yields many practical applications analogous to Lipschitz continuity which are easy to implement and are computationally competitive.

## B Notations and Background

**Notations.** Let $\mathbb{R}^{\geq 0} := [0, \infty)$. For any function $f : \mathcal{X} \to \mathcal{Y}$, we denote graph$(f) := \{(x, y) \in \mathcal{X} \times \mathcal{Y} : f(x) = y\}$. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$. $(\mathcal{X}, \mathcal{F}_\mathcal{X}, \mathbb{P}_\mathcal{X})$, $(\mathcal{Y}, \mathcal{F}_\mathcal{Y}, \mathbb{P}_\mathcal{Y})$ are probability spaces and $(\mathcal{X} \times \mathcal{Y}, \mathcal{F}_\mathcal{X} \otimes \mathcal{F}_\mathcal{Y}, \mathbb{P}_\mathcal{X} \times \mathbb{P}_\mathcal{Y})$ denotes the product measurable space of the probability spaces $\mathcal{X}, \mathcal{Y}$. Since our contribution focuses on the supervised learning regime, we colloquially refer to $\mathcal{X}, \mathcal{Y}$ as the input and labels, respectively. We call any probability measure $\mathbb{P}_{\mathcal{X} \times \mathcal{Y}}$ absolutely continuous to $\mathbb{P}_\mathcal{X} \times \mathbb{P}_\mathcal{Y}$ (i.e. $\mathcal{D}_{\mathcal{X},\mathcal{Y}}(E) = 0$ for every $E \in \mathcal{X} \times \mathcal{Y}$ with $(\mathbb{P}_\mathcal{X} \times \mathbb{P}_\mathcal{Y})(E) = 0$) a data-distribution and denote it as $\mathcal{D}_{\mathcal{X},\mathcal{Y}}$. If $(\mathcal{Z}, d_\mathcal{Z})$ is a metric space with metric $d$ and $A \subset \mathcal{Z}$, then for any $z \in \mathcal{Z}$, $d_{\mathcal{Z}}(z, A) = \inf_{a \in A} d_{\mathcal{Z}}(a, z)$. We say that a metric space is bounded by some $B \in \mathbb{R}^{\geq 0}$, if $\sup_{x', x \in \mathcal{X}} d(x, x') < B$. Denote by $\text{Id}_{\mathcal{Z}} : \mathcal{Z} \to \mathcal{Z}$ the identity function of $\mathcal{Z}$. Let $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{\geq 0}$ be a loss function such that $\mathcal{L}(y, y') = 0$ if and only if $y = y'$. For any $f : \mathcal{X} \to \mathcal{Y}$ and $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$, we denote $\Delta\mathcal{L}_f^{(x,y)}(x', y') := |\mathcal{L}(f(x), y) - \mathcal{L}(f(x'), y')|$. Unless otherwise specified, it will be assumed that $f$ is a measurable function from $\mathcal{X}$ to $\mathcal{Y}$ with a metric decomposition (see Def. 1).

**Lipschitz continuity.** Given two metric spaces $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}})$ a function $f : \mathcal{X} \to \mathcal{Y}$ is $K$-Lipschitz continuous if there exists $K \in \mathbb{R}^{\geq 0}$ such that for all $x, x' \in \mathcal{X}$, $d_{\mathcal{Y}}(f(x), f(x')) \leq K d_{\mathcal{X}}(x, x')$.

# C Examples of Metric Decompositions

We present several common neural network architectures and some possible metric decompositions for them.

*Example* 1. Suppose that $f : \mathbb{R}^n \to \mathbb{R}^m$ is a fully-connected neural network with $n$ hidden layers, weight matrices $W_i$, biases $b_i$ for $i \in [n]$, and activation functions $\sigma_i$ for $i \in [n-1]$. Let $r(W_i)$ denote the number of rows of $W_i$. Then, for any $1 < p \leq \infty$ consider the set of metric spaces $(\mathbb{R}^n, \ell_p), (\mathbb{R}^{r(W_1)}, \ell_p), \ldots, (\mathbb{R}^{r(W_n)}, \ell_p), (\mathbb{R}^m, \ell_p)$, we define the metric decomposition of $h_i$ such that $h_0 : \mathbb{R}^n \to \mathbb{R}^{r(W_1)}$, $h_i : \mathbb{R}^{r(W_i)} \to \mathbb{R}^{r(W_{i+1})}$, and $h_n : \mathbb{R}^{r(W_n)} \to \mathbb{R}^m$. Each of these functions are simply the hidden layers in the fully-connected network. That is,

$$h_i(x) = \sigma_i(W_i x + b_i). \tag{C.1}$$

*Example* 2. If $f$ is a convolutional neural network, we can decompose it in the same way as before. Except, $h_i$ is now the convolution operation.

*Example* 3. We present two distinct metric decompositions of a residual network. Consider two fully-connected layers $A, B$ such that $x \xrightarrow{A} A(x) \xrightarrow{B} B(A(x)) \xrightarrow{x} B(A(x)) + x$. Here, the input $x$ feeds back into the output layer $B$ creating a residual block (the set of layers between the input and the residual connection).

We can aggregate each residual block as one metric decomposition. That is, let $h = B(A(x)) + x$. Then, $x \xrightarrow{h} h(x)$. Clearly, this is the same function as before; moreover, we yield the metric decomposition of $h(x)$. This is the approach we use in practice when dealing with residual connections. Moreover, this is also the standard way to count layers in computer vision and natural language processing.

We can also represent each layer within the residual block as a part of a metric decomposition. Define $A' : x \mapsto (A(x), x)$, $B' : (A(x), x) \mapsto (B(A(x)), x)$ and $x' : (B(A(x)), x) \mapsto (B(A(x)) + x, x)$. Then, it follows that $x \to A' \to B' \to x'$ forms a metric decomposition. Here, the metric in each layer is with respect to the quotient space where $(a, a') \sim (b, b')$ if and only if $a = b$. Therefore, we also recover the same vector space structure.

We again emphasize that any particular metric decomposition does not affect our theoretical results. Our propositions and theorems only rely on the fact that a metric decompositions exist.

# D Proof of Robustness

**Lemma D.1.** *Let $(X, d)$ be a metric space. Suppose that $x \in X$ and $f_x(z) = d(x, z)$. Then, $f_x$ is 1-Lipschitz with respect to the metric $d$. Moreover, if $A \subset X$ and $f_A(z) = \inf_{a \in A} d(x, a)$. Then, $f_A$ is also 1-Lipschitz.*

*Proof.* Fix some $x \in X$. By the definition of 1-Lipschitzness, it suffices to show that for all $z, y \in X$, $|f_x(z) - f_x(y)| \leq d(z, y)$. Thus,

$$\begin{aligned} |f_x(z) - f_x(y)| &= |d(x,z) - d(x,y)|, \\ &= |d(x,z) + d(z,y) - d(z,y) - d(x,y)|, \\ &\leq |d(x,y) - d(z,y) - d(x,y)|, \\ &\leq d(z,y). \end{aligned}$$

The latter statement follows from the same argument above with obvious modification. ∎

Next, we state the McDiarmid's Inequality [40] and Lévy's Inequalities [5] without proof.

**Definition 6.** *A function $f : X_1 \times X_2 \times \ldots \times X_n \to \mathbb{R}$ satisfies the bounded differences property if there are constants $c_1, c_2, \ldots, c_n$ such that for all $1 \leq i \leq n$ and $x_1 \in X_1, x_2 \in X_2, \ldots, x_n \in X_n$,*

$$\sup_{x_i' \in X_i} |f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i. \tag{D.1}$$

**Theorem D.2** (McDiarmid's Inequality)**.** *Assume that the function $f : X_1 \times X_2 \times \ldots \times X_n \to \mathbb{R}$ satisfy the bounded differences property with bounds $c_1, \ldots, c_n$. Consider the independent random variables $Y_1, \ldots, Y_n$ where $Y_i \in X_i$ for all $1 \leq i \leq n$. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}[f(Y_1, \ldots, Y_n) - \mathbb{E}[f(Y_1, \ldots, Y_n)] \geq \varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right), \tag{D.2}$$

$$\mathbb{P}[f(Y_1, \ldots, Y_n) - \mathbb{E}[f(Y_1, \ldots, Y_n)] \leq -\varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right). \tag{D.3}$$

Let $(X, d)$ be a metric space and $f : X \to \mathbb{R}$ be a Lipschitz function $f$ with Lipschitz constant $C$. Consider the measure space formed by $X$, the $\sigma$-algebra of all Borel sets of $X$ and a probability measure $\mathbb{P}$. Let $Y$ be a random variable taking values in $X$ and distributed according to $\mathbb{P}$.

**Definition 7** (Concentration Functions)**.** *For all $t > 0$, the concentration functions of $X$ is defined by*

$$\alpha(t) = \sup_{A \subset X : \mathbb{P}(A) \geq 1/2} \mathbb{P}[d(Y, A) \geq t], \tag{D.4}$$

*where $d(Y, A) = \inf_{x \in A} d(x, Y)$.*

Informally, the concentration function $\alpha(t)$ represents the scatter of the random variable $Y$ in the metric space. Specifically, for a fixed $t > 0$, if $\alpha(t) \approx 0$, then $Y$ is dispersed throughout the metric space since for any subset of $X$ that has significant probability mass, with high probability $Y$ is $t$ away from this subset.

**Theorem D.3** (Lévy's Inequalities)**.** *For any Lipschitz function $f$ with Lipschitz constant $C > 0$,*

$$\mathbb{P}[f(Y) \geq \mathbf{M}f(Y) + t] \leq \alpha\left(\frac{t}{C}\right) \quad \textit{and} \quad \mathbb{P}[f(Y) \leq \mathbf{M}f(Y) - t] \leq \alpha\left(\frac{t}{C}\right), \tag{D.5}$$

*where $\mathbf{M}f(Y)$ is the median of $f(Y)$. That is, $\mathbb{P}[f(Y) \geq \mathbf{M}f(Y)] \geq 1/2$ and $\mathbb{P}[f(Y) \leq \mathbf{M}f(Y)] \leq 1/2$.*

*Proof.* Here, we directly quote a proof from [5]. Consider the set $A = \{x \in X : f(x) \leq \mathbf{M}f(Y)\}$. By the definition of a median, $\mathbb{P}[A] \geq 1/2$. On the other hand, by the Lipschitz property of $f$,

$$A_t = \left\{x \in X : d(x, A) \leq \frac{t}{C}\right\} \subset \left\{x \in X : f(x) < \mathbf{M}f(Y) + \frac{t}{C}\right\}.$$

The inequalities now follow from the definition of the concentration function (the second follows from the first by considering $-f$). ∎

Leveraging Thm. D.2 and Thm. D.3, we can bound the distance between a fixed non-measure zero set and a point. Other than the boundedness of the metric space, we assume all the same notation as before.

**Lemma D.4.** *Suppose that $(X, d)$ is a bounded metric space such that $\sup_{x,x' \in X} d(x, x') < B$ for some $B > 0$. Let $A \subset X$ such that $\mathbb{P}[A] > 0$ and $\delta > 0$. Then,*

$$\mathbb{P}[d(Y, A) \geq \delta] \leq \exp\left(-\frac{2}{B^2}\left(\delta - B\sqrt{\frac{1}{2}\log\frac{1}{\mathbb{P}[A]}}\right)^2\right).$$

*Proof.* Let $f_A(x) = d(x, A)$. Then, by Lem. D.1, $f_A(\cdot)$ is 1-Lipschitz with respect to $d$. Since the metric space is bounded by constant $B > 0$, $f$ satisfies the bounded differences propety with constant $B$. By Theorem D.2,

$$\mathbb{P}[\mathbb{E}[f_A(Y)] - f_A(Y) \geq \delta] \leq e^{-2\delta^2/B^2}. \tag{D.6}$$

If $\delta = \mathbb{E}[f_A(Y)]$, then the left-hand side becomes $\mathbb{P}[f_A(Y) \leq 0] \geq \mathbb{P}[A]$. Therefore, by the previous inequality,

$$\mathbb{P}[A] \leq \mathbb{P}[f_A(Y) \leq 0], \tag{D.7}$$

$$\mathbb{P}[A] \leq \exp(-2\,\mathbb{E}[f_A(Y)]^2/B^2), \tag{D.8}$$

$$\mathbb{E}[f_A(Y)] \geq B\sqrt{\frac{1}{2}\log\frac{1}{\mathbb{P}[A]}}. \tag{D.9}$$

Therefore,

$$\mathbb{P}\left[d(Y, A) \geq \underbrace{\delta + B\sqrt{\frac{1}{2}\log\frac{1}{\mathbb{P}[A]}}}_{\text{(a)}}\right] \leq e^{-2\delta^2/B^2}. \tag{D.10}$$

The statement of the theorem then follows directly from a substitution of the term labeled (a) above. ■

Now, we are ready to prove the main regarding robustness (Thm. 2.1). For coherence, we restate the statement of the theorem before detailing the proof.

**Theorem.** *Let $A \subset \mathcal{X} \times \mathcal{Y}$ such that $\mathbb{P}_{\mathcal{X}}[A] > 0$ and $\delta, \eta > 0$. Let $A' = \{(x', y') \in \mathcal{X} \times \mathcal{Y} : \exists (x, y) \in A, \Delta\mathcal{L}_f^{(x,y)}(x', y') > \eta\}$. If $f : \mathcal{X} \to \mathcal{Y}$ is $\varepsilon$-knowledge continuous with respect to the hidden layer indexed by $k$ and $(\mathcal{Z}_k, d_k)$ is bounded by $B > 0$, then*

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}[A' \mid d_k(f^k(x), f^k(A)) < \delta] \leq \frac{\varepsilon\delta}{\eta\left(1 - \exp\left[-\Omega\left(\frac{\delta}{B} - \sqrt{\log\frac{1}{\mathbb{P}[A]}}\right)^2\right]\right)}. \tag{D.11}$$

*Proof.* By the definition of conditional probability, we have that

$$\mathbb{P}_{(x',y')\sim\mathcal{D}}\left[A' \mid d_k(f^k(x), f^k(x')) < \delta\right] = \frac{\mathbb{P}_{(x',y')\sim\mathcal{D}}\left[A' \text{ and } d_k(f^k(x), f^k(x')) < \delta\right]}{\mathbb{P}_{(x',y')\sim\mathcal{D}}[d_k(f^k(x), f^k(x'))]}. \tag{D.12}$$

We start by bounding the numerator of Eq. D.12. By the definition of $\varepsilon$-knowledge continuity,

$$\mathbb{E}\,\sigma_f^k(x, y) = \iint \frac{\Delta\mathcal{L}_f^{(x,y)}(x', y')}{d_k(f^k(x), f^k(x'))}\, d(\mathbb{P} \times \mathbb{P}), \tag{D.13}$$

$$\geq \iint_{d_k(f^k(x), f^k(x')) < \delta} \frac{\Delta\mathcal{L}_f^{(x,y)}(x', y')}{d_k(f^k(x), f^k(x')}\, d(\mathbb{P} \times \mathbb{P}), \tag{D.14}$$

$$\geq \frac{1}{\delta}\iint_{d_k(f^k(x), f^k(x')) < \delta} \Delta\mathcal{L}_f^{(x,y)}(x', y')\, d(\mathbb{P} \times \mathbb{P}), \tag{D.15}$$

$$\delta\,\mathbb{E}\,\sigma_f^k(x, y) \geq \iint_{\substack{d_k(f^k(x), f^k(x')) < \delta \\ (x,y)\in A}} \Delta\mathcal{L}_f^{(x,y)}(x', y')\, d(\mathbb{P} \times \mathbb{P}). \tag{D.16}$$

This gives us an upper-bound of expectation of $\Delta\mathcal{L}_f^{(x,y)}(x', y')$ over the set of all points that are within $\delta$-radius from $A$. Next, by Markov's inequality,

$$\mathbb{P}[A' \text{ and } d_k(f^k(x), f^k(x')) < \delta] \le \frac{\delta\,\mathbb{E}\,\sigma_f^k(x,y)}{\eta}, \tag{D.17}$$

$$\le \frac{\delta\varepsilon}{\eta}. \tag{D.18}$$

The last inequality follows from the fact that $f$ is $\varepsilon$-knowledge continuous. Now, by applying the complement of Lem. D.4, we lower-bound the denominator and yield the following

$$\mathbb{P}_{(x',y')\sim\mathcal{D}}\left[A' \mid d_k(f^k(x), f^k(x')) < \delta\right] \le \frac{\varepsilon\delta}{\eta\left(1 - \exp\left(-\frac{2}{B^2}\left(\delta - B\sqrt{\frac{1}{2}\log\frac{1}{\mathbb{P}[A]}}\right)^2\right)\right)}. \tag{D.19}$$

The proof is concluded by applying big-Omega notation to the exponentiated. ■

**Corollary D.5.** *If $(\mathcal{Z}_k, d_k)$ is unbounded, then*

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}[A' \mid d_k(f^k(x), f^k(A)) < \delta] \le \frac{\varepsilon\delta}{\eta(1-\mathbb{P}[A])}. \tag{D.20}$$

*If $\mathbb{P}[A] = 0$, then*

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}[A' \mid d_k(f^k(x), f^k(A)) < \delta] \le \frac{\varepsilon\delta}{\eta}. \tag{D.21}$$

*Proof.* These results follow from directly taking the limit as $B \to \infty$ and applying some of the bounds acquired in the proof of Thm. 2.1. This yields Eq. D.20. Next, setting $\mathbb{P}[A] = 0$ easily results in Eq. D.21. ■

## E Proof of Expressiveness

Here, we show the main result regarding the expressiveness of $\varepsilon$-knowledge continuous estimators (Prop. 2.3). For completeness, we restate the statement of the proposition before proceeding with the proof.

**Proposition.** *Suppose $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}}) := (\mathcal{X}, d_{\mathcal{X}})$ are **compact** metric spaces, $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ is the **set of all continuous functions** from $\mathcal{X}$ to $\mathcal{Y}$ such that $\int d_{\mathcal{X}}(x, x')^{-1} d\mu_f < \infty$ and $\mathcal{L}$ be Lipschitz continuous in both coordinates. Then, there exists a universal function approximator $\mathcal{U}$ of $\mathcal{F}$ that is knowledge continuous (i.e. $\mathbb{E}\,\sigma_f^k(x,y) < \infty$ for some $k$).*

*Proof.* By assumption, $\mathcal{X} = \mathcal{Y}$ and $d_{\mathcal{X}} = d_{\mathcal{Y}}$. First, we consider the set of all Lipschitz continuous functions from $\mathcal{X} \to \mathcal{X}$. Clearly, the set of all Lipschitz continuous functions separate points in $\mathcal{X}$ by the fact that the $d_{\mathcal{X}}$ is Lipschitz continuous (see Lem. D.1). Thus, since $\mathcal{X}$ is compact, by the Stone-Weierstrass Theorem [52] the set of Lipschitz continuous functions must be dense in the set of all continuous functions from $\mathcal{X}$ to $\mathcal{X}$. This implies that for any sequence $\varepsilon_1, \varepsilon_2, \ldots$ we can choose Lipschitz continuous functions $f_1, f_2, \ldots$ such that $\int \mathcal{L}(f_n(x), y)\, d\mu_f < \varepsilon_n$. It remains to show that each of these functions are in fact knowledge continuous. Since $\mathcal{X}$ is a metric space, we consider the trivial metric decomposition of our sequence of functions (see Remark **??**). Specifically, we denote

$h_1 = \mathrm{Id}_{\mathcal{X}}$ and proceed to bound $\mathbb{E}\,\sigma_f^1(x,y)$.

$$\mathbb{E}\,\sigma^1_{f_n}(x,y) = \iint \frac{\Delta\mathcal{L}^{(x,y)}_{f_n}(x',y')}{d_{\mathcal{X}}(x,x')}\,(d\mu_f \times d\mu_f), \tag{E.1}$$

$$\leq \iint \frac{|\mathcal{L}(f_n(x),y) - \mathcal{L}(f_n(x'),y) + \mathcal{L}(f_n(x'),y) - \mathcal{L}(f_n(x'),y')|}{d_{\mathcal{X}}(x,x')}\,(d\mu_f \times d\mu_f), \tag{E.2}$$

$$\leq \iint \frac{|\mathcal{L}(f_n(x),y) - \mathcal{L}(f_n(x'),y)|}{d_{\mathcal{X}}(x,x')}\,d(\mu_f \times \mu_f) \tag{E.3}$$

$$+ \iint \frac{|\mathcal{L}(f_n(x'),y) - \mathcal{L}(f_n(x'),y')|}{d(x,x')}\,(d\mu_f \times d\mu_f), \tag{E.4}$$

$$\leq \iint \frac{L d_{\mathcal{X}}(f(x),f(x'))}{d_{\mathcal{X}}(x,x')}\,d(\mu_f \times \mu_f) + \iint \frac{L d_{\mathcal{X}}(y,y')}{d_{\mathcal{X}}(x,x')}\,d(\mu_f \times \mu_f), \tag{E.5}$$

$$\leq \iint LK\,d(\mu_f \times \mu_f) + LB\int \frac{1}{d_{\mathcal{X}}(x,x')}\,d\mu_f, \tag{E.6}$$

$$= LK + LB\int d_{\mathcal{X}}(x,x')^{-1} d\mu_f, \tag{E.7}$$

where $L$ is the Lipschitz constant of $\mathcal{L}$, $K$ is the Lipschitz constant of the $f_n$, and $B$ bounds the metric space $\mathcal{X}$ (since any compact metric space is bounded). The remaining assumption in the proposition concludes the proof of the proposition. ■

# F Proof of Equivalence Between Lipschitz Continuity and Knowledge Continuity

We present the proofs of the results that establish conditions when knowledge continuity implies Lipschitz continuity and vice versa. As before, we restate all of the statements before providing their proof. First, we identify conditions under which knowledge continuity implies Lipschitz continuity (Prop. 2.4).

**Proposition.** *Suppose that $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}})$ are metric spaces. Let the first $n$ metric decompositions of $f : \mathcal{X} \to \mathcal{Y}$ be $K_i$-Lipschitz continuous, for $i \in [n]$. If $f$ is $\varepsilon$-knowledge continuous with respect to the $n^{th}$ hidden layer and $d_{\mathcal{Y}}(f(x), f(x')) \leq \eta \Delta\mathcal{L}^{(x,y)}_f(x',y)$ for all $x, x' \in \mathcal{X}$, $y \in \mathcal{Y}$, and some $\eta > 0$, then $f$ is Lipschitz continuous in expectation. That is,*

$$\mathbb{E}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{d_{\mathcal{Y}}(f(x),f(x'))}{d_{\mathcal{X}}(x,x')} \leq \varepsilon\eta\prod_{j=1}^{n} K_j. \tag{F.1}$$

*Proof.* We proceed to bound the knowledge continuity of $f$ from below.

$$\mathbb{E}\,\sigma^k_f(x,y) \geq \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}\,\mathbb{E}_{\substack{(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}\\ y'=y}} \frac{\Delta\mathcal{L}^{(x,y)}_f(x',y)}{d_k(f^k(x),f^k(x'))}, \tag{F.2}$$

$$\geq \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}\,\mathbb{E}_{\substack{(x',y')\sim\mathcal{D}\\ y'=y}} \frac{\Delta\mathcal{L}^{(x,y)}_f(x',y)}{\prod_{j=1}^n K_j d_{\mathcal{X}}(x',x)}, \tag{F.3}$$

$$\geq \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}}\,\mathbb{E}_{\substack{(x',y')\sim\mathcal{D}\\ y'=y}} \frac{\frac{1}{\eta} d_{\mathcal{Y}}(f(x),f(x'))}{\prod_{j=1}^n K_j d_{\mathcal{X}}(x,x')}, \tag{F.4}$$

$$= \mathbb{E}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{\frac{1}{\eta} d_{\mathcal{Y}}(f(x),f(x'))}{\prod_{j=1}^n K_j d_{\mathcal{X}}(x,x')}. \tag{F.5}$$

Eq. F.2 comes from the fact that we take the expectation only over pairs of points $(x,y),(x',y')$ where $y = y'$ and also because the summand is always nonnegative. Then, we inductively apply the definition of $K_i$-Lipschitz continuity to yield Eq. F.3. Eq. F.4 follows directly from the assumption in the statement of the proposition. Since the expression in Eq. F.4 now has no dependence on the label distribution, we may expand the expectation which results in Eq. F.5. Lastly, by the definition of $\varepsilon$-knowledge continuity,

$$\varepsilon \geq \mathbb{E}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{\frac{1}{\eta} d_{\mathcal{Y}}(f(x), f(x'))}{\prod_{j=1}^{n} K_j d_{\mathcal{X}}(x,x')},$$
$$\varepsilon\eta \prod_{j=1}^{n} K_j \geq \mathbb{E}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x,x')},$$

and this concludes the proof of the proposition. ∎

To prove Cor. 2.5, we need the following auxiliary result from [74].

**Proposition F.1.** *For a neural network $f : \mathbb{R}^n \to \mathbb{R}^K$ with Lipschitz constant $L$ under $\ell_p$-norm, define the resulting classifier $g$ as $g(x) := \arg\max_{k\in[K]} f_k(x)$ for an input $x$. Then, $g$ is provably robust under perturbations $\|\delta\|_p < \frac{\sqrt[p]{2}}{2L}$margin$(f(x))$, i.e.*

$$g(x+\delta) = g(x) \qquad \textit{for all } \|\delta\|_p < \frac{\sqrt[p]{2}}{2L}\textit{margin}(f(x)). \tag{F.6}$$

*Here, margin$(f(x))$ is the difference between the largest and second largeset output logit.*

The following proof is from [74].

*Proof.* Let $f_j(x)$ denote the $j^{\text{th}}$ coordinate of $f(x)$. We proceed by way of contraposition. Suppose that $g(x) \neq g(x+\delta)$ for some $\delta \in \mathbb{R}^n$. We show that $\|\delta\|_p \geq \frac{\sqrt[p]{2}}{2L}\text{margin}(f(x))$. Let $g(x) = \alpha$ and $g(x+\delta) = \beta$. Then,

$$\|f(x+\delta) - f(x)\|_p = \left(\sum_{k=1}^{K} |f_k(x+\delta) - f(x)_k|^p\right)^{1/p}, \tag{F.7}$$
$$\geq \left(|f_\alpha(x+\delta) - f_\alpha(x)|^p + |f_\beta(x+\delta) - f_\beta(x)|^p\right)^{1/p}. \tag{F.8}$$

The minimum of Eq. F.8 is achieved when $f_\alpha(x+\delta) = f_\beta(x+\delta) = (f_\alpha(x) + f_\beta(x))/2$. Then, through a direct substitution we have that

$$\|f(x+\delta) - f(x)\|_p \geq \frac{\sqrt[p]{2}}{2}(f_\alpha(x) - f_\beta(x)), \tag{F.9}$$

by the definition of margin$(f(x))$, $f_\alpha(x) - f_\beta(x) \geq \text{margin}(f(x))$. Lastly, by the definition of $L$-Lipschitz continuity, we have that

$$L\|\delta\|_p \geq \|f(x+\delta) - f(x)\|_p \geq \frac{\sqrt[p]{2}}{2}\text{margin}(f(x)). \tag{F.10}$$

Rearranging this expression results in the proposition. ∎

We are now ready for the proof of Cor. 2.5. We simply Prop. F.1 in conjunction with Markov's inequality to bound the Lipschitz constant.

**Corollary.** *Suppose that assumptions of Prop. 2.4 are true. And also assume that $(\mathcal{X}, d_{\mathcal{X}}) = (\mathbb{R}^n, \ell_p)$, $(\mathcal{Y}, d_{\mathcal{Y}}) = (\mathbb{R}^m, \ell_p)$, for $1 \leq p \leq \infty$. Define a classifier from $f : \mathbb{R}^n \to \mathbb{R}^m$, $g$, where $g(x) := \arg\max_{k\in[m]} f_k(x)$ for any $x \in \mathbb{R}^n$. Then, with probability $1 - \frac{\varepsilon\eta}{t}\prod_{j=1}^{n} K_j$, $g(x) = g(x+\delta)$ for all $\|\delta\|_p < \frac{\sqrt[p]{2}}{2t}$margin$(f(x))$ and $t > 0$. $f_k(x)$ is the $k^{th}$ coordinate of $f(x)$ and margin$(f(x))$ denotes the difference between the largest and second-largest output logits.*

*Proof.* By Prop. 2.4, we have that

$$\mathbb{E}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x,x')} \leq \varepsilon\eta \prod_{j=1}^{n} K_j. \tag{F.11}$$

By Markov's inequality,

$$\mathbb{P}_{(x,y),(x',y')\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \left[ \frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x,x')} \geq t \right] \leq \frac{\varepsilon\eta}{t} \prod_{j=1}^{n} K_j. \tag{F.12}$$

We yield the corollary by directly applying Prop. F.1 assuming that $f$ is $t$-Lipschitz continuous. ■

Next, we establish conditions under which Lipschitz continuity implies knowledge continuity (Prop. 2.6).

**Proposition.** *Let $(\mathcal{X}, d_{\mathcal{X}}), (\mathcal{Y}, d_{\mathcal{Y}})$ be a metric spaces. Let $f : \mathcal{X} \to \mathcal{Y}$ be $\varepsilon$-Lipschitz continuous and $\mathcal{L}(f(x), y)$ be $\eta$-Lipschitz continuous with respect to both coordinates. If the first $n$ metric decompositions of $f$ are $K_i$-Lipschitz continuous, then $f$ is knowledge continuous with respect to the $n^{th}$ hidden layer. That is,*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathcal{X},\mathcal{Y}}} \sigma_f^n(x,y) \leq \varepsilon\eta \prod_{j=1}^{n} \frac{1}{K_j}. \tag{F.13}$$

*Proof.* Let us start with the definition of $\varepsilon$-Lipschitz continuity and lower-bound it. For any $(x,y),(x',y') \in \mathcal{X}\times\mathcal{Y}$,

$$\frac{d_{\mathcal{Y}}(f(x), f(x'))}{d_{\mathcal{X}}(x,x')} \leq \varepsilon, \tag{F.14}$$

$$\frac{d_{\mathcal{Y}}(f(x), f(x'))}{\prod_{j=1}^{n} \frac{1}{K_j} d_k(f^k(x), f^k(x'))} \leq \varepsilon, \tag{F.15}$$

$$\frac{\frac{1}{\eta}|\mathcal{L}(x,y) - \mathcal{L}(x',y')|}{\prod_{j=1}^{n} \frac{1}{K_j} d_k(f^k(x), f^k(x'))} \leq \varepsilon, \tag{F.16}$$

$$\frac{|\mathcal{L}(x,y) - \mathcal{L}(x',y')|}{d_k(f^k(x), f^k(x'))} \leq \varepsilon\eta \prod_{j=1}^{n} \frac{1}{K_j}. \tag{F.17}$$

Eq. F.15 follows from inductively applying the definition of Lipschitz continuity on the metric decompositions of $f$. Specifically, $d_{i+1}(f^{i+1}(x), f^{i+1}(x')) \leq K_i d_i(f^i(x), f^i(x))$. Then, by the Lipschitz continuity of $\mathcal{L}$ in both coordinates we yield Eq. F.16. Since the Lebesgue integral preserves order, Eq. F.17 directly implies the statement of the proposition and this concludes the proof. ■

# G Predicting Adversarial Robustness with Volatility

As discussed in Section 3, we regress $k$-volatility scores for a variety of models across all layers against their empirical adversarial robustness. Herein, we describe this experimental procedure and detail the results. Throughout this section, we adopt the shorthand KVS $:= \mathbb{E}\,\sigma_f^k(x,y)$ and refer to this as the knowledge volatility score.

We run all our experiments against the IMDB dataset [39] with TextFooler [29] as the benchmark adversarial attack. We run linear regression to predict the number of successful adversarial attacks, using model type and model size. We then incorporate our vulnerability score, calculated over all layers, and notice how our $R^2$ changes.

For our linear regression, we use the LinearRegression class from sklearn (version 1.3.2), and default hyperparameters ($\alpha = 1.0$, max_iter = 1000). To calculate the number of adversarial attacks, we use TextFooler algorithm [29] on a holdout test set with respect to a pretrained model. We say that an adversarial attack is successful if the model previously characterized it correctly, but under the perturbation of TextFooler, the model now classifies it incorrectly.

For each model, we lay out our features as follows:

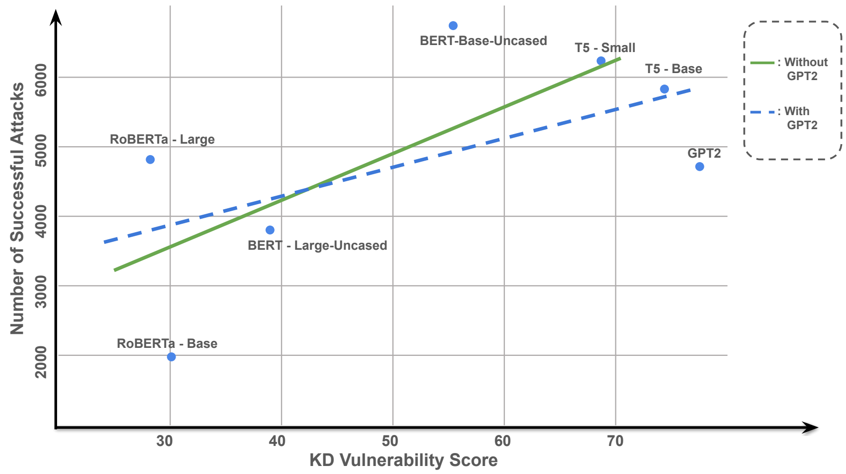


Figure 2: Regression analysis of Knowledge discontinuity vs. number of successful attacks under TextFooler. Knowledge discontinuities alone can explain 35% of the variance of successful adversarial attacks against a model ($R^2 = 0.35$). The line of best fit is given by $\mathit{SucessfulAttacks} = 49(\mathit{KVS}) + 2254$.

1. A 0 or 1 representing whether this model is encoder-only
2. A 0 or 1 representing whether this model is decoder-only
3. A 0 or 1 representing whether this model is encoder-decoder
4. A floating point representing the natural log of the number of parameters in this model
5. The vulnerability score associated with this model

For example, the following vector represents bert-large:

$$[1, 0, 0, 19.630, 54.044]$$

We choose to use the logarithm of the model size. Intuitively, we expect that past a certain size, a well-trained model will perform so well that it essentially masters the task, and there is little adversarial robustness to be gained by adding more parameters.

After running our linear regression, we proceed to obtain the coefficients, and then calculate the permutation importance of each of our features. We get the following results below for our coefficients:

| | Without vulnerability score | With vulnerability score |
|---|---|---|
| encoder | -548.43 | 1484.91 |
| decoder | -556.89 | -2816.49 |
| encoder-decoder | 1105.32 | 1331.57 |
| ln(num_params) | -362.59 | 65.50 |
| vulnerability score | N/A | 95.74 |

We calculate importance values using 100 random permutations. We ultimately get the following table:

| | Without vulnerability score | With vulnerability score |
|---|---|---|
| encoder | 0.0652 | 0.403 |
| decoder | 0.0195 | 0.712 |
| encoder-decoder | 0.177 | 0.291 |
| ln(num_params) | 0.0442 | -6.08e-05 |
| vulnerability score | — | 2.57 |
| $R^2$ | 0.282 | 0.479 |

Notice the importance of the vulnerability score, especially in proportion to the other features. Clearly, this illustrates both the predictive power and importance of our vulnerability score.

# H Localizing Volatile Hidden Representations

We seek to localize volatile hidden representations, both in the sense of which layers are more volatile, and which areas of the representation space for a given layer are more volatile. We consider the same selection of models in Appendix G, the same dataset (IMDB), and the same attack (TextFooler).

Throughout this section, we adopt the shorthand KVS $:= \mathbb{E}\, \sigma_f^k(x, y)$ and refer to this as the knowledge volatility score.

## H.1 Per-Layer Volatility

We start by plotting the KVS for each of our models, against the actual number of successful adversarial attacks. We use this as a proxy for analyzing volatility, since the more volatile, the higher the correlation between these two variables.

Then, to analyze this on a per-layer basis, we notice that KVS can be calculated independently for any given layer, since each layer emits its own distance metric.

Thus, we ultimately plot $R^2$ vs relative depth for our given models. We notice that the foremost and final hidden layers are most explanatory (see Fig. 4). However, we see that GPT2 admits a surprising behavior, in that its middle hidden layers are most participatory in adversarial vulnerability. We now specifically look at this as a case study. To do this, we repeat the experiments in Appendix G across 10 relative depths and plot the $R^2$ with and without GPT2 (see Fig. 3). Indeed, without GPT2 we see that the trend of $R^2$ seems to be more linear. These results directly inform the choice of hyperparameters in Appendix I since we want to minimize KVS only over the highly salient layer, rather than all of them.

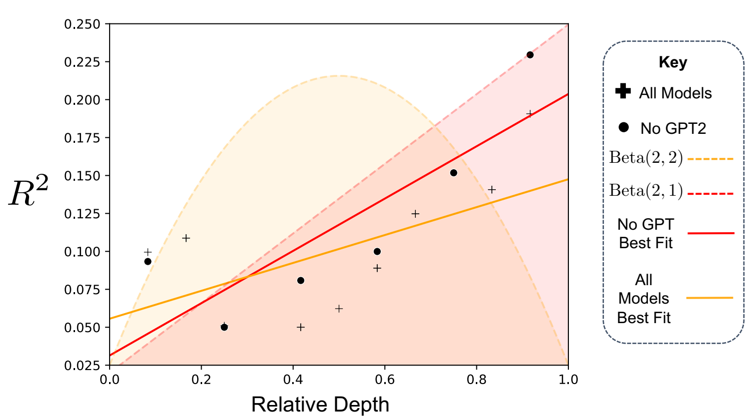


Figure 3: The explained variance of knowledge continuities for each relative depth across all models and without GPT2. The distribution of points warrant the use of various parameterizations of the Beta distribution in Alg. 2 for different models.

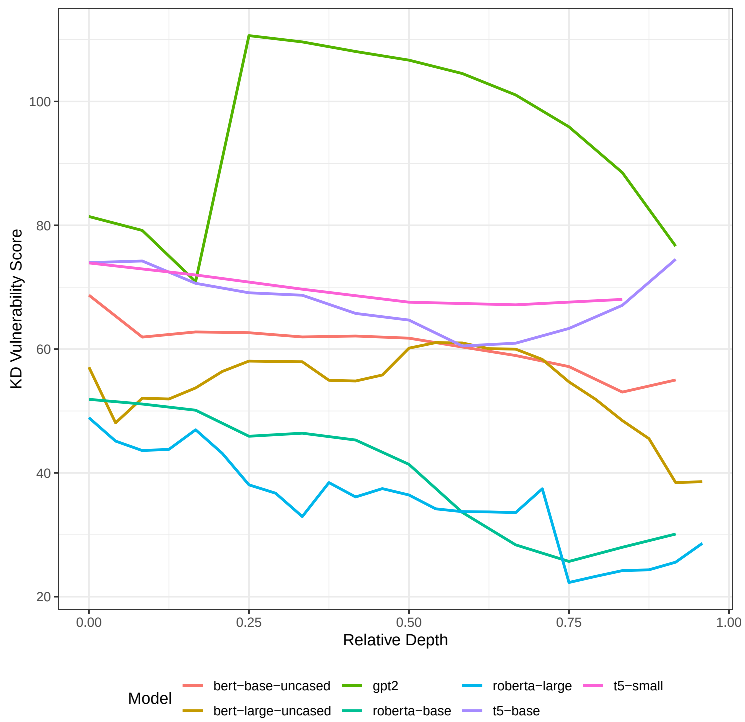


Figure 4: The KVS versus relative depth for BERTBase-Uncased, BERT-Large-Uncased, T5-Base, T5-Small,RoBERTa-Base, RoBERTa-Large, GPT2. Notice that we can track which layers are responsible for what portion of the vulnerability score of each model. Notice that GPT2 has a spike toward the middle, and teeters out toward the end– perhaps this is because the deeper layers are responsible for decoding, and have less of an effect on classification performance. Such a plot could be a useful for both practical applications and future research, as a computationally efficient method to roughly assess how different layers may contribute to adversarial vulnerability.

### H.2 Per-Model Volatility

We start by exploring the KVS of each of our test models. We notice that KVS cannot be predicted by surface-level features such as size or model type alone. This is shown clearly in Fig. 5. Yet, as discussed in Appendix G, it is still able to predict actual adversarial vulnerability with moderate power. Thus, we conjecture that KVS captures a complex aspect of the model's vulnerability which cannot be solely attributed to its size or type.

## I Regularizing Knowledge Continuity

In this section, we provide a comprehensive overview of regulating knowledge continuity to achieve robustness. We first show a simple algorithm that estimates $k$-volatility. Then, we demonstrate how this can be used to augment any loss function to achieve regularization. We present some theoretical guarantees that revolve around the unbiasedness of our estimation algorithm and some guarantees of its rate of convergence. Lastly, we present detailed discussion of the results shown in Table 1 including training details and ablation studies over the hyperparameters.

### I.1 Estimating Knowledge Continuity Algorithmically

We first present a method for estimating the knowledge continuity of a hidden representation space. This is shown in Alg. 1. In the following subsection, we provide some guidance to choosing the subsampling hyperparameter $M$. In theory, one should choose $M = N$. However, if $N \gg 1$, this can become quickly intractable. Therefore, we multiplicatively bound the error of the unbiased estimator with respect to $M$ and the variance of $k$-volatility. As discussed in the main text, the choice of metric

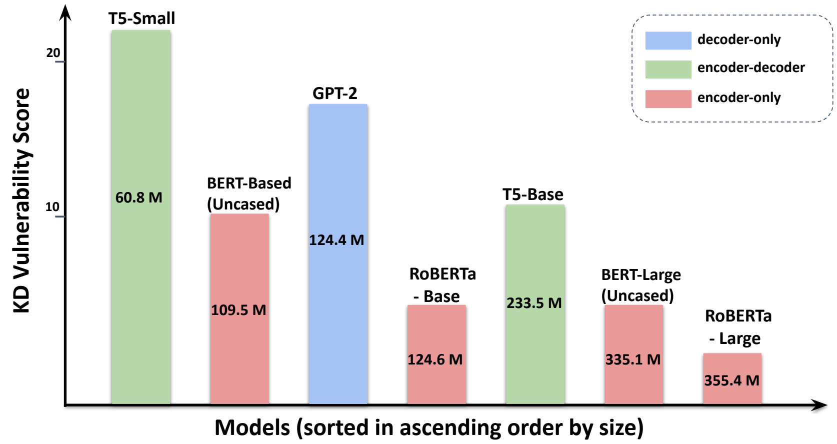


Figure 5: The KVS of each model, in the ascending order of model size. As shown, a model's KVS cannot be solely attributed to its size or type.


Figure 6: Left: Actual Adversarial Attacks, Right: Predicted Vulnerabilities using KVS

(or representation space) which we enforce knowledge continuity against is crucial as it determines the type of robustness we will achieve. Therefore, in Alg. 2, we incorporate this detail by sampling the index of the hidden layer using some Beta distribution specified by hyperparameters $\alpha, \beta$. Note that we choose the Beta distribution for simplicity, however, it can be replaced by any distribution like a mixture of Gaussians.

In contrast to existing adversarial training methods such as [26] and [51] which only use the embeddings, our algorithm gives the practitioner more control over which hidden layer (or distance metric) to enforce smoothness. In this way, if the practitioner has some knowledge *a priori* of the attacker's strategy, they may choose to optimize against the most suitable metric. We present a brief discussion of the various tradeoffs when choosing $\alpha, \beta$ in the following section as well as a detailed empirical analysis in the following subsections. $\lambda$ is the weight we put on the regularizer in relation to the loss function $\mathcal{L}$. We provide a detailed ablation study of the effects of $\lambda$ in the following subsections. We surprisingly find that even for $\lambda \ll 1$ we can achieve significant edge in terms of robustness over existing methods. This is in contrast to virtual adversarial training methods such as [36] which requires applying a $\lambda$-value magnitudes larger. Moreover, for larger $\lambda$, we find that the accuracy of the model is not compromised. This provides some empirical support for Theorem 2.2.

**Algorithm 1** Estimating knowledge continuity.

**Input:** A batch of $N$ data points $\{(x_i, y_i)\}_{i=1}^N$, $M \le N$, neural network $f$ with $n$ hidden layers, and some $k \in [n]$
**Output:** An estimation of $\mathbb{E}\,\sigma_f^k(x, y)$.
Subsample $M$ indices $n_1, \ldots, n_m$ uniformly at random from $[N]$ without replacement
$\sigma_f^k \leftarrow 0$
Losses $\leftarrow \{\mathcal{L}(f(x_{n_i}), y_{n_i})\}_{i=1}^M$
**for** $(i, j) \in [M] \times [M]$ **do**
  Dist $\leftarrow d_k(f^k(x_{n_i}), f^k(x_{n_j}))$
  $\sigma_f^k \leftarrow \sigma_f^k + |\text{Losses}_i - \text{Losses}_\ell| / \textsc{Dist}$
**end for**
**return** $\sigma_f^k$

**Algorithm 2** Regularization of knowledge continuity.

**Input:** $\alpha, \beta, M, \lambda > 0$.
A neural network $f$ with $n$ hidden layers, loss function $\mathcal{L}$, and batch $\{(x_i, y_i)\}_{i=1}^N$.
**Output:** Loss with added knowledge continuity regularization score.
$X \sim \text{Beta}(\alpha, \beta)$
$\sigma_f^k \leftarrow$ (Alg. 1)$(f, M, k \leftarrow \max(\lfloor Xn \rfloor, 1))$
**return** $\frac{1}{N} \sum_{i=1}^N \mathcal{L}(f(x_i), y_i) + \frac{1}{M^2} \lambda \sigma_f^k$

### I.2 Theoretical Guarantees of Knowledge Continuity Regulation

In this subsection, we demonstrate that Alg. 1 is indeed an unbiased estimator for knowledge continuity and also provide some bounds on the rate of convergence of this estimation.

**Proposition I.1** (Alg. 1 is an Unbiased Estimator)**.** *Assuming that each data point in the batch, $\{(x_i, y_i)\}_{i=1}^N \sim \mathcal{D}_{\mathcal{X},\mathcal{Y}}$, is sampled i.i.d., then Alg. 1 is an unbiased estimator for $\mathbb{E}\,\sigma_f^k(x, y)$.*

*Proof.* Let $\hat{\theta}$ be the random variable representing the output of Alg. 1. It suffices to show that

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\,\sigma_f^k(x, y),$$

where the expectation on the left-hand side is taken over the set of all batches. By the definition of Alg. 1,

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\left( \sum_{i=1}^M \sum_{j=1}^M \frac{1}{M^2} \frac{\Delta \mathcal{L}_f^{(x_{n_j}, y_{n_j}) x_{n_i}, y_{n_i}}}{d_k(f^k(x_{n_i}), f^k(x_{n_j}))} \right), \tag{I.1}$$

$$= \sum_{i=1}^M \sum_{j=1}^M \frac{1}{M^2} \mathbb{E}\left( \frac{\Delta \mathcal{L}_f^{(x_{n_j}, y_{n_j}) x_{n_i}, y_{n_i}}}{d_k(f^k(x_{n_i}), f^k(x_{n_j}))} \right), \tag{I.2}$$

$$= \mathbb{E}\,\sigma_f^k(x, y). \tag{I.3}$$

The second equality follows from the linearity of expectation. ■

Next, we state a proposition that gives some theoretical guidance for choosing the hyperparameter $M$ in Alg. 1. In practice, one should choose $M$ to be the batch size as to ensure accurate estimation of the knowledge discontinuity score. We recognize, however, that if $N \gg 1$, choosing $M = N$ may be intractable. We multiplicatively bound the error of the unbiased estimator with respect to $M$ and the overall variance of the $\delta$-knowledge discontinuity.

**Definition 8.** *A random variable $\hat{\theta}$ is an $(\varepsilon, \delta)$-multiplicative estimator of a random variable $\theta$ if*

$$\mathbb{P}[\hat{\theta} \notin (1 \pm \varepsilon)\theta] \le 1 - \delta.$$

The next result is a well-known result from [44] with applications found in [13] and [25].

**Theorem I.2** (Median of Means)**.** *Given $\varepsilon, \delta > 0$, and an unbiased estimator $\theta$, $\hat{\theta}$. We can achieve an $(\varepsilon, \delta)$-multiplicative estimator of $\theta$ with $K$ independent samples of $\hat{\theta}$ where*

$$K = O\left(\frac{\mathrm{Var}(\hat{\theta})}{(\varepsilon \mathbb{E}\hat{\theta})^2} \ln \frac{1}{\delta}\right),$$

*where $\mathrm{Var}(\hat{\theta})$ is the variance of the estimator $\hat{\theta}$.*

**Proposition I.3.** *Suppose $\varepsilon, \delta, \delta' > 0$, then we can achieve an $(\varepsilon, \delta)$-multiplicative estimator of the $\delta'$-knowledge discontinuity in layer $j$ with $M = \Theta(K)$ using Alg. 1 where*

$$K = O\left(\frac{\delta' \, \mathrm{Var}(KD)}{(\varepsilon \mathbb{E}[\Delta \mathcal{L}(f; x, y)])^2} \ln \frac{1}{\delta}\right),$$

*where $\Delta \mathcal{L}(f; x, y)$ difference in loss of $f$ on any two data points sampled from $\mathcal{D}$ and $KD$ is the random variable that represents the $\delta'$-knowledge discontinuities across $\mathcal{D}$.*

*Proof.* Consider a variation of the algorithm where we only draw a pair of points. In other words, fix $M = 2$. Denote the two data points we are considering to be $(x_1, y_1), (x_2, y_2)$. Then, let

$$X := \begin{cases} \frac{|\mathcal{L}(f; x_1, y_1) - \mathcal{L}(f; x_2, y_2)|}{d_j(h_j(x), h_j(x'))}, & \text{if } \|h_j(x_1) - h_j(x_2)\| < \delta', \\ 0 & \text{o/w.} \end{cases}$$

Since we've already shown that $X$ is an unbiased estimator (see Prop. I.1) of the $\delta'$-knowledge discontinuities, it remains to find the variance and squared expectation and apply the Median of Means theorem (see Theorem I.2). First, we lower bound $(\mathbb{E}X)^2$:

$$\begin{aligned}
(\mathbb{E}X)^2 &= \left(\frac{1}{2} \int_{\mathcal{D}|_X} \mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_x}} \left[\frac{|\mathcal{L}(f; x, y) - \mathcal{L}(f; x', y')|}{d_j(h_j(x), h_j(x'))}\right] d\mu_X\right)^2, \\
&\qquad \text{(from Prop. I.1)} \\
&= \frac{1}{4} \int_{\mathcal{D}|_X \times \mathcal{D}|_X} \mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_{x_1}}} \left[\frac{|\mathcal{L}(f; x_1, y_1) - \mathcal{L}(f; x', y')|}{d_j(h_j(x), h_j(x'))}\right] \cdot \\
&\quad \mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_{x_2}}} \left[\frac{|\mathcal{L}(f; x_2, y_2) - \mathcal{L}(f; x', y')|}{d_j(h_j(x), h_j(x'))}\right] d\mu_X(x_1) d\mu_X(x_2), \\
&\geq \frac{1}{4\delta^2} \int_{\mathcal{D}|_X \times \mathcal{D}|_X} \mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_{x_1}}} \left[|\mathcal{L}(f; x_1, y_1) - \mathcal{L}(f; x', y')|\right] \cdot \\
&\quad \mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_{x_2}}} \left[|\mathcal{L}(f; x_2, y_2) - \mathcal{L}(f; x', y')|\right] d\mu_X(x_1) d\mu_X(x_2), \\
&\qquad \text{(since } d_j(h_j(x), h_j(x'))\text{)} \\
&\geq \frac{1}{4\delta^2} \int_{\mathcal{D}|_X} \left(\mathbb{E}_{(x', y') \sim \mathcal{D}|_{V_x}} \left[|\mathcal{L}(f; x, y) - \mathcal{L}(f; x', y')|\right]\right)^2 d\mu_X(x), \\
&\qquad \text{(only consider terms where } x_1 = x_2\text{)} \\
&\geq \frac{1}{4\delta^2} \int_{\mathcal{D}|_X} \left(\int_{\mathcal{D}|_{V_x}} |\mathcal{L}(f; x, y) - \mathcal{L}(f; x', y')|^2\right) d\mu_X(x), \\
&\qquad \text{(only consider terms in the product that agree)} \\
&\geq \frac{1}{4\delta^2} \int_{\mathcal{D}|_X \times \mathcal{D}|_X} |\mathcal{L}(f; x, y) - \mathcal{L}(f; x', y')|^2 \chi_\delta(x, x') \, d\mu(x) d\mu(x')
\end{aligned}$$

which follows from Tonelli's theorem and $\chi_\delta(x, x') = 1$ if and only if $d_j(h_j(x), h_j(x')) < \delta$ and 0 otherwise. Then, by symmetry, this is equivalent to

$$= \frac{\mathbb{E}[\Delta \mathcal{L}(f; x, y)]^2}{4\delta^2}.$$

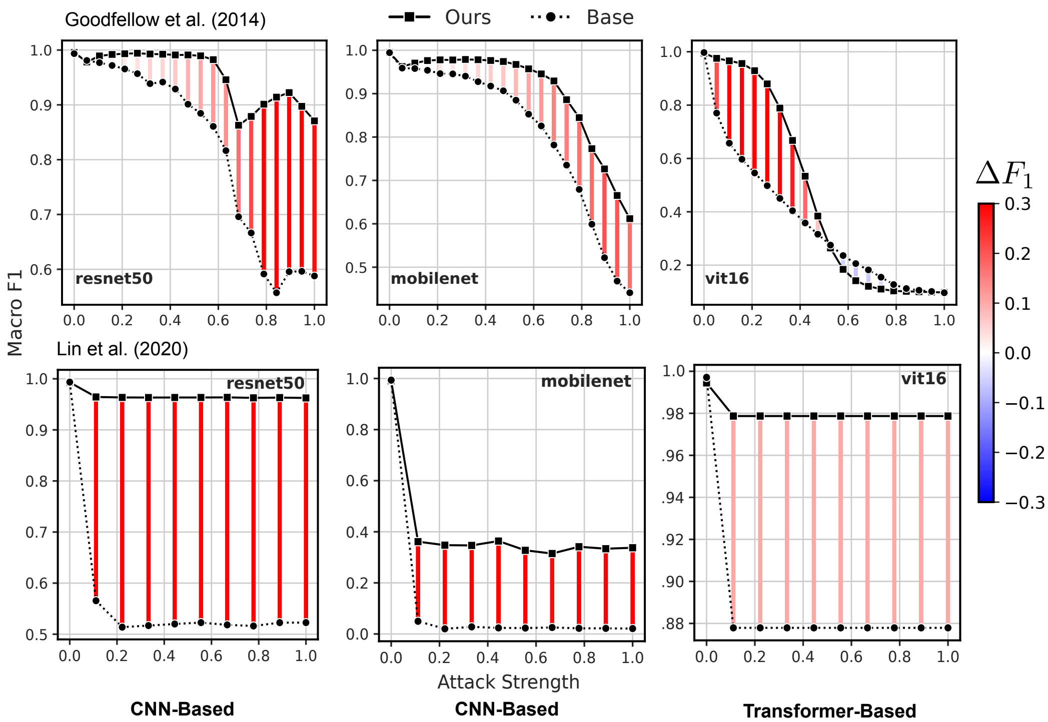


Figure 7: Regulating knowledge continuity on a host of vision models (ResNet50, MobileNetV2, and ViT16). Base models are trained with cross-entropy loss. KCReg (Our) models are finetuned with the additional regularization objective described in Alg. 2. Two adversarial attacks are then performed: the fast-gradient sign method from [19], and an iterative attack SI-NI-FGSM from [35]. We see that regulating knowledge continuity consistently improves/stabilizes robustness. Performance is measured using F1 and the attack strength corresponds to the maximum perturbation magnitude in L2 allowed. Since the pixel values of the images are bounded between $[0, 1]$, we also constrain the attack strength to be between $[0, 1]$.

The last equality follows from the fact that Now, we bound the variance of the estimator by above:

$$\begin{aligned}
\operatorname{Var} X &= \mathbb{E}X^2 - (\mathbb{E}X)^2, \\
&= \int_{\mathcal{D}|_X \times \mathcal{D}|_X} \frac{|\mathcal{L}(f; x_1, y_1) - \mathcal{L}(f; x_2, y_2)|^2}{d_j(h_j(x), h_j(x'))^2}\, d\mu(x_1) d\mu(x_2) - (\mathbb{E}\mathrm{KD})^2 && \text{(from Prop. I.1)} \\
&= \mathbb{E}\mathrm{KD}^2 - (\mathbb{E}\mathrm{KD})^2 = \operatorname{Var}(\mathrm{KD}).
\end{aligned}$$

Thus, combining both expressions with Theorem. I.2 we yield the desired result. ■

### I.3 Regulating Knowledge Continuity "In the Wild"

We compare our regularization algorithm with several state-of-the-art adversarial and virtual adversarial training algorithms. These results are presented in Table. 1. Additional experiments on MNIST are also performed. These are presented in Fig. 7.

### I.4 Ablation Studies

Herein, we present ablation studies for the crucial hyperparameters in our regularization algorithm, Alg. 2: $\lambda$ which is the weight we assign the knowledge continuity regulation loss and $(\alpha, \beta)$ which determines the sampling behavior of the index of the hidden representation space.

Table 1: Comparison of our knowledge continuity algorithm to existing works across various model families and adversarial attack methods. TF, BA, ANLI denote adversarial attacks [29], [34], and [45], respectively. Regulating knowledge continuity to improve robustness is superior across almost all tasks and attacks.

| Arch. | Method | IMDB | $\text{IMDB}_{\text{TF}}$ | $\text{IMDB}_{\text{BA}}$ | $\text{ANLI}_{\text{R1}}$ | $\text{ANLI}_{\text{R2}}$ | $\text{ANLI}_{\text{R3}}$ |
|---|---|---|---|---|---|---|---|
| | Base | 93.6 | 47.9 | 45.2 | 44.5 | 45.6 | 33.8 |
| BERT | TF | 93.3 | 69.2 | 62.5 | ✗ | ✗ | ✗ |
| ~110M params | ALUM | 93.5 | 56.9 | 47.8 | 45.2 | 46.7 | 46.3 |
| | **KCReg (ours)** | **94.8** | **75.1** | **84.9** | **45.6** | **46.9** | **45.3** |
| | Base | 93.6 | 63.9 | 54.9 | 42.7 | 44.9 | 43.4 |
| GPT2 | TF | 92.0 | 64.5 | 51.3 | ✗ | ✗ | ✗ |
| ~1.5B params | ALUM | 94.9 | 49.4 | 27.5 | 43.8 | 45.2 | 44.6 |
| | **KCReg (ours)** | **94.9** | **87.8** | **90.6** | **47.1** | **48.1** | **44.7** |
| | Base | 93.7 | 53.9 | 39.3 | 46.1 | 44.7 | 46.0 |
| T5 | TF | 96.8 | 77.8 | 60.6 | ✗ | ✗ | ✗ |
| ~220M params | ALUM | 95.1 | 67.1 | 51.9 | 44.5 | 44.8 | 44.4 |
| | **KCReg (ours)** | **94.9** | **89.3** | **91.3** | **48.2** | **45.0** | **44.3** |

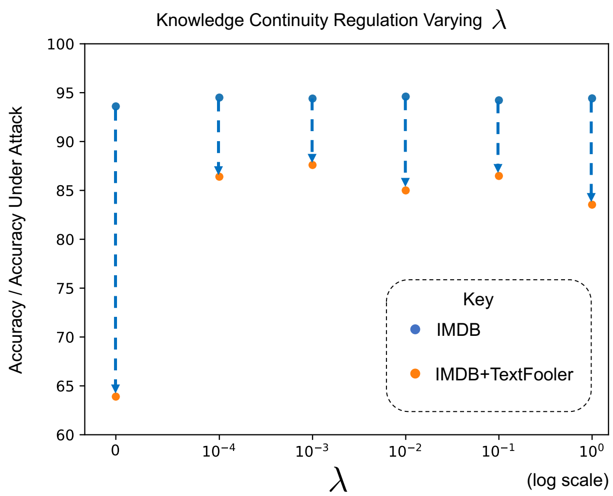


Figure 8: The accuracy of the model (both not under/under adversarial attack) on the IMDB dataset versus varying the weight given to the knowledge continuity regularization term ($\lambda$).

**Ablation Study of $\lambda$.** The weight given to the regularizer ($\lambda$) is ablated over, with the results shown in Fig. 8. For any positive $\lambda$, there is an immediate large improvement in adversarial robustness. Next, as $\lambda$ is systematically increased by factors of 10, we do not see a significant change in the accuracy (not under attack). This corroborates Theorem. 2.2, as it demonstrates that regulating knowledge discontinuities (no matter how strongly) is not at odds with minimizing the empirical risk of our model. On the other hand, we also do not see a significant increase in adversarial robustness as $\lambda$ increases. This may imply that we have reached the threshold of adversarial robustness under TextFooler [29]. Specifically, the adversarial attacks generated by TextFooler may not be valid in that they have flipped the ground-truth label. Therefore, we believe that a good $\lambda$ for this particular application should lie somewhere between 0 and $1 \times 10^{-4}$.

**Ablation Study of** $(\alpha, \beta)$ In this subsection, we briefly discuss how the $\alpha, \beta$ hyperparameters which determine the shape of the Beta distribution in Alg. 2 affect the final performance and robustness of our model on the IMDB dataset. Recall that the shape of the Beta distribution determines the index of the hidden layers we are using the compute the knowledge continuity. Thus, they are crucial in determining the behavior of our regularizer.

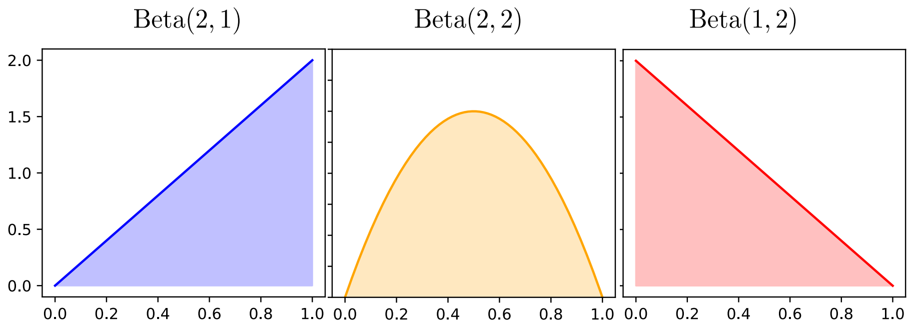


Figure 9: The Beta distributions that we ablated over with the probability density function of their parameterizations shown.

We finetune {BERT, T5, GPT2} models on the IMDB dataset with the hyperparameters described in the next subsection. The results are displayed in Table 2. Across all models we observe a decrease in robustness for $\alpha = 1, \beta = 2$. These values correspond to a right-skewed distribution which places high sampling probability on the earlier (closer to the input) hidden layers. Intuitively, perturbations in the early layers should correspond to proportional textual perturbations in the input text. Pure textual perturbations with respect to some metric like the Levenshtein distance should be only loosely if not completely (un)correlated with the actual labels of these inputs. Therefore, enforcing knowledge continuity with respect to this metric should not see increase robustness. Moreover, we also observe a larger decrease in accuracy (not under attack) with the same parameters. This suggests that maintaining this sort of knowledge continuity in the earlier layers is harder to converge on and there may be a "push-and-pull" behavior between optimizing knowledge continuity and accuracy (not under attack). Surprisingly, we observe no significant difference between the other $\alpha, \beta$ values shown in the table.

Table 2: We train finetune {BERT, T5, GPT2} using knowledge continuity regularization, as described in Alg. 2. We varied the $\alpha, \beta$ hyperparameters for the Beta distribution as to determine the effect of these parameters on model performance and robustness. The rows of the table are labeled with the format: Model+Reg$_{(\alpha,\beta)}$. The bolded entries of the table correspond to the best performing metrics out of the knowledge continuity regulated models.

| Model | IMDB | IMDB$_{\text{TF}}$ |
|---|---|---|
| BERT$_{\text{BASE}}$ | 93.6 | 47.9 |
| BERT$_{\text{BASE}}$+Reg$_{(2,1)}$ | **94.8** | **75.1** |
| BERT$_{\text{BASE}}$+Reg$_{(2,2)}$ | 89.2 | 74.1 |
| BERT$_{\text{BASE}}$+Reg$_{(1,2)}$ | 87.0 | 68.2 |
| GPT2 | 93.6 | 63.9 |
| GPT2+Reg$_{(2,1)}$ | 94.6 | 85.0 |
| GPT2+Reg$_{(2,2)}$ | **94.9** | **87.8** |
| GPT2+Reg$_{(1,2)}$ | 93.1 | 84.9 |
| T5$_{\text{BASE}}$ | 93.7 | 53.9 |
| T5$_{\text{BASE}}$+Reg$_{(2,1)}$ | **95.0** | 88.9 |
| T5$_{\text{BASE}}$+Reg$_{(2,2)}$ | 94.9 | **89.3** |
| T5$_{\text{BASE}}$+Reg$_{(1,2)}$ | 94.6 | 88.1 |

We did not formally benchmark other configurations of $\alpha, \beta$ such as increasing their magnitude to impose a sharper distribution. During training, we noticed that using these sharper distributions both significantly slowed the model's convergence and decreased the model's accuracy (not under attack). It could be that though knowledge continuity itself is a *local* property the enforcement of this *local* property requires change on a *global* scale. In other words, one cannot simply reduce the knowledge discontinuities or uniformly converge with respect to one layer without participation from other layers. The extent to which other layers are involved in the regularization of a specific one is an interesting question that we leave for future research.

### I.5 Training Details

In this section, we describe in detail the training objectives, procedures, algorithms, and hyperparameters that we used in the main text and further experiments done in the appendix.

**Brute-Force Adversarial Training.** For all models undergoing adversarial training, we first finetune the model against the training set. Then, attack it using the TextFooler [29] algorithm with examples from the training set. After the attacks are concluded, we then incorporate the text of successful adversarial attacks back into the training set and proceed to finetune again. This procedure iteratively continues. For the sake of computational efficiency, for all models we applied this procedure once. The parameters we are using during the adversarial attack is the same hyperparameters we actually use at test-time. Specifically, we impose a query budget of 300 queries.

**Plain Finetuning on IMDB.** The IMDB dataset consist of 50,000 examples with 25,000 for training and 25,000 for testing. We split the test set 40%-60% to create a validation and test set of 10,000 and 15,000 examples, respectively. Examples were sampled uniformly at random during the splitting process. Since adversarial attacks were costly, we uniformly subsampled 5,000 examples from this 15,000 to benchmark robustness in the experiments related to the regularizer. However, for the experiments estimating the knowledge vulnerability score, we performed adversarial attacks on all 15,000 datapoints in the test set. We found no significant difference between robustness estimation on this 5,000 subsample versus and the entire 15,000 dataset.

We train all models using the following hyperparameter and optimizer configurations:

Table 3: Training hyperparameters and optimizer configurations for finetuning models {BERT, GPT2, T5} on IMDB without any form of regularization or adversarial training.

| HYPERPARAMETER | VALUE |
|---|---|
| OPTIMIZER | ADAM |
| ADAM $\beta_1$ | 0.9 |
| ADAM $\beta_2$ | 0.999 |
| ADAM $\varepsilon$ | $1 \times 10^{-8}$ |
| MAX GRADIENT NORM | 1.0 |
| LEARNING RATE SCHEDULER | LINEAR |
| EPOCHS | 20 |
| BATCH SIZE | 32 |
| LEARNING RATE | $5 \times 10^{-5}$ |
| WEIGHT DECAY | $1 \times 10^{-9}$ |

**Knowledge Discontinuity Regulation on IMDB.** For enforcing the knowledge discontinuity on IMDB, we use a constant $\lambda = 1 \times 10^{-2}$ for all models. As shown in Table 2, we varied $\alpha, \beta \in \{1, 2\} \times \{1, 2\}$ and displayed the best models in terms of robustness in Table. 1 in the main text. We train all models for 50 epochs. Other than that all the other hyperparameters and optimizer configurations are the same as regular finetuning (see Table 3).

**Knowledge Discontinuity Regulation on ANLI.** Optimizing over the ANLI dataset was significantly harder than on IMDB. As a result, for each model class {BERT, GPT2, T5} we performed a quick hyperparameter search over $\lambda$ ($1 \times 10^{-4}$), the learning rate ($5 \times 10^{-5}$), and weight decay ($1 \times 10^{-9}$) fixing the parameterization of the Beta distribution to be the best values on the IMDB dataset. That is, for T5: $\alpha = 2, \beta = 1$; BERT-Base-Uncased: $\alpha = 2, \beta = 1$; GPT2: $\alpha = 2, \beta = 2$.

**ALUM on IMDB and ANLI.** We train all ALUM models for 50 epochs (the same as knowledge discontinuity regularized models). For hyperpararmeters specific to the ALUM algorithm we choose all of the same ones as its authors, [36], with the exception of $\alpha$ (analogous to the $\lambda$ in our algorithm, essentially the weight put on the virtual adversarial training loss term). The authors of the original paper choose $\alpha = 10$. We, however, found that this applied to finetuning does not converge at all. Thus, with a rough binary search in the parameter space we found $\alpha = 1 \times 10^{-3}$ to be the best with respect to both performance and robustness.

We keep the same hyperparameters on ANLI, however, we impose early stopping during the training process. That is, we choose the best model with respect to its performance on the **dev** set.

# J Limitations

The certification guarantees of our definition knowledge continuity is a probabilistic one. Specifically, this randomness is over the data distribution. However, this does not protect against out-of-distribution attacks that plague large language models such as [59, 76]. More work is needed to yield deterministic results that do not become vacuous in discrete settings. As mentioned in Section 2.4, our expressiveness bounds only apply under little restrictions to the metric decompositions of the estimator $f$. Though we see some empirical verification for this in Appendix I, it remains unclear whether or not we can tighten these bounds.

# K Broader Impacts

This contribution is concerned with robust deep learning models. As deep learning becomes ubiquitous as the mode for artificial intelligence, their applications in increasingly critical areas to the lay and corporations alike demand not only both high inferential accuracy and confidence. Robustness addresses this latter point, by making deep learning models more robust, we improve the trustworthiness of their decision-making and protect them against adversaries. More specifically, our contribution unifies separate robustness efforts from continuous and discrete domains.

# L Reproducibility

All of our experiments were conducted on four NVIDIA RTX A6000 GPUs as well as four NVIDIA Quadro RTX 6000 GPUs. The rest of our code base including implementations of the algorithms and figures described in the manuscript are attached as supplementary materials.