On the Representation Power of Set Pooling Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Point clouds and sets are ubiquitous, unusual and unstructured data-types which present unique problems to machine learning when used as inputs. Since sets are inherently unaffected by permutations, the input space for these problems naturally forms a non-Euclidean space whose topology depends on the task and which is oftentimes not even a manifold. Moreover, similar inputs can have wildly varying cardinalities and so the input space in its most general form is infinite-dimensional. Despite these mathematical difficulties, *PointNet* [8] and *Deep Sets* [16] form two foundational contributions for deep learning in this area. In this paper we study the expressive power of such networks and prove new cardinality-agnostic universality results for point clouds as well as extensions of these models beyond point clouds. These results completely characterize the approximable functions and so can be used to compare the representational strength of the underlying model classes. In particular, a normalized version of the DeepSets architecture cannot uniformly approximate the diameter function but can uniformly approximate the center-of-mass function whereas PointNet can uniformly approximate the former but not the latter. Additionally, even when limited to a fixed input cardinality, PointNet cannot uniformly approximate the average value of a continuous function over sets of more than two points. We additionally obtain explicit error lower-bounds for this error of approximation and a present a simple geometric method to produce arbitrarily many examples of this failure-mode.

1 Introduction

The architectures introduced in *PointNet* [8] and *Deep Sets* [16] are foundational contributions to the direct analysis of point clouds and sets via deep learning. These works provided two of the earliest such methods and continue to be of theoretical interest since both provide architectures which are inherently *permutation-invariant*. In addition to this, a single invariant model of either approach can handle sets of differing sizes and can theoretically scale to any cardinality with enough computational resources. These methods achieve this through careful application of pooling and weight sharing with their only difference being their choice of pooling function.

In either case the procedure to process a point cloud A in $\mathbb{R}^n$ is simply as follows: (1) apply a network $\boldsymbol{\varphi}$ to each element of A, (2) apply a permutation-invariant pooling operation to aggregate these point-features into a global feature for A (e.g. max-pooling for PointNet, sum-pooling for DeepSets), and lastly (3) pass this global feature to the second network $\boldsymbol{\rho}$ to obtain the final output. For general PointNet and DeepSets permutation-invariant models, this can be concisely expressed mathematically as

$$\boldsymbol{\psi}_{PointNet}(A) = \boldsymbol{\rho}\left(\max_{\boldsymbol{a}\in A} \boldsymbol{\varphi}(\boldsymbol{a})\right), \quad \boldsymbol{\psi}_{DeepSets}(A) = \boldsymbol{\rho}\left(\sum_{\boldsymbol{a}\in A} \boldsymbol{\varphi}(\boldsymbol{a})\right),$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

respectively. Depending on the output layer of $\boldsymbol{\rho}$ this can be used for either set-regression or set-classification. Note that arbitrarily large point clouds can be passed through and that the permutation-invariance of the max/sum-pooling operations ensure that rearrangement of the elements of A do not alter the output.

Importantly, each of these prior works provide their own universal approximation theorems (UATs) to support the empirical success of their methods. In this work we prove substantial extensions of these results motivated by the following points:

Unified Approach and Minimal Universality: The pre-existing universality results use very different different approaches. Additionally, these results do not explore the minimum architecture requirements needed for universality. In this paper we present a proof method that applies to more than one model and provides universal "shallow" examples.

Cardinality Limitations: The universality results in [8, 16] both assume that the cardinality of the inputs are fixed to some size n and thus do not make use of the cardinality-agnostic nature of these models. This gap is of worthy of investigation because (1) real-world point cloud datasets can have heterogeneous cloud sizes and (2) it may happen that the deployed model encounters point clouds with cardinalities that did not exist in the training data (e.g. if better sensors become available). It is not immediately clear whether one should expect these model classes to have enough power to universally approximate the functions of interest when allowing for such changes in set size.

Infinite Input-Width Limits: One approach for learning with mesh inputs is to sample the mesh and feed the resulting point cloud into a suitable neural network [8, 9, 3]. Although PointNet and DeepSets can readily accept samples of any size and via any sampling method, the computation graph at the input layers necessarily widens as the point clouds get larger. Understanding the expressiveness and consistency as sampling cardinality (and hence input layer size) grows to infinity may provide theoretical insight on this approach to learning from meshes. Models such as Neural ODEs [1] and Neural Tangent Kernels [4] have benefited from similar considerations (infinite depth and infinite hidden-layer width limits respectively).

DeepSets Extension Conjecture: It was conjectured in the supplementary materials of [16] (below the proof Theorem 9) that the DeepSets invariant model should be extendable to input sets of countably infinite or even unaccountably infinite size and retain universality in that setting. However, they note that there are fundamental topological obstructions to answering this question. We provide partial resolutions to this conjecture by showing that a change in the pooling function allows for inputs beyond finite sets (Sec 3.4).

Comparison of Representational Power: In this work we slightly modify the permutation-invariant DeepSets model by use of average-pooling instead of sum-pooling. The only change is re-scaling by the set size before applying $\boldsymbol{\rho}$ so we call this version **normalized-DeepSets**. We find that:

- PointNet (normalized-DeepSets) can uniformly approximate only the functions which are uniformly continuous with respect to the Hausdorff (Wasserstein) metric (Theorem 3.4).
- Only the constant functions can be simultaneously uniformly approximated by both PointNet and normalized-DeepSets when input sets are allowed to be arbitrarily large (Theorem 4.1).
- Even when cardinality is fixed to size k there is a substantial difference in approximation power. In particular, PointNet cannot uniformly approximate averages over sets such as the center-of-mass. We prove an explicit error lower bound for these tasks for PointNet and provide simple method for generating examples of this failure-mode (Theorem 4.2).

Related work. Besides the base invariant Pointnet model introduced in [8], the authors also include a variant for point cloud segmentation, a subnetwork to learn alignment, and later introduce a hierarchical variant of PointNet in [9]. The authors of [16] also introduce a distinct permutation-*equivariant* model where permutation of the inputs results in a corresponding permutation of the output. Though such equivariant networks are also commonly referred to as DeepSets networks, they are fundamentally different and are generally not permutation-*invariant*. We will always refer to the invariant models in the context of this paper and thus we do not consider the equivariant case here. Besides these models see [3] for a comprehensive survey of deep learning methods for point clouds.

There has also been much activity regarding the representational power of permutation-invariant models. Whether permutation-invariant functions can be *exactly* represented by sum-decomposition – i.e. as $\rho(\sum_{a\in A}\varphi(a))$ – has been addressed by [16, 13, 14] with positive and negative results

depending on whether the point clouds come from a countable or uncountable universe. In particular, [16] proves that sum-decomposition is possible when the universe is countable and [13] shows that there are continuous permutation-invariant functions on $\mathbb{R}$ that are not sum-decomposable when the latent space is too small. Additionally, [14] considers similar questions but in the context of graph neural networks and also assesses the relative ability of max-pooling, average-pooling, and sum-pooling to distinguish multisets. Regarding approximation of permutation-invariant functions [15] also proves a universal approximation result for fixed cardinality and also establishes a universality results for general G-invariant functions by appealing to Stone-Weierstrass and Hilbert's finiteness theorems from classical invariant theory.

2 Preliminaries

In this paper, we focus on understanding which functions on the space of point clouds can be uniformly approximated by architectures like that of *PointNet* and *Deep Sets*. Although tasks such as point cloud segmentation or point cloud generation are not directly addressed, some results may apply depending on architecture details.

2.1 PointNet, DeepSets, and Normalized-DeepSets

In addition to the max-pooling of PointNet and sum-pooling of DeepSets, we also consider average-pooling. For $A \subseteq \mathbb{R}^n$ a finite point cloud, the network architectures are respectively given by

$$\psi_{\max}(A) = \boldsymbol{\rho}\left(\max_{\boldsymbol{a}\in A} \boldsymbol{\varphi}(\boldsymbol{a})\right), \quad \psi_{\text{ave}}(A) = \boldsymbol{\rho}\left(\frac{1}{|A|}\sum_{\boldsymbol{a}\in A} \boldsymbol{\varphi}(\boldsymbol{a})\right), \quad \psi_{\text{sum}}(A) = \boldsymbol{\rho}\left(\sum_{\boldsymbol{a}\in A} \boldsymbol{\varphi}(\boldsymbol{a})\right).$$

Here $\boldsymbol{\varphi} : \mathbb{R}^n \to \mathbb{R}^m$ creates features for each point in A which gets symmetrically pooled and then passed to $\boldsymbol{\rho} : \mathbb{R}^m \to \mathbb{R}^\ell$ (where $\max$ is the component-wise maximum). We will usually take $\ell = 1$ and in practice we want both $\boldsymbol{\rho}$ and $\boldsymbol{\varphi}$ to be neural networks or some other parametrized model with learnable parameters. Once again, note that because we use a symmetric operation before $\boldsymbol{\rho}$, the output will not depend on the ordering of points in the point cloud, and because the pooling operations scale to arbitrary finite cardinalities the size of the point cloud is not an issue.

It may at first seem that the use of average-pooling instead of sum-pooling makes no difference. Indeed, this is true when cloud cardinality is fixed since then the $|A|^{-1}$ factor can be absorbed into ρ. However, it does make a difference when cardinality is free to change. To see this imagine a finer and finer uniform sampling of the unit circle – if we use average-pooling, the inside of ρ will converge to the average value of φ on the circle, but it may diverge if we use sum-pooling (for example if $\varphi > 0$ everywhere ρ is linear). As a result, this subtle change will be helpful in our later analysis.

It will help to introduce some simplifying notation. Let $\mathcal{F}(\Omega)$ denote the set of all nonempty finite subsets of a set Ω (i.e. point clouds in Ω), $\mathcal{F}^{\leq k}(\Omega)$ the set of nonempty subsets of size $\leq k$, and $\mathcal{F}^k(\Omega)$ the set of k-point subsets. Now consider a set Ω, a function $f : \Omega \to \mathbb{R}$, and define $\max_f, \text{ave}_f, \text{sum}_f : \mathcal{F}(\Omega) \to \mathbb{R}$ as the set-functions given by

$$\max_f(A) = \max_{\boldsymbol{a}\in A} f(\boldsymbol{a}), \quad \text{ave}_f(A) = \frac{1}{|A|}\sum_{\boldsymbol{a}\in A} f(\boldsymbol{a}), \quad \text{sum}_f(A) = \sum_{\boldsymbol{a}\in A} f(\boldsymbol{a}),$$

respectively. We make sense of this in the natural way if we use vector-valued $\boldsymbol{f}$ by operating component-wise. When convenient, we refer to these operations as max-neurons, averaging-neurons, and sum-neurons respectively.

With these conventions, $\rho \circ \max_{\boldsymbol{\varphi}}$, $\rho \circ \text{sum}_{\boldsymbol{\varphi}}$, and $\rho \circ \text{ave}_{\boldsymbol{\varphi}}$ will be the general form of what we mean by the PointNet, DeepSets, and normalized-DeepSets architectures respectively. The case of $\rho \circ \text{sum}_{\boldsymbol{\varphi}}$ will prove difficult to treat in the same manner as the others and so will get a separate treatment.

Some natural questions are 1) what are the natural nontrivial topologies for $\mathcal{F}(\Omega)$ which make these architectures continuous, 2) how expressive are these approaches, and 3) how deep is deep enough for function approximation?

2.2 Function Spaces and Uniform Approximation

From now on, we only consider $\mathbb{R}$-valued functions unless otherwise stated. Let $\mathcal{B}(A)$ be the set of bounded functions on a set A, let $\mathcal{C}(X)$ be the set of continuous functions on a topological space X, and let $\mathcal{U}(M)$ be the uniformly continuous functions on a metric space (M, d). Additionally we can define the bounded versions $\mathcal{C}_b(X) = \mathcal{C}(X) \cap \mathcal{B}(X)$ and $\mathcal{U}_b(M) = \mathcal{U}(M) \cap \mathcal{B}(M)$. We can equip the space of bounded functions $\mathcal{B}(A)$ with the supremum norm $|||f|||_A = \sup_{a\in A} |f(a)|$ – but we reserve $\|\cdot\|$ for the Euclidean norm. This makes all the bounded families $\mathcal{U}_b(M), \mathcal{C}_b(X), \mathcal{B}(A)$ into Banach spaces. We can equip the other spaces with $|||\cdot|||_A$ as well but we may obtain infinities and hence it will not be a true norm in general but this will at least induce a topology on these spaces. However, if X is compact or M has compact metric completion then we have $\mathcal{C}(X) = \mathcal{C}_b(X)$ and $\mathcal{U}(M) = \mathcal{U}_b(M)$. For more background see [10]. Finally, the uniform closure of a family of functions is just the topological closure with respect to the supremum norm or equivalently, it is the set of all functions which can be uniformly approximated via members of the family.

If given an injective map $i : A \to X$, then we say that $\varphi : A \to \mathbb{R}$ (uniquely) continuously extends to X if there is a (unique) $\widetilde{\varphi} \in \mathcal{C}(X)$ such that $\widetilde{\varphi} \circ i = \varphi$. We say a family of functions $\mathcal{N}$ on A (uniquely) continuously extends to X if every $\varphi \in \mathcal{N}$ (uniquely) continuously extends to X.

We will make use of the following lemma.

Lemma 2.1. *Let $\mathcal{N} \subseteq \mathcal{B}(D)$ where D is a dense subset of a compact Hausdorff space X. Suppose $\mathcal{N}$ has a continuous extension to X denoted by $\mathcal{N}' \subseteq \mathcal{C}(X)$ which is dense. Then the uniform closure of $\mathcal{N}$ in $\mathcal{B}(D)$ is $r(\mathcal{C}(X))$ where $r : \mathcal{C}(X) \to \mathcal{C}_b(D)$ is the domain restriction map. If X can be metrized by a metric d, then $r(\mathcal{C}(X)) = \mathcal{U}(D)$.*

Letting $D = \mathcal{F}(\Omega)$, this lemma suggest the following plan of attack: find a compact metric space (X, d) in which we can realize $\mathcal{F}(\Omega)$ as a dense subset and hope that our class of neural networks $\mathcal{N}$ continuously extends to a dense subset of $\mathcal{C}(X)$. If we can do that, then we know the uniform closure of our class of neural networks are precisely the uniformly continuous functions on $\mathcal{F}(\Omega)$ with respect to the metric inherited from X. This motivates the next subsection.

2.3 Topologies and Metrics for Point Clouds

From now on we will assume (Ω, d) is a compact metric space and when $\Omega \subseteq \mathbb{R}^n$ it will be compact and equipped with the Euclidean metric. Let $\mathcal{K}(\Omega)$ denote the set of all compact subsets of Ω and $\mathcal{P}(\Omega)$ denote the set of all Borel probability measures on Ω. The Hausdorff metric d_H [7, 6] is a natural metric for $\mathcal{K}(\Omega)$ and 1-Wasserstein metric d_W [12] (also called the Earth-mover distance) is a natural metric for $\mathcal{P}(\Omega)$. With these metrics, $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ become compact metric spaces of their own because Ω is compact. From now on we will assume these two spaces are always equipped with the aforementioned metrics.[1]

We also briefly mention $M(\Omega)$ the Banach space of finite signed regular Borel measures on Ω. By the Riesz-Markov theorem it is the topological dual space of $\mathcal{C}(\Omega)$. Of interest to us is that $\mathcal{P}(\Omega) \subseteq M(\Omega)$ and that the weak-* topology on $\mathcal{P}(\Omega)$ coincides with the topology induced by d_W. This means that $d_W(\mu_n, \mu) \to 0$ iff $\int f \, \mathrm{d}\mu_n \to \int f \, \mathrm{d}\mu$ for all $f \in \mathcal{C}(\Omega)$.

Next, note that $\mathcal{F}(\Omega) \subseteq \mathcal{K}(\Omega)$ and let $i_{\mathcal{K}}$ denote the natural inclusion map. We can also define an injective map $i_{\mathcal{P}} : \mathcal{F}(\Omega) \to \mathcal{P}(\Omega)$ by mapping $A \in \mathcal{F}(\Omega)$ to its associated empirical measure $i_{\mathcal{P}}(A) = \mu_A = \frac{1}{|A|} \sum_{a\in A} \delta_a \in \mathcal{P}(\Omega)$ where δ_a is the Dirac delta measure supported at a. The injective maps $i_{\mathcal{K}}$ and $i_{\mathcal{P}}$ allow us to induce the d_H and d_W metrics on $\mathcal{F}(\Omega)$. We will denote the metrized versions by $\mathcal{F}_H(\Omega)$ and $\mathcal{F}_W(\Omega)$ respectively and use the same convention for $\mathcal{F}_H^k(\Omega)$ and $\mathcal{F}_W^k(\Omega)$. Another important fact to know is that $i_{\mathcal{K}}$ and $i_{\mathcal{P}}$ embed $\mathcal{F}(\Omega)$ as a dense subset of $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$. The former follows from compactness of the members of $\mathcal{K}(\Omega)$ and to see why the latter is true see [2, 12].

For $f \in \mathcal{C}(\Omega)$, define $\mathrm{Max}_f : \mathcal{K}(\Omega) \to \mathbb{R}$ and $\mathrm{Ave}_f : \mathcal{P}(\Omega) \to \mathbb{R}$ as the functions given by $\mathrm{Max}_f(K) = \max_{x\in K} f(x)$ and $\mathrm{Ave}_f(\mu) = \int_\Omega f \, \mathrm{d}\mu = \mathbb{E}_{x\sim\mu}[f(x)]$.

[1] The topology induced on $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ by d_H and d_W depends only on the topology of (Ω, d) and not on the choice of metric d. Indeed, we could have proceeded without metrics at all and used these topologies directly from the topology of Ω and they are called the Vietoris tpology [6] and weak-* topologies [10] respectively. However, metric spaces are sufficient generality so we take advantage of their existence.

Lemma 2.2. *Let (Ω, d) be compact, $f \in \mathcal{C}(\Omega)$. Then $\mathrm{Max}_f \in \mathcal{C}(\mathcal{K}(\Omega))$ and $\mathrm{Ave}_f \in \mathcal{C}(\mathcal{P}(\Omega))$ and $\mathrm{Max}_f \circ i_{\mathcal{K}} = \max_f$ and $\mathrm{Ave}_f \circ i_{\mathcal{P}} = \mathrm{ave}_f$. As a consequence, PointNet and normalized-DeepSets are uniformly continuous on $\mathcal{F}_H(\Omega)$ and $\mathcal{F}_W(\Omega)$ respectively.*

This lemma tells us that the max-neurons and averaging-neurons continuously extend to $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ and hence so do PointNet and normalized-DeepSets architectures (since we merely compose with the continuous ρ after). Thus, we will be able analyze such architectures as continuous functions on compact metric spaces, which is mathematically a much better problem than studying them as set-theoretic functions on an un-metrized $\mathcal{F}(\Omega)$.

2.4 Generalized Neural Network Notation

For a collection $\mathcal{A}$ of functions from X to Y and a collection $\mathcal{B}$ of functions from Y to Z, we denote the set of all compositions by $\mathcal{B} \circ \mathcal{A} = \{f \circ g \mid f \in \mathcal{B}, g \in \mathcal{A}\}$. In the case of a single function $\sigma : Y \to Z$ we let $\sigma \circ \mathcal{A} = \{\sigma \circ f \mid f \in \mathcal{A}\}$.

Next, let Aff denote the set of all affine functionals on $\mathbb{R}^N$, i.e. any function of the form $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$. We write $\mathcal{N}^\sigma := \mathrm{span}\{\sigma \circ \mathrm{Aff}\}$ to denote the set of single hidden-layer neural networks with linear scalar output-layer and activation function σ. The H-layered networks are denoted by $\mathcal{N}^{\boldsymbol{\sigma}}$ where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_H)$ is a list of H-many activation functions where $\mathcal{N}^{(\boldsymbol{\sigma},\tau)} := \mathcal{N}^{\boldsymbol{\sigma},\tau} := \mathrm{span}(\tau \circ \mathcal{N}^{\boldsymbol{\sigma}})$.

Next we define various classes of PointNet networks are themselves comprised of sub-networks. Let $\mathcal{N}^\sigma_{\max} := \mathcal{N}^{\sigma;\varnothing}_{\max} := \mathrm{span}\{\max_f \mid f \in \mathcal{N}^\sigma\}$ then define $\mathcal{N}^{\sigma;\tau}_{\max} := \mathrm{span}\{\tau \circ \mathcal{N}^\sigma_{\max}\}$. Like before, we can inductively define deeper networks, but we can also use deeper "weight" networks as well to create $\mathcal{N}^{\boldsymbol{\sigma};\boldsymbol{\tau}}_{\max}$ – thus we have two distinct notions of depth.

Next, we do the same for normalized-DeepSets. Let $\mathcal{N}^\sigma_{\mathrm{ave}} := \mathcal{N}^{\sigma;\varnothing}_{\mathrm{ave}} := \mathrm{span}\{\mathrm{ave}_f \mid f \in \mathcal{N}^\sigma\}$. Note that $\mathrm{ave}_f + \mathrm{ave}_g = \mathrm{ave}_{f+g}$ and $\alpha \mathrm{ave}_f = \mathrm{ave}_{\alpha f}$. Thus since $\mathcal{N}^\sigma$ is a linear space, taking the span has no effect and $\mathcal{N}^\sigma_{\mathrm{ave}} = \{\mathrm{ave}_f \mid f \in \mathcal{N}^\sigma\}$. Going one layer deeper yields $\mathcal{N}^{\sigma;\tau}_{\mathrm{ave}} := \mathrm{span}\{\tau \circ \mathcal{N}^\sigma_{\mathrm{ave}}\}$ which gets us new functions. Like with PointNet, we can inductively develop deeper families in two ways.

By Lemma 2.2 we can extend all the operations of our neural networks to $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ in a natural way. This let's us talk about about PointNet networks on $\mathcal{K}(\Omega)$ and normalized-DeepSets networks on $\mathcal{P}(\Omega)$ which we'll define analogously by replacing $\max_f$ with Max_f and ave_f with Ave_f. Thus

$$\mathcal{N}^{\sigma;\tau}_{\mathrm{Max}} = \mathrm{span}\{\tau \circ \mathcal{N}^\sigma_{\mathrm{Max}}\}, \quad \mathcal{N}^{\sigma;\tau}_{\mathrm{Ave}} = \mathrm{span}\{\tau \circ \mathcal{N}^\sigma_{\mathrm{Ave}}\}$$

where $\mathcal{N}^\sigma_{\mathrm{Max}} = \mathrm{span}\{\mathrm{Max}_f \mid f \in \mathcal{N}^\sigma\}$ and $\mathcal{N}^\sigma_{\mathrm{Ave}} = \mathrm{span}\{\mathrm{Ave}_f \mid f \in \mathcal{N}^\sigma\}$. As before, the linear structure of $\mathcal{N}^\sigma$ means $\mathcal{N}^\sigma_{\mathrm{Ave}}$ could have been defined without the span.

3 Universality Results

3.1 Topological UAT

In it is proven in [5] that $\mathcal{N}^\sigma$ with $\sigma \in \mathcal{C}(\mathbb{R})$ has universal approximation property iff σ is not a polynomial. For this reason, we will say a $\sigma \in \mathcal{C}(\mathbb{R})$ is 'universal' if it is non-polynomial and denote the set of all such such functions by $\mathfrak{U}(\mathbb{R})$. Using this theorem and Stone-Weierstrass we prove a UAT for certain kinds of generalized neural networks on an abstract compact Hausdorff space. Though we independently arrived at this theorem, we later discovered this result was essentially proven in [11] (Theorem 5.1) for a different context. However, our proof is slightly different and so we provide a detailed proof in the supplementary materials for completeness and the benefit of the reader.

Recall that a family of functions S on Ω separates points if for any $x \neq y$ there is an $f \in S$ so that $f(x) \neq f(y)$.

Theorem 3.1 (Topological-UAT)**.** *Let X be a compact Hausdorff space and $\sigma \in \mathfrak{U}(\mathbb{R})$. If $S \subseteq \mathcal{C}(X)$ separates points and contains a nonzero constant, then $\mathrm{span}(\sigma \circ \mathrm{span}\, S)$ is dense in $\mathcal{C}(X)$. Additionally, if S also happens to be a linear subspace, then $\mathrm{span}(\sigma \circ S)$ is dense in $\mathcal{C}(X)$.*

3.2 Point Cloud UAT

With the following lemma, we will have met all the conditions required to use the topological-UAT on $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$.

Lemma 3.2 (Separation Lemma)**.** *Let $\Omega \subseteq \mathbb{R}^N$ be compact and $\sigma \in \mathfrak{U}(\mathbb{R})$. Then the set of functions $S_{\mathrm{Max}} = \{\mathrm{Max}_f \mid f \in \mathcal{N}^\sigma\}$ and $S_{\mathrm{Ave}} = \{\mathrm{Ave}_f \mid f \in \mathcal{N}^\sigma\}$ separate points and contain constants.*

The following theorems show that one hidden layer in the inner network and one hidden layer in the outer network suffice to prove the universal approximation theorems for PointNet and normalized-DeepSets.

Theorem 3.3. *Let $\Omega \subseteq \mathbb{R}^N$ be compact and $\sigma, \tau \in \mathfrak{U}(\mathbb{R})$. Then $\mathcal{N}_{\mathrm{Max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{Ave}}^{\sigma;\tau}$ are dense in $\mathcal{C}(A)$ and $\mathcal{C}(B)$ respectively, whenever $A \subseteq \mathcal{K}(\Omega)$ and $B \subseteq \mathcal{P}(\Omega)$ are closed subsets.*

Proof. Recall $\mathcal{N}_{\mathrm{Max}}^{\sigma;\tau} = \mathrm{span}\,\{\tau \circ \mathrm{span}\, S_{\mathrm{Max}}\}$ and $\mathcal{N}_{\mathrm{Ave}}^{\sigma;\tau} = \mathrm{span}\,\{\tau \circ S_{\mathrm{Ave}}\}$. Since $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ are compact metric spaces, and A and B are closed, they must be also be compact Hausdorff. By Lemma 3.2 we know S_{Max} and S_{Ave} separate points and contain nonzero constants and so the topological-UAT (Theorem 3.1) yields the desired result. □

For $f : \mathcal{F}(\Omega) \to \mathbb{R}$ the following result tells us that under mild hypothesis that (1) PointNets can uniformly approximate f iff f is d_H-uniformly continuous and (2) normalized-DeepSets can uniformly approximate f iff f is d_W-uniformly continuous.

Theorem 3.4 (Point-Cloud-UAT)**.** *Let $\Omega \subseteq \mathbb{R}^N$ be compact. If $\sigma, \tau \in \mathfrak{U}(\mathbb{R})$, then the uniform closure of $\mathcal{N}_{\mathrm{max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{ave}}^{\sigma;\tau}$ within $\mathcal{B}(\mathcal{F}(\Omega))$ is $\mathcal{U}(\mathcal{F}_H(\Omega))$ and $\mathcal{U}(\mathcal{F}_W(\Omega))$ respectively.*

Proof. $\mathcal{F}_H(\Omega)$ and $\mathcal{F}_W(\Omega)$ are isometrically isomorphic to $i_{\mathcal{K}}(\mathcal{F}(\Omega))$ and $i_{\mathcal{P}}(\mathcal{F}(\Omega))$ which are in turn dense in $(\mathcal{K}(\Omega), d_H)$ and $(\mathcal{P}(\Omega), d_W)$. By Lemma 2.2 we have that $\mathcal{N}_{\mathrm{max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{ave}}^{\sigma;\tau}$ continuously extend to $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ as $\mathcal{N}_{\mathrm{Max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{Ave}}^{\sigma;\tau}$. By Theorem 3.3 we know $\mathcal{N}_{\mathrm{Max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{Ave}}^{\sigma;\tau}$ are dense in $\mathcal{C}(\mathcal{K}(\Omega))$ and $\mathcal{C}(\mathcal{P}(\Omega))$. Finally, by Lemma 2.1 we have the desired result. □

It's worth noting that we could have directly used another UAT proven in [11] for neural networks on locally convex spaces [11](Corollary 5.1.2) for the universality of normalized-DeepSets. However this result would not directly work for PointNets so we chose the above route for consistency of technique and to be self-contained.

We now prove as a corollary a refinement of the universal approximation theorems of [8] and [16], both of which applied to the the case of k-point point clouds (for fixed k). In this version of the theorem we are able to restrict the depth of the neural network to just three hidden layers (when counting the pooling). Additionally, as with the above, this result simultaneously establishes which functions *cannot* be uniformly approximated by these architectures. The proof is essentially the same as Theorem 3.4.

Corollary 3.5. *Let $\Omega \subseteq \mathbb{R}^N$ be compact. If $\sigma, \tau \in \mathfrak{U}(\mathbb{R})$, then the uniform closure of $\mathcal{N}_{\mathrm{max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{ave}}^{\sigma;\tau}$ within $\mathcal{B}(\mathcal{F}^k(\Omega))$ are $\mathcal{U}(\mathcal{F}_H^k(\Omega))$ and $\mathcal{U}(\mathcal{F}_W^k(\Omega))$ respectively.*

Proof. $\mathcal{F}_H^k(\Omega)$ and $\mathcal{F}_W^k(\Omega)$ are isometrically isomorphic to $i_{\mathcal{K}}(\mathcal{F}^k(\Omega))$ and $i_{\mathcal{P}}(\mathcal{F}^k(\Omega))$ which are in turn dense in their respective closures which we denote $\mathcal{G}_H(\Omega) \subseteq \mathcal{K}(\Omega)$ and $\mathcal{G}_W(\Omega) \subseteq \mathcal{P}(\Omega)$. Thus by Lemma 2.2 and Theorem 3.3 we have that $\mathcal{N}_{\mathrm{Max}}^{\sigma;\tau}$ and $\mathcal{N}_{\mathrm{Ave}}^{\sigma;\tau}$ are dense in $\mathcal{C}(\mathcal{G}_H(\Omega))$ and $\mathcal{C}(\mathcal{G}_W(\Omega))$. Finally, by Lemma 2.1 we have the desired result. □

3.3 Stability and Lipschitz Property

By refining the continuity arguments for Max_f and Ave_f, we can deduce the Lipchitz properties of these maps from the Lipschitz properties of f.

Lemma 3.6. *Suppose (Ω, d) is compact and $f : \Omega \to \mathbb{R}$ is Lipschitz continuous with Lipschitz constant L. Then $\mathrm{Max}_f \in \mathcal{K}(\Omega)$ and $\mathrm{Ave}_f \in \mathcal{P}(\Omega)$ have Lipschitz constant $2L$ and L.*

As a consequence of this, we can show that when the activation functions are Lipschitz, the whole neural network architecture will be Lipschitz. One implication of this is that if one were to sample a

mesh sufficiently well, and knew the Lipschitz constant of the whole PointNet/normalized-DeepSets network, then it would be possible to bound how big of a difference in the outputs two different samplings of the same object would produce. This bound would not depend on the cardinality, but on the quality of the sample with respect to the metrics d_H and/or d_W.

Theorem 3.7. *Suppose $\Omega \subseteq \mathbb{R}^N$ is compact. Then every PointNet and normalized-DeepSets network with Lipschitz activation functions is Lipschitz on $\mathcal{F}_H(\Omega)$ and $\mathcal{F}_W(\Omega)$ respectively.*

Proof. Recall that a PointNet network is of the form $\psi = \rho \circ \max_f$ for neural networks $f : \Omega \to \mathbb{R}^k$ and $\rho : \mathbb{R}^k \to \mathbb{R}^N$. Since both f and ρ are an alternating composition of affine transforms and nonlinear transforms, and all of these are Lipschitz, it follows that both f and ρ are Lipschitz. By Lemma 3.6, f being Lipschitz implies $\max_f$ is Lipschitz. Thus ψ is Lipschitz since it is a composition of Lipschitz functions. The analogous argument applies for the case of normalized-DeepSets. □

3.4 Theoretical Issues with Standard DeepSets

To use the universal approximation results developed here, it was key for the there to exists an underlying compact Hausdorff space on which to study the networks. In the case of PointNet and normalized-DeepSets, the compact Hausdorff space was not obvious, but we identified the appropriate extension spaces $\mathcal{K}(\Omega)$ and $\mathcal{P}(\Omega)$ to which we could unambiguously and continuously extend the networks. This gave a partial answer to the DeepSets extension conjecture of [16] in the sense that if we replace sum-pooling with max-pooling then these networks can be made to theoretically accept sets of uncountably infinite cardinality (so long as they are compact i.e. members of $\mathcal{K}(\Omega)$). Similarly, for sufficiently nice sets like smooth manifolds $M \subseteq \mathbb{R}^n$ one can feed the uniform probability measure μ_M of M to the extension of normalized-DeepSets to $\mathcal{P}(\Omega)$.

However, we were not able to do this for standard DeepSets. This is because, generally speaking, there is no compact Hausdorff space to which we can continuously extend sum_f for all reasonable neural networks f. This is because a continuous function on a compact space is necessarily a bounded function but $\mathrm{sum}_f : \mathcal{F}(\Omega) \to \mathbb{R}$ is generally unbounded. To illustrate this, let $\Omega = [0, 1] \subseteq \mathbb{R}$ and let f be a continuous function, e.g. a neural network, which is not identically zero. Then there must be a closed interval K on which $|f| > \epsilon$ for some $\epsilon > 0$. From this we can see that $\mathrm{sum}_f(A)$ for $A \subseteq K$ grows arbitrarily large in absolute value as the size of A increases. Thus, in this case, every sum_f except sum_0 is unbounded on $\mathcal{F}(\Omega)$ meaning there is no compact extension space.

One potential way to address this issue would be to only consider bounded non-linearities such as the sigmoid or arctan as a way to control the wild behavior as cloud size gets large. Even so, it is not immediately clear what the natural underlying compact Hausdorff space for such a model would be and it may very well depend on the activation functions. Alternatively, alternative tools may be needed that don't depend on compactness.

4 Limitations of PointNets and Normalized-DeepSets

Note that unlike the classical universal approximation theorem we should not expect to be able uniformly approximate $\mathcal{C}(\mathcal{F}_H(\Omega))$ or $\mathcal{C}(\mathcal{F}_W(\Omega))$ since their elements might not even be bounded functions. For example, $\alpha_K(A) = d_H(A, K)^{-1}$ is unbounded but continuous on $\mathcal{F}_H(\Omega)$ whenever $K \in \mathcal{K}(\Omega)$ but $K \notin \mathcal{F}(\Omega)$. Subtler still, we do not even obtain all elements of $\mathcal{C}_b(\mathcal{F}_H(\Omega))$ and $\mathcal{C}_b(\mathcal{F}_W(\Omega))$. As an example of this, observe that $\beta_K = \sin \circ \alpha_K$ is bounded and is also continuous on $\mathcal{F}_H(\Omega)$ because α_K is. In both cases α_K and β_K are not uniformly d_H-continuous and hence cannot be approximated. Analogous constructions can be made for d_W.

We'll now compare the representation power of these two architectures. Let $\Omega \subseteq \mathbb{R}^N$ be compact. We define the point cloud diameter function $\mathrm{Diam} : \mathcal{F}(\Omega) \to \mathbb{R}$ and point cloud center-of-mass function $\mathrm{Cent} : \mathcal{F}(\Omega) \to \mathbb{R}^N$ by $\mathrm{Diam}(A) = \max_{\boldsymbol{x},\boldsymbol{y} \in A} d(\boldsymbol{x}, \boldsymbol{y})$ and $\mathrm{Cent}(A) = \frac{1}{|A|} \sum_{\boldsymbol{x} \in A} \boldsymbol{x}$.

Theorem 4.1. *Let (Ω, d) be an infinite compact metric space with no isolated points. Then a function $f : \mathcal{F}(\Omega) \to \mathbb{R}$ is continuous with respect to both d_H and d_W iff it is constant. As a corollary,* Diam *is uniformly approximable by PointNet networks but not by normalized-DeepSets networks and* Cent *is uniformly approximable by normalized-DeepSets networks but not PointNet networks.*

While it is interesting to know that these neural networks describe fundamentally different kinds of functions when we allow for unbounded cloud cardinality, in practice there is always a bound due to computational resource limitations. Nevertheless, the above result sheds some light on the radically different nature of the PointNet and normalized-DeepSets models in the limit of large cloud size. The next result shows that even when we bound the cloud cardinality, PointNet still cannot uniformly approximate many simple functions such as the center-of-mass function.

Theorem 4.2. *Let (Ω, d) be an infinite compact metric space with no isolated points and $f : \Omega \to \mathbb{R}^N$ a continuous function. Then for every d_H-continuous $F : \mathcal{F}^{\leq k}(\Omega) \to \mathbb{R}^d$ there exists $A \in \mathcal{F}^k(\Omega)$ such that*

$$\|F(A) - \mathrm{ave}_f(A)\| \geq \frac{k-2}{2k} \operatorname{Diam}(f(\Omega)).$$

Since PointNet architectures are d_H-continuous they must suffer from the above uniform-norm error lower-bound. This means that when cloud cardinality is at least three, they cannot uniformly approximate functions of the form ave_f for continuous f on e.g. the unit ball. In particular, they cannot uniformly approximate the center-of-mass function because this corresponds to the case where $\Omega \subseteq \mathbb{R}^N$ and f is the identity map.

It is possible to see the intuition for why PointNet cannot uniformly approximate Cent by considering the following. Continuously moving $k-2$ points in a k-element cloud entirely to either of the remaining two points ultimately produces the same 2-element set in a d_H-continuous way. By d_H-continuity, this means PointNet must output similar predictions for the center-of-mass as we approach this 2-element set along each of these two paths. However, this leads to conflicting estimates for the center of mass along the way.

To obtain the explicit error lower bound in Theorem 4.2, we need a slightly more detailed result. The proof in the appendix for following result formalizes the above intuition.

Lemma 4.3. *Let (Ω, d) have infinite cardinality and no isolated points, $F : \mathcal{F}^{\leq k}(\Omega) \to \mathbb{R}^N$ an arbitrary d_H-continuous map, and $f : \Omega \to \mathbb{R}^N$ a continuous function. If $k \geq 3$, then for every distinct $p, q \in \Omega$ and $0 < \tau < 1$ there exists a k-point set $A \subseteq \Omega$ with $p, q \in A$ and*

$$\|F(A) - \mathrm{ave}_f(A)\| > (1-\tau)\left(\frac{k-2}{2k}\right)\|f(p) - f(q)\|$$

In particular, PointNet architectures satisfy this error bound since they are d_H-continuous.

proof of Theorem 4.2. By compactness of Ω, we can choose $p, q \in \Omega$ so that the diameter is achieved i.e. $\|f(p) - f(q)\| = \operatorname{Diam}(f(\Omega))$. Then we take $\tau \to 0$ in Lemma 4.3. □

5 Experiments

The proof of Lemma 4.3 not only establishes the error bound but also suggest an algorithmic approach to finding point clouds that exhibit the failure of uniform approximation. This let's us produce difficult examples for the centor-of-mass problem for PointNet (even if the weights change). When Ω is additionally convex – e.g. the unit disk D in $\mathbb{R}^2$ – it becomes fairly easy to construct many examples of A_p^δ and A_q^δ explicitly for a given PointNet model, allowing us to empirically verify the uniform-norm error lower bound. In the following experiment, we train a simple PointNet architecture to learn the center-of-mass for 10-element point clouds in D. We train on a synthetic dataset of 1 million point clouds (each element uniformly sampled from D) labeled with their center-of-mass. The PointNet architecture has 500K trainable parameters. The network has the form $F(A) = \boldsymbol{\rho}(\max_{\boldsymbol{a} \in A} \boldsymbol{\varphi}(a))$ where $\boldsymbol{\varphi}$ has 2-D input layer, 500-D hidden layer, and 500-D linear output layer, and $\boldsymbol{\rho}$ has 500-D input layer, 500-D hidden layer and 2-D linear output layer (in accordance with the Point-Cloud-UAT). The hidden layer activation functions of $\boldsymbol{\varphi}$ and $\boldsymbol{\rho}$ are ReLU. Since it is not possible to train with respect to the uniform-norm, we opt to train with the traditional L^2 loss.

To form our problematic examples, we pick a nonzero $\tau = 0.01$ and two distinct points $p, q \in D$ at random. We set $\delta = \tau \|p - q\| / 4$. We then sample D another 8 times and then linearly pull those 8 points towards p and towards q to within δ. This $\{p, q\}$ adjoined with the 8 points pulled towards p and q form A_p^δ and A_q^δ (resp). So that the criteria in the proof of Lemma 4.3 is satisfied, we continue pulling the 8 points closer until $F(A_p^\delta)$ and $F(A_q^\delta)$ are within δ of $F(\{p, q\})$. We are theoretically

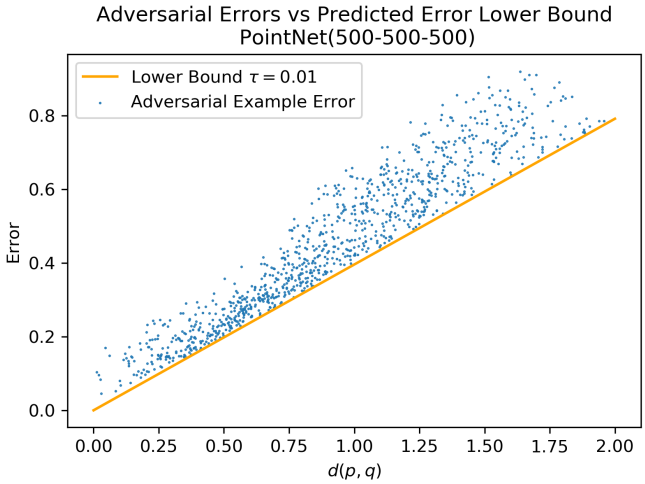

Figure 1: In our experimental tests of the error lower bound we see that we are always able to find problematic examples (blue) for PointNet on the task of computing the center-of-mass. The method always produces errors at least as large as the theoretical guarantee (orange).

ensured one of these two will have error larger than our bound. In Figure 1 we plot the the produced error vs the distance between p and q that were used to make A_p^δ and A_q^δ. As predicted, all the errors for the problematic examples lie above the line representing the uniform-norm error lower bound.

This experiment was chosen because the task is very simple (computing the mean), not very high-dimensional (ten 2D points per cloud), and yet PointNet fails to eliminate its errors through training on unlimited synthetic data. Moreover, the simple to produce problematic examples show that PointNet never violates the error lower bound.

6 Conclusions and Future Work

The failure of the perceptron model to learn the XOR function was a blow that motivated the search for new models. As multi-layer feed-forward neural networks began to show promise, the spectre of limited representation power loomed over the field until the universal approximation theorem resolved the question. In this work we advance state-of-the-art for the same question in the case of PointNet and a modification of DeepSets. Through this work we see that the choice of pooling function can have a dramatic effect on the properties of these networks. Moreover, since these networks form the foundation for much recent work in geometric deep learning, we hope that this program could prove useful for other models as well. However, in the case of standard DeepSets with sum-pooling, there are still some question left unanswered due to the unboundedness of the inner computation.

Directions for future research could include attempting to apply this approach for other pooling functions or blends of pooling functions such as average and max-pooling. Perhaps these techniques can be used to the representational power of heierachical point cloud models such as PointNet++ [9]. Another direction which deserves attention is whether the inability of PointNet to approximate certain functions is useful as a kind of inductive bias and if so can other useful inductive biases be chosen with by simply choose the right pooling function?

Finally, we can see that both PointNet and normalized-DeepSets naturally extend to very rich infinite-dimensional spaces. How can this be leveraged to attack problems such as learning with with mesh data? Can there be efficient and practical realizations of these theoretical extension networks and how would they fare?

References

[1] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

[2] Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

[3] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[5] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[6] Ernest Michael. Topologies on spaces of subsets. *Transactions of the American Mathematical Society*, 71(1):152–182, 1951.

[7] James Raymond. Munkres. *Topology*. Prentice Hall, 2000.

[8] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[9] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5099–5108. Curran Associates, Inc., 2017.

[10] Walter Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

[11] M.b. Stinchcombe. Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Networks*, 12(3):467–477, 1999.

[12] Cedric Villani. *Optimal transport: old and new*. Springer, 2009.

[13] Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Ingmar Posner, and Michael Osborne. On the limitations of representing functions on sets. *arXiv preprint arXiv:1901.09006*, 2019.

[14] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[15] Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *arXiv preprint arXiv:1804.10306*, 2018.

[16] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3391–3401. Curran Associates, Inc., 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] The work is of a theoretical nature and so it is difficult to say exactly what impact it could have on society if any.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes] Full proofs can be found in main paper or in supplemental materials.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Will be included with supplemental.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The experiment was with synthetic data so was mostly a sanity check for the theorems so there was not much to describe.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] The sole experiment is not of a nature that lends itself to error bars. It merely checks whether every single case has an error above the theoretical prediction. A single violation of this would suggest there is an error in the code or proof.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] It is a deliberately lightweight experiment so this was omitted.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]