We would like to thank the reviewers for taking the time to carefully read, evaluate, and give feedback on our submission.

Assumptions 3.1 and 3.2- Reviewer 2: *"[T]his paper... assumes that the feature map is regular (Assumption 3.1)... I am worried that the [assumptions] are too restrictive"*. (Assumption 3.1) is necessary and sufficient for $\{f \circ \phi : f \in \mathcal{F}\}$ to be dense in $C(X, \mathbb{R}^n)$ if $\mathcal{F}$ is. Indeed, if ϕ were not continuous then $\{f \circ \phi : f \in \mathcal{F}\}$ may fail to belong to $C(X, \mathbb{R}^n)$. If ϕ were not injective then there would exist $x_1, x_2 \in X$ such that $\phi(x_1) = \phi(x_2)$ and therefore any $g \in \{f \circ \phi : f \in \mathcal{F}\}$ satisfies $g(x_1) = g(x_2)$, and likewise for limits of any sequence in $\{f \circ \phi : f \in \mathcal{F}\}$. Hence $\{f \circ \phi : f \in \mathcal{F}\}$ would be a proper closed subset of $C(X, \mathbb{R}^n)$ and therefore it could not be dense. (Assumption 3.2) is almost sharp and a characterization can be obtained using the $\mathcal{Z}$-sets as defined in [1]. However, it is unlikely that a non-pathological example can be generated which fails our assumptions but meets a refinement using $\mathcal{Z}$-sets.

Examples - Reviewer 3: *"Guidelines[/Examples] for building such input and readout maps"*.

Aside from the examples arising from classification and Riemannian Exponential/Logarithm map examples discussed in Sections 3.1 and 3.2, two examples of feature maps between Euclidean spaces, satisfying the Assumptions 3.1 and 3.2 are the following. Let $g : \mathbb{R}^m \to \mathbb{R}^d$ be a continuous function, then $\phi(x) \triangleq (x, g(x))$ satisfies Assumption 3.1. Alternatively, if $A_{i,j}$ are any full-rank square matrices if $j \neq 2$ and $A_{2,1} \circ A_{1,1}$ is well-defined, then the set of DNNs of the form $\rho \circ f \circ \phi$ where

$$\begin{aligned} \rho(x) &= Leaky\text{-}ReLU \bullet (A_{1,K}x + b_{1,K}) \circ \cdots \circ Leaky\text{-}ReLU \bullet (A_{1,1}x + b_{1,1}) \\ f(x) &= [A_{2,2} \circ \bullet\, ReLU \bullet (A_{2,1}x + b_{2,1}) + b_{2,2}] \\ \phi(x) &= Leaky\text{-}ReLU \bullet (A_{3,k}x + b_{3,k}) \circ \cdots \circ Leaky\text{-}ReLU \bullet (A_{3,1}x + b_{3,1}). \end{aligned} \tag{1}$$

are universal since the input and output maps both satisfy Assumptions 3.1 and 3.2. Note that the matrices $A_{i,j}$ may be highly sparse with at-least m (resp. n) non-zero entries for ρ (resp. ϕ).

As a class of non-examples, if $A_1, \ldots, A_K, B_1, \ldots, B_k$ are any square matrices and C_2, C_1 are composable matrices then, the set of DNNs of the form

$$ReLU \bullet (A_n x + b_n) \circ \cdots \circ ReLU \bullet (A_1 x + b_1) \tag{2}$$

are not universal and the input and output maps violate Assumptions 3.1 and 3.2.

Deeper Layers -Reviewer 2: *"[It's] common practice [to] change the internal structure of the architecture beyond the input and output layers...people often inject a particular inductive bias [into the DNN]"*.

Examples (1) and (2) show that in a DNN, the matrices $A_{i,j}$ ($j \neq 2$) can be chosen as we like so-long as they are of full-rank. Therefore, for a DNN to be universal, we only need the middle two layers, described by f, to be fully-connected. In particular, this gives us the flexibility of encoding many "inductive biases" into the architecture since only the two middle layers cannot be modified freely, as long as the involved matices' ranks are preserved.

Relation to Future Research -Reviewer 3: *Example: Generalizability via Dropout but while maintaining approximation capabilities*. Consider a DNN of the form $\rho \circ (A_{2,2} \circ \sigma \bullet (A_{1,2}x + b_{1,1}) + b_{2,2}) \circ \phi$ where ϕ and ρ are as in 1, B_i are arbitrary composable matrices and c_i are vectors of appropriate dimension. Since 1 only requires defining ρ and ϕ to be of full-rank but can be highly sparse. Therefore, Theorem 3.3 implies that if dropout is used to improve generalizability, it can only maintain universal approximation if the dropout procedure is constrained so that it preserves the matrix's rank. This is interesting, since the generalization effects of dropout are well-understood but the impact of drop-out on an architecture's approximation abilities, or more generally sparsely-connected DNNs, is so-far not.

Numerical Illustrations/Experiments - Reviewers 1-4: To illustrate the effect of properly (or poorly) choosing the networks' input and output layers we implement the architecture of (1) (Good), the architecture defined by (2) (Bad), and a shallow feed-forward network with no additional input and output maps (Vanilla) as a baseline model. Our implementations are on the California housing dataset [3], with the objective of predicting the median housing value, the test-set consists of 30% of the total data, pre-processing as in [2] and would be included in the camera-ready version. As anticipated, applying a readout and feature map satisfying our conditions can only improves the performance of our the architecture by learning a good representation of the data. In contrast, a poorly chosen feature map degrades the model's performance.

	Good	Bad	Vanilla		Good	Bad	Vanilla
MAE: Test	0.381672	2.073648	0.428122	MAE: Train	0.316039	5.637205	0.375244
RMSE: Test	0.420023	2.056685	0.434948	RMSE: Train	0.374318	5.548146	0.391001

Illustration of Stakes - Reviewer 4: *"One can hope that something as simple as the softmax function ... does not spoil the UA of the previous layers"*. It is not surprising that the softmax function preserve's the ability for an architecture to approximate any classifier point-wise in $[0,1]^n$ and uniformly in $(0,1)^n$. However, point-wise convergence of a deep-classifier to any "non-fuzzy classifier" (taking values in $[0,1]^n - (0,1)^n$) is not robust since this means that selected network depends on the size of the training set, both in practice and in theory. Theorem 3.9 guarantees, amongst other things, that a single network can theoretically be trained which approximately performs the classification task with uniform precision on all of X for any classifier, even the "non-fuzzy classifiers" taking values in $[0,1]^n - (0,1)^n$. Similar issues arise with the other mentioned examples and we would be happy to add a brief discussion outlining each.

[1] Guilbault, Craig R. and Tirel, Carrie J. On the dimension of Z-sets. Topology Appl., 160, 2013, 1.

[2] A. Geron, Handson-ML. Accessed: 2020-05-15.

[3] Kaggle. California housing prices. Accessed: 2020-05-15.