VQ-Kernels: Unraveling Deep Learning of High-Dimensional Data Geometry

T. Mitchell Roddenberry*
Rice University

Thomas Walker
Rice University

Ronald R. Coifman
Yale University

Randall Balestriero
Brown University

Richard G. Baraniuk
Rice University

Editors: Marco Fumero, Clementine Domine, Zorah Lähner, Irene Cannistraci, Bo Zhao, Alex Williams

Abstract

Deep Networks (DNs) are state-of-the-art predictors, able to navigate billion dimensional spaces to produce compressed embeddings of datasets. While most of the focus has been on improving the performance of these embeddings, we ask instead a different question: *how can Deep Networks teach us about the data geometry.* Through the spline theory of DNs, we derive a novel kernel that characterizes DNs as vector quantizers implementing affine functions over a partition of the domain, where the regions are coupled in a manner not immediately obvious from the partition geometry. We employ this kernel in the interpretation of DNs, illustrating their internalization of the training data geometry.

1 Introduction

Deep Networks (DNs) can navigate high-dimensional spaces to yield effective compressed representations of a dataset; thus, interpreting and explaining DNs is important but challenging. There have been multiple paradigms shaping the progress in this direction, leading to pockets of progress, which, sparingly, culminate in satisfying insights into increasingly capable DNs. For example, Grad-CAM [8] provides a sample-wise understanding of a DN's behavior by exploring its gradients; however, its inherently visual and local application limits its broad applicability. Similarly, feature-wise techniques, such as LIME [7], SHAP [4], and Sparse Autoencoders (SAEs) [2], provide a more holistic understanding of a DN's behavior by going beyond individual samples, but fall short of generalizing insights. To overcome these challenges, there is a necessity to derive explanations from beyond point-wise statistics; especially as DNs become increasingly subjected to fine-tuning, distillation and extrapolation, it becomes important to understand the geometry of a DN's learned representations.

In this work, we appeal to the spline theory of DNs [1] to characterize the geometry of a DN's learned representation through its input space partitioning. Although this partitioning has been studied before, its combinatorial construction [5] has limited its practical applicability. In this work, we link the *neural activation patterns* of a DN when applied to a given input to the partition geometry, yielding a pseudodistance[2] on the domain that reflects the vector quantization (VQ) by the piecewise affine layers. In Fig. 1, we demonstrate how the eigenvectors of our proposed pseudodistance kernel reflect the geometry of the training data.

*Correspondence to TMR: mitch@roddenberry.xyz

[2] A pseudodistance is a metric distance that allows distinct points to have zero distance.

Proceedings of the III edition of the Workshop on Unifying Representations in Neural Models (UniReps 2025).

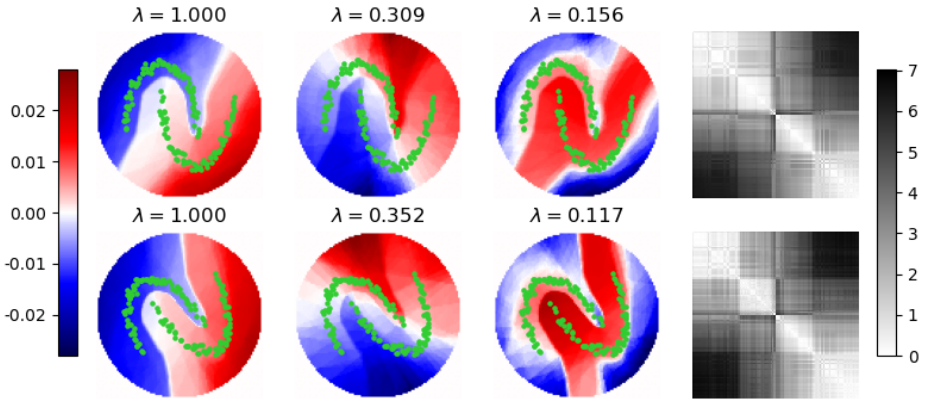

Figure 1: The vector quantization (VQ) of a deep network (DN) reflects the geometry of the training data. **(Top)** Dominant eigenvectors *(left)* of the VQ-Kernel *(right)* of a DN trained to classify a two-moons point cloud. **(Bottom)** Eigenvectors and distance matrix for a DN trained on a rotation of the same point cloud. The geometry of the learned partition approximately commutes with the rigid transformation of the training data, indicating that the DN adapts to the underlying geometry of the data.

2 VQ-Kernels: Tractable Spline Insights into Deep Learning Geometry

The Spline Theory of Deep Networks DNs with continuous piecewise affine (CPA) layers, namely those employing continuous piecewise affine (CPA) nonlinearities (e.g. ReLU), are known to themselves be CPA [1]. In particular, the corresponding linear regions partition the input space in a manner that can be characterized by the collection of hyperplanes induced from the level-sets of the DN's nonlinearities [3]. The structure of the input space as formed by this partitioning is our focus.

Consider a CPA DN composed of L CPA layers. The ℓ^{th} layer is defined as a map $f^{(\ell)} : \mathbb{R}^{D^{(\ell-1)}} \to \mathbb{R}^{D^{(\ell)}}$, with $z \mapsto \sigma\left(A^{(\ell)}z + b^{(\ell)}\right)$, for integers $D^{(\ell)}$, matrices $A^{(\ell)} \in \mathbb{R}^{D^{(\ell)} \times D^{(\ell-1)}}$, and vectors $b^{(\ell)} \in \mathbb{R}^{D^{(\ell)}}$. We put $D^{(0)} = D$. Assuming that $\sigma = \text{ReLU}$ is applied elementwise, this layer constitutes a CPA function with respect to the partition defined by the collection of hyperplanes $\mathcal{H}^{(\ell)} = \{(A^{(\ell)}_{j,:}, b^{(\ell)}_j)\}_{j=1}^{D^{(\ell)}}$, where we have identified each hyperplane in $\mathcal{H}^{(\ell)}$ with a row in $A^{(\ell)}$ and the corresponding entry in $b^{(\ell)}$. That is to say, for each binary vector $s \in \{-1, +1\}^{\mathcal{H}}$, there is a matrix-vector pair (A_s, b_s) such that for all $x \in \text{sign}^{-1}_{\mathcal{H}}(s)$, we have $f^{(\ell)}(x) = A_s x + b_s$.

The VQ-Kernel Since the input space geometry is constructed via hyperplanes derived from the parameters of the DN, it is clear that it is informative of the learned representations; we probe this structure through pseudometrics.

Let $\mathcal{H}$ be a collection of hyperplanes in $\mathbb{R}^D$, so that for each $H \in \mathcal{H}$, there is a corresponding pair $(a_H, b_H) \in \mathbb{R}^D \times \mathbb{R}$ where $H = \{x \in \mathbb{R}^D : \langle a_H, x\rangle + b_H = 0\}$. We exclude the degenerate case where $a_H = 0$, as the corresponding hyperplane is either the empty set or the entire domain. Correspondingly, we can assume without loss of generality that $\|a_H\| = 1$.

Each $H \in \mathcal{H}$ divides $\mathbb{R}^D$ into two regions: one where $\langle a_H, x\rangle + b_H \geq 0$, and the other where $\langle a_H, x\rangle + b_H < 0$. Then, the partition of $\mathbb{R}^D$ defined by $\mathcal{H}$ is determined by the patterns of intersections of these regions: this is easily encoded as a binary vector $\text{sign}_{\mathcal{H}} : \mathbb{R}^D \to \{-1, +1\}^{\mathcal{H}}$, with $x \mapsto [\text{sign}(\langle a_H, x\rangle + b_H)]_{H \in \mathcal{H}}$, where $\text{sign} : \mathbb{R} \to \{-1, +1\}$ is the signum function. This *quantization* of the domain $\mathbb{R}^D$ allows us to pull back simple metrics on binary strings $\{-1, +1\}^{\mathcal{H}}$ to yield pseudodistances on the domain. In particular, we pull back the *Hamming distance* to define the pseudodistance

$$d_{\mathcal{H}} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{\geq 0}, \quad (x, y) \mapsto d_H(\text{sign}_{\mathcal{H}}(x), \text{sign}_{\mathcal{H}}(y)),$$

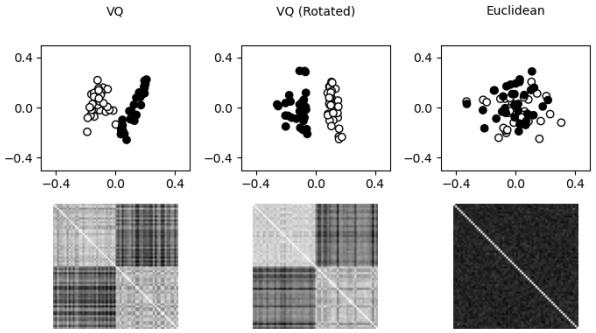

Figure 2: The vector quantization (VQ) of a DN reflects the geometry of high-dimensional training data. **(Top)** MDS embedding of VQ-Kernel trained on high-dimensional two-moons data (left), a rotated version of the same data (center), and the Multidimensional Scaling (MDS) embedding of the Euclidean distance matrix (right). **(Bottom)** Distance matrices. Observe that the low-dimensional structure of the training data is apparent in the VQ-Kernels, as opposed to the Euclidean distance matrix.

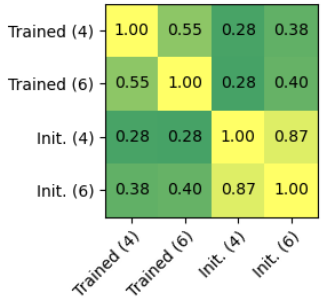

Figure 3: Cosine similarities of heat kernels derived from VQ-Kernels for two models with different architectures, pre- and post-training on MNIST. Although not identical, the VQ-Kernels of networks with different architectures become more similar when trained on the same dataset.

where the Hamming distance d_H is the number of hyperplanes in H where the signs of the arguments differ. It is clear that $d_{\mathcal{H}}$ indeed satisfies the nonnegativity, reflexivity, and triangle inequality properties of a distance, but evaluates to zero if both inputs are contained in the same partition region defined by $\mathcal{H}$. This implies that if x, x' are contained in the same partition region, $d_{\mathcal{H}}(x, y) = d_{\mathcal{H}}(x', y)$.

Observe that $d_{\mathcal{H}}$ measures the pattern of relations of points to each hyperplane, amounting to a distance defined on the partition itself, where two sets in the partition are close if they are on the same "side" of a large number of hyperplanes. Define the ℓ^{th} partial DN as $F^{(\ell)} : \mathbb{R}^{D^{(0)}} \to \mathbb{R}^{D^{(\ell)}}$, with $x \mapsto (f^{(\ell)} \circ F^{(\ell-1)})(x)$, where $F^{(0)}$ is the identity function. For each $\ell \geq 1$ and pair of points $x, y \in \mathbb{R}^D$, define

$$d^{(\ell)}(x, y) = d_{\mathcal{H}^{(\ell)}}(F^{(\ell-1)}(x), F^{(\ell-1)}(y)).$$

When needed, we will also write $d^{(\ell)}(x, y; \mathcal{N})$, where $\mathcal{N}$ gathers the parameters and architecture of the DN. In other words, $d^{(\ell)}$ is the pullback of $d_{\mathcal{H}^{(\ell)}}$ via the map $F^{(\ell-1)}$.

Given a finite set of such layers indexed by $\ell = 1, \ldots, L$, we define the overall pseudodistance on $\mathbb{R}^D$ as the sum of each of the L intermediate pseudodistances:

$$d(x, y; \mathcal{N}) = \sum_{\ell=1}^{L} w_\ell d^{(\ell)}(x, y; \mathcal{N}),$$

for scalars $w_\ell > 0$. Unless otherwise specified, we take $w_\ell = 1$. This yields the *VQ-Kernel*:

Definition 1. For a DN with parameters $\mathcal{N}$, the *VQ-Kernel* of a sample $\mathcal{U} = (u_i)_{i=1}^m \subseteq \mathbb{R}^d$ is the matrix $K = (k_{ij}) \in \mathbb{R}^{m \times m}$ where $k_{ij} = d(u_i, u_j; \mathcal{N})$.

Partition distances. For a partition Ω determined by a set of hyperplanes $\mathcal{H}$ on a domain $\mathbb{R}^D$, the distance $d_{\mathcal{H}}$ is the natural pullback of an incidence-based distance on Ω. Define a distance d on Ω so that for two regions $\omega_1, \omega_2 \in \Omega$, $d(\omega_1, \omega_2) = 0$ if $\omega_1 = \omega_2$, and $d(\omega_1, \omega_2) = 1$ if ω_1 and ω_2 have a shared face. Otherwise, $d(\omega_1, \omega_2)$ is defined as the minimal distance that obeys the triangle inequality and the above constraint. Using this distance on Ω, we define a pseudodistance on $\mathbb{R}^D$ via the pullback of the projection map $\mathcal{P}_\Omega : \mathbb{R}^D \to \Omega$, that is, $d_\Omega(x, y) = d(\mathcal{P}_\Omega(x), \mathcal{P}_\Omega(y))$.

Lemma 1. *For a partition Ω determined by a (nondegenerate) set of hyperplanes $\mathcal{H}$ on a domain $\mathbb{R}^D$, the pseudodistances d_Ω and $d_{\mathcal{H}}$ are identical. (Proof in Appendix B.1)*

For a network $\mathcal{N}$ with one hidden layer, the partition Ω over which $\mathcal{N}$ is a piecewise affine map is exactly determined by the set of hyperplanes in the hidden layer. Hence, Lemma 1 demonstrates that such networks yields distances $d(\cdot, \cdot; \mathcal{N})$ that are identical to the partition geometry distance $d_\Omega(\cdot, \cdot)$. However, when $\mathcal{N}$ is a *deep network* (composed of multiple layers), the distance defined by the network is only *upper bounded by* the partition distance, summarized in the following lemma.

Lemma 2. *Let Ω be a partition of $\mathbb{R}^D$ determined by a (generic) DN $\mathcal{N}$. Then, for any points $x, y \in \mathbb{R}^D$, $d(x, y; \mathcal{N}) \leq d_\Omega(x, y)$. (Proof in Appendix B.2)*

Lemma 2 indicates the difference between the distance $d(\cdot, \cdot; \mathcal{N})$ and the corresponding partition distance d_Ω. As illustrated in Fig. 5, the compositional nature of the DN creates "shortcuts" between regions of the domain $\mathbb{R}^D$ that are not visible through immediate examination of the final partition alone. In the regime of Lemma 1, the VQ partition distance between points simply counts the number of regions separating them. In the setting of Lemma 2 attained by a DN, the VQ distance reflects the *hierarchical* partitioning of the domain by the composition of layers. See Appendix B.2 for further discussion.

3 Applications

3.1 Comparing DNs

The comparison of DNs beyond their realization as function is difficult, particularly in high dimensions. We apply the VQ-Kernel as a means to compare the geometric structures internalized by DNs with different architectures. Consider two ReLU MLPs: one with 4 layers, and the other with 6 layers.[3] We train both models to convergence on the MNIST digits training set, and then compute their VQ-Kernels on the test set. We also compute the VQ-Kernels using the randomly initialized set of parameters for each model. With these kernels in hand, we consider to what extent the VQ-Kernel is dependent on the training data, and to what extent it is dependent on the architecture.

To do so, we evaluate cosine-similarity between heat kernel matrices derived from the VQ-Kernel, *i.e.*, the matrix $H = \{h_{ij}\}$ such that $h_{ij} = \exp(-d(u_i, u_j; \mathcal{N})/\sigma^2)$ for some $\sigma^2 > 0$. We illustrate the results of this for both architectures (pre- and post-training) in Fig. 3. Although the kernels of the trained models are not identical, they are more similar to each other than at initialization, suggesting that the VQ-Kernel reasonably reflects the geometry of the training data, even when evaluated on unseen points.

3.2 Understanding Coordinate-Free Data Geometry

The complex coupling of affine regions in DNs offers a plausible explanation of how these networks are able to approximate functions on low-dimensional structures in high-dimensional ambient spaces. To illustrate that the VQ-Kernel captures this, we consider a high-dimensional variant of the two-moons classification problem. We consider a two-class classification dataset in $\mathbb{R}^2$, then concatenate each point $u_i \in \mathbb{R}^2$ with a random vector $z_i \in \mathbb{R}^{98}$ such that the resulting vector in $\mathbb{R}^{100}$ has unit norm. We then train an MLP on the original labels of the dataset, and examine the resulting VQ-Kernel evaluated on the training data. Similar to the example in Fig. 1, we repeat this procedure on a rotated version of the dataset, to illustrate the invariance to rigid transformations. In Fig. 2, we observe that the VQ-Kernel exhibits an approximately low-rank structure, reflecting the intrinsic dimensionality of the training data. The multidimensional scaling (MDS) embedding of the training data according to the VQ-Kernel demonstrates the separability of the classes. Furthermore, this low-dimensional structure is not visible when using the Euclidean metric, which yields an unstructured kernel and thus a poor embedding, as shown.

4 Conclusion

We have demonstrated that the VQ-Kernel, defined via the pullback of the Hamming distance applied to neural activation patterns, is a viable means of studying the geometry of a DN. Lemmas 1 and 2 demonstrate the properties of the VQ-Kernel in the shallow and deep architectural settings as it

[3] See Appendix A for further details on the experimental parameters.

relates to the partition of the domain into polytopes, over which the realized mapping is piecewise affine. We have also demonstrated via simple numerical experiments that the VQ-Kernel reflects the geometry of the data on which the network was trained, in both low and high-dimensional settings.

Limitations of this approach to be resolved in future work include leveraging the proposed VQ-Kernel for deep learning tasks such as model distillation or neural architecture search. Furthermore, better understanding of the theoretical properties of the VQ-Kernel for DNs remain an open challenging question we hope to investigate.

Acknowledgments and Disclosure of Funding

This work was supported by ONR grant N00014-23-1-2714, ONR MURI N00014-20-1-2787, DOE grant DE-SC0020345, and DOI grant 140D0423C0076.

References

[1] Randall Balestriero and Richard G Baraniuk. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109(5):704–727, 2020.

[2] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] Ahmed Imtiaz Humayun, Randall Balestriero, Guha Balakrishnan, and Richard G. Baraniuk. Splinecam: Exact visualization and characterization of deep network geometry and decision boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3789–3798, 2023.

[4] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 47684777, 2017.

[5] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *arXiv preprint arXiv:1402.1869*, 2014.

[6] Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 11351144, 2016.

[8] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pages 618–626, 2017.

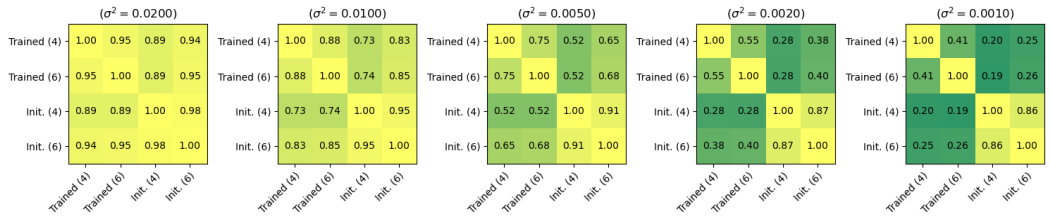

Figure 4: Cosine similarities of heat kernels derived from VQ kernels for two models with different architectures, pre- and post-training on MNIST, with varying scale parameter σ^2.

A Experimental Details

A.1 Section 1 (Figure 1)

We generate a two-moons binary classification dataset with $n = 150$ points in $\mathbb{R}^2$, scaled to be contained in the square $[-1, 1]^2$. For the alternate dataset, we rotate the coordinates by $\pi/6$ radians. We train ReLU MLPs with $L = 3$ hidden layers of width 96 using the binary cross-entropy loss until convergence.

We then compute the VQ kernels for each model over a the unit disc sampled according to 96×96 discretization of $[-1, 1]^2$, yielding two matrices $K, K^{\text{rot}} \in \mathbb{R}^{N \times N}$, corresponding to the models trained on the original training dataset and its rotated version, respectively. In Fig. 1, we compute the three dominant eigenvalue/eigenvector pairs of the MDS matrix $B = (I - J/N)K(I - J/N)$ normalized by the leading eigenvalue of B, where I denotes the identity matrix and J denotes the all-ones matrix. In the rightmost columns of Fig. 1, we show the VQ-Kernel for the models evaluated over their training datasets, ordered according to the labels.

A.2 Section 3.1 (Figure 3)

We split the MNIST digits dataset into a training set of $n_{tr} = 8000$ and a test set of $n_{ts} = 2000$ images, all of size 28×28. We train ReLU MLPs with $L \in \{4, 6\}$ layers, all of width 16. Then, we compute the VQ-Kernels over the test set for both the randomly initialized and trained models.

To compare, we take the cosine similarity between heat kernels applied to the VQ-Kernels, defined as the elementwise exponential $h_{ij} = \exp(-k_{ij}^2/\sigma^2)$, for some $\sigma^2 > 0$. We demonstrate the results of this for a variety of scale parameters σ^2 in Fig. 4.

A.3 Section 3.2 (Figure 2)

To demonstrate the advantage of a learned VQ-Kernel over kernels based on the ambient Euclidean metric, we develop a "high-dimensional two-moons" binary classification dataset. We begin with the binary classification dataset in $\mathbb{R}^2$ described in Appendix A.1. Then, for each training data point u_j, we draw a uniform random vector $z_j \sim \mathcal{U}\left(\sqrt{1 - \|0.1u_j\|^2}\mathbb{S}^{98}\right)$, where $\mathbb{S}^{98}$ denotes the unit sphere in $\mathbb{R}^{98}$. We then concatenate $0.1u_j$ and z_j to form a new data point in $\mathbb{S}^{100} \subset \mathbb{R}^{100}$. The scaling factor of 0.1 reduces the effective signal-to-noise ratio of the training data, although the useful information of the data points lie in a two-dimensional subspace. Once again, we also create an alternate version of the dataset rotated by $\pi/6$ radians in the "signal" subspace. We then repeat the process explained in Appendix A.1, this time using a ReLU MLP with $L = 3$ layers of width 64 and plot the MDS embedding of the VQ-Kernel evaluated on the training data for both the dataset and the rotated version.

B Proofs

B.1 Lemma 1

The lemma essentially follows from the discussion in [6, Section 1.2]; we provide more details here. Let $\mathcal{H}$ be a set of hyperplanes in general position defining a partition of $\mathbb{R}^D$ into a set of (potentially

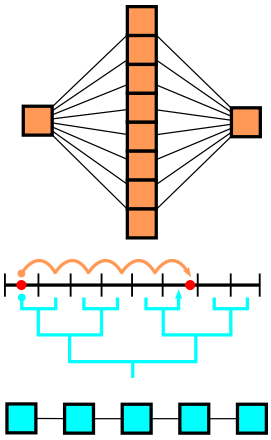

Figure 5: Networks with identical partitions may have different VQ distances. Although a wide, shallow network *(top)* and a deep, narrow network *(bottom)* may define an identical partition of the domain *(center)* as a set of polytopes, the VQ-Kernel more accurately characterizes the coupling of the affine regions in the deep regime.

unbounded) convex polytopes Ω. Let $x, y \in \mathbb{R}^D$ be given arbitrarily. Denote $r = d_\Omega(x, y)$, and their respective polytopes by $\omega_0 \ni x$ and $\omega_r \ni y$, so that there exists a sequence $[\omega_0, \omega_1, \ldots, \omega_r]$ where for each $j > 0$, ω_j shares a face with ω_{j-1} corresponding to a unique hyperplane $H \in \mathcal{H}$. Denote the sequence of such hyperplanes by $[H_1, H_2, \ldots, H_r]$. Observe that for $j > 0, v \in \omega_{j-1}, w \in \omega_j$, we have $d_{\mathcal{H}}(v, w) = 1$. By the triangle inequality for $d_{\mathcal{H}}$, it then follows that $d_{\mathcal{H}}(x, y) \leq d_\Omega(x, y)$. A similar argument establishes the inequality in the opposite direction, thus proving the lemma.

B.2 Lemma 2

Let $\mathcal{N}$ be a deep network defining a partition of $\mathbb{R}^D$ into a set of (potentially unbounded) polytopes Ω. Assuming nondegeneracy of the partitions defined by the deep network, let two polytopes $\omega_1, \omega_2 \in \Omega$ be given that share a face. Then, for some $\ell \geq 1$, there is a hyperplane $H \in \mathcal{H}^{(\ell)}$ such that ω_1 and ω_2 are divided by H, but are divided by *no other* hyperplanes in the network. Therefore, if $d_\Omega(x, y) = 1$, it is also the case that $d(x, y; \mathcal{N}) = 1$. The lemma then follows by the triangle inequality.

For the interested reader, we provide a specific example in $\mathbb{R}^1$ that attains a strict inequality $d(x, y; \mathcal{N}) < d_\Omega(x, y)$ for certain pairs of points x, y. Let $\mathcal{N}$ be a deep network with L layers such that $D^{(\ell)} = 1$ for all $\ell \geq 0$, and the corresponding affine maps are defined as

$$f^{(\ell)}(z) = 2z - \sigma(2z - 1).$$

This deep network can be thought of as a ReLU network with skip connections. Restricting to the unit interval $I = [0, 1]$, $\mathcal{N}$ defines a uniform partition of I into 2^L intervals, despite using only L neurons. That is to say, for all $x, y \in I$, $d(x, y; \mathcal{N}) \leq L$. At the same time, we can find points $x, y \in I$ such that $d_\Omega(x, y) \approx 2^L$, yielding a strict inequality $d(x, y; \mathcal{N}) < d_\Omega(x, y)$ for such points.

To achieve the same partition with a shallow network would require 2^L neurons, in which case Lemma 1 implies that $d(x, y; \mathcal{N}) = d_\Omega(x, y)$ for all $x, y \in I$, as illustrated in Fig. 5.