10-2022

Shell theory: A statistical model of reality

Wen-yan LIN
Singapore Management University, daniellin@smu.edu.sg

Siying LIU
Institute for Infocomm Research

Changhao REN
Singapore University of Technology and Design

Ngai-Man CHEUNG
Singapore University of Technology and Design

Hongdong LI
Australian National University

See next page for additional authors

Citation

LIN, Wen-yan; LIU, Siying; REN, Changhao; CHEUNG, Ngai-Man; LI, Hongdong; and MATSUSHITA, Yasuyuki. Shell theory: A statistical model of reality. (2022). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 44, (10), 6438-6453.
Available at: https://ink.library.smu.edu.sg/sis_research/6222

Shell Theory: A Statistical Model of Reality

Wen-Yan Lin, Siying Liu, Changhao Ren, Ngai-Man Cheung, Hongdong Li, Yasuyuki Matsushita

Abstract—The foundational assumption of machine learning is that the data under consideration is separable into classes; while intuitively reasonable, separability constraints have proven remarkably difficult to formulate mathematically. We believe this problem is rooted in the mismatch between existing statistical techniques and commonly encountered data; object representations are typically high dimensional but statistical techniques tend to treat high dimensions a degenerate case. To address this problem, we develop a dedicated statistical framework for machine learning in high dimensions. The framework derives from the observation that object relations form a natural hierarchy; this leads us to model objects as instances of a high dimensional, hierarchal generative processes. Using a distance based statistical technique, also developed in this paper, we show that in such generative processes, instances of each process in the hierarchy, are almost-always encapsulated by a distinctive-shell that excludes almost-all other instances. The result is shell theory, a statistical machine learning framework in which separability constraints (distinctive-shells) are formally derived from the assumed generative process.

Index Terms—high dimension, statistics, semantic manifold, anomaly detection, one-class learning, life-long learning, incremental learning, hierarchical models, generative models

1 Introduction

We view the world through the lens of semantic concepts like cats and dogs, houses and mountains, streets and cars. Such semantics help us to frame our thoughts, communicate with our neighbors, formulate machine learning problems. Semantics seem so intuitive, we seldom think about them; however, when we do, semantics begin to seem rather peculiar.

The salient feature of semantics is its ability to unambiguously group a huge variety of different objects into a single class. For instance, members of the cat class can have long or short fur; be big or small; be orange, white or black. Despite the variations, we almost invariably agree on what is and is not a cat. This raises a series of fundamental but overlooked questions: Why is semantic classification possible? What are the constraints underlying it? And perhaps, most importantly, what are semantics?

To formulate principled answers to these questions, we attempt to develop a mathematical model of the generative processes that give rise to our reality, At first, modeling reality seems impossibly difficult. However, language can itself be considered a successful model of reality. This leads us to a solution based on mathematically mirroring the semantic structure of language.

Our model is based on two key observations. First, semantics are rarely (if ever?) defined in terms of a single attribute. Rather, semantics appear to reference patterns that only emerge when considering multiple attributes. This suggests a formulation based on high dimensional statistics. Second, each object seems to be simultaneously a member of several hierarchically related semantic classes. For instance, a Siamese cat is also a cat, a mammal, an animal, etc. This suggests a hierarchical relationship between key statistical dependencies. Using these observations, we develop the following mathematical representation of reality:

- Objects are represented by high dimensional vectors of attributes. These are also termed data-points;
- Data-points are outcomes of some high dimensional, stochastic generative processes;
- Generative processes are related to one another through a global hierarchy;
- Semantics are symbolic representations of individual generative processes.

The challenge lies in fusing these representations into a coherent mathematical model. Unfortunately, current statistical techniques tend to breakdown as dimensions increase. This is because most statistical techniques are formulated in terms of probability density. Other things being equal, the volume of a statistical sample space increases exponentially with the number of dimensions, making high dimensional sample spaces vast and sparsely populated; this in turn, makes density ill-defined and causes the failure of many basic statistical tools.

We address this problem by developing a distance based approach to statistics. The vastness of a high dimensional sample spaces makes co-incidental similarity extremely rare. Thus, high dimensional distances can acts as proxies for statical dependence and vice-versa, providing us with a simple but effective mechanics for statistical analysis. Applying this to high dimensional, hierarchal generative processes, we show that the instances (objects) of each generative process (sematic label) are almost-always be encapsulated by a distinctive-shell (separability constraint) that excludes almost-all other instances. We term this phenomenon

- *Wen-Yan Lin is with Singapore Management University. e-mail: daniellin@smu.edu.sg*
- *Siying Liu is with Institute for Infocomm Research (I2R). email:liusy1@i2r.a-star.edu.sg*
- *Hongdong Li is with Australian National University. email: hongdong.li@anu.edu.au*
- *Ngai-Man Cheung and Changhao Ren are with Singapore University of Technology and Design. email:ngaiman_cheung@sutd.edu.sg; changhao_ren@sutd.edu.sg*
- *Yasuyuki Matsushita is with Osaka University. email:yasumat@ist.osaka-u.ac.jp*

shell theory. [1]

Distinctive-shells provide the first, explicit mathematical representation of the separability constraints underlying machine learning; reformulating/ reinterpreting algorithms in terms of the separability constraints provides a refreshingly new perspective to many machine learning problems. Examples discussed in this paper include:

Surprisingly simple manifold discovery: Distinctive-shells are probably the often hypothesized semantic manifold. As distinctive-shells can be directly estimated from data statistics, complex manifold discovery algorithms can be replaced with four lines of Python code.

Shell-Normalization: We show how the sensitivity of distinctive-shell's can be tailored to specific problems through a data pre-processor which we term shell-normalization. From this perspective, traditional normalization is a special case of shell-normalization which is tailored to the multi-class classification problem.

From One Class to Multiple Classes: Shell theory provides a framework whereby independently trained one-class learners can be fused to form a multi-class classifier. This unifies the two problems into a learning mechanics which is eerily similar to that employed by humans.

Anomaly Detection: Shell theory shows how anomaly detection can be formulated as a robust least squares shell-fitting. This allows for the development of algorithms that are simple, fast and reliable.

Statistical Maximum Distance: The geometric maximum distance between two unit-normalized data-points is 2. Shell theory makes the quirky prediction that there exists a "statistical maximum distance" of $\sqrt{2}$, which is much less than the geometric maximum.

In summary: This paper takes an unorthodox approach to machine learning. Rather than the traditional focus on designing algorithms for discovering semantic representations, we focus on understanding the natural patterns that semantics reference. As one of the first works in this direction, it is almost-surely incomplete; however, we feel it is essential to start a conversation on this topic. After all, without understanding the semantics used for understanding, have we understood anything [1]?

1.1 Related Works

This paper builds on a foundation established by the wider machine learning community. One set of works we found especially inspiring were those linking Newtonian mechanics to the semantic manifold hypothesized by the machine learning community [2], [3], [4]. This is an intriguing insight. However, modeling an entire generative process makes for very complex mathematics; hence, it is hard to make specific machine learning predictions with such models. To simplify the mathematics, shell theory only models key aspects of the generative process.

One key aspect is surely high dimensions. Indeed, a number of works have already used high dimensional statistics to derive conclusions that are similar to the aforementioned physics based techniques [5], [6], [7]. However, such mathematical techniques are still in their infancy and conclusions often differ wildly. Thus, some models claim high dimensions are intrinsically cursed [6], [7], [8], while others use the same constraints to show it is a blessing [5], [9], [10]. Shell theory provides a unifying framework in which it is possible to derive both curse and blessing by tweaking the assumed statistical dependency.

Shifting our focus from the fields of physics and mathematics to that of machine learning, we see a similar push for more interpretable techniques [4], [11], [12], [13], [14], better understanding of the generative processes [15], [16], [17], [18], [19], [20], [21], [22], [23] and hypothesized existence of naturally occurring groupings [5], [24]. Gaussian Mixture Models and Expectation Maximization [25], [26], [27] are early examples of the use of statistics for interpretable machine learning. The tradition has since been extended to deep-learning, with notable examples being variational auto-encoders [28], [29], contrastive-loss [24], [30] and triplet-loss [31], [32].

In traditional machine learning, statistical formulations are almost density based; however, they typically avoid explicit estimation of sample density. Rather, they tend to assume that probability densities follow a Gaussian distribution, that can be parametrized in terms of its centroid. This leads to Gaussian mixture algorithms that are remarkably similar to algorithms derived through shell theory. Perhaps unsurprisingly, such algorithms are often effective even in high dimensions. Their drawback is the occasional gap between theoretical predictions and empirical observation (for example, classification may be accurate even if sample density does not corresponds to the algorithm's predicted probability density); this sometimes leads to "mysterious failures" that are attributed to the "curse-of-dimensions".

From the perspective of traditional machine learning, shell theory can be seen as providing an alternative, more rigorous, derivation of the traditional Gaussian mixture algorithm; this provides better alignment of theoretical predictions with empirical observation and allows statistical analysis to be extended to new domains like semantic manifolds and normalization.

Finally, shell theory has a rather interesting relationship with artificial neural networks (deep-learners). Neural networks like VGG [33], ResNet [34], GAN [35], [36] and variational auto-encoders [28], [29] are very effective at specific tasks. However, networks must be customized for each task, leading to a proliferation of problem specific networks. By making the separability constraints explicit, shell theory allows a single a pre-trained neural network to perform well on many different machine learning problems (Sec. 6, Sec. 7, Sec. 8), with an accuracy that is comparable to or better than the state-of-the-art.

2 Preliminaries

When developing statistical models, the instinct is to think in terms of probability density functions; however, density becomes an increasingly ill-defined concept, as the number of dimensions increase. Other things being equal, the volume of a statistical sample space will increase exponentially with the number of dimensions. As a result, high dimensional sample spaces tend to be huge. In fact, they may be so huge that even all data in the world can only populate them sparsely. This means that in high dimensions, probability density often makes a poor approximation of sample density and vice-versa.

Although increasing dimensions take away a statistical tool, they provide another. Vast sample spaces make coincidental similarity is rare. If so, distances between high dimensional random vectors would reflect statistical dependencies and vice-versa. We formalize this intuition using a set of theorems, which make

1. The term "shell", is a reference to the surface of a high dimensional hyper-sphere.

distance based statistical analysis possible. We use these theorems to statically model the origins of our data, giving rise to shell theory.

Shell theory introduces concepts, notations and lemmas, which may seem intimidatingly unfamiliar. To aid understanding, we have traced each derivation back to first principles; we also provide an introduction to statistical modeling, random variables and high dimensions, which we urge all readers to peruse.

2.1 Mathematical Preliminaries

Following the conventions in statistics, we use random variables to denote the outcomes of stochastic events. Random variables are always capitalized; thus, the outcome of a dice throw may be denoted by random variable Z.

Random variables are place-holders for undetermined outcomes and have no specific value. Specific outcomes are referred to as instances or data-points. These are denoted by the small letter variant of their respective random variables, with z representing an instance of Z.

Random vectors are formed by concatenating two or more random variables. Both ordinary vectors and random vectors are boldfaced. Thus, the outcome of ten dice rolls may be denoted by a ten dimensional random vector $\mathbf{Z}$, with $\mathbf{z}$ representing an instance of $\mathbf{Z}$.

A reference list of the notations used in this paper is placed below:

Definition 1.

- *$d^2(.)$ denotes an operator for normalized squared ℓ_2 norm, such that for $\mathbf{x} \in \mathbb{R}^k$, $d^2(\mathbf{x}) = \frac{\|\mathbf{x}\|^2}{k}$. If $\mathbf{x}$ is the difference between two vectors, we refer to $d^2(\mathbf{x})$ as **normalized squared distance**, or distance for short;*
- *$d(.)$ is the normalized ℓ_2 norm operator, $d(.) = \sqrt{d^2(.)}$;*
- *$\mathrm{S}(\boldsymbol{\mu}, r)$ denotes a thin shell centered at $\boldsymbol{\mu}$, with radius r.*

Let $\mathbf{Z} = [\mathbf{Z}[1], \mathbf{Z}[2], \ldots, \mathbf{Z}[k]]^T$ denote a k-dimensional random vector where $\mathbf{Z}[i]$ is a random variable,

- *$\mathbf{z}$ denotes the instantiation of $\mathbf{Z}$;*
- *$\mathbf{Z}$ is **high dimensional** if and only if its fourth order moments are finite; its dimension $k \to \infty$; And there are only a finite number of pairwise dependencies between its dimensions;*
- *$d^2(.)$ operator can be applied to random vectors. $d^2(\mathbf{Z}) = \frac{\|\mathbf{Z}\|^2}{k}$ is a random variable formed by averaging $\mathbf{Z}$'s squared elements;*
- *$\boldsymbol{\mu}_{\mathbf{Z}} = \mathbb{E}(\mathbf{Z}) = [\mathbb{E}(\mathbf{Z}[1]), \ldots, \mathbb{E}(\mathbf{Z}[k])]^T$ is a vector of each dimension's expectation;*
- *$v_{\mathbf{Z}} = \sum_{i=1}^{k} \frac{\mathrm{var}(\mathbf{Z}[i])}{k}$ is the average variance;*
- *$\overset{a.s}{=}$ denotes almost-surely-equal. Thus, if as $k \to \infty$,*

$$P(\|d^2(\mathbf{Z} - \mathbf{c}) - t\| < \epsilon^2) \to 1, \quad \epsilon > 0,$$

for an arbitrarily small ϵ, is written $d^2(\mathbf{Z} - \mathbf{c}) \overset{a.s.}{=} t$.

***Unit-vector-normalization** refers to the data pre-processing step in which each instance is subtracted by some common vector (also known as the normalization vector) and the resultant rescaled to a unit-vector;*
As dimension $k \to \infty$, unit-vector-normalization makes individual entries to tend to 0. Thus, for unit-vector-normalized data, the definitions of $d^2(.)$ and $v_{\mathbf{Z}}$ must be modified to avoid a division by k:

- *$d^2(.)$ is a squared ℓ_2 norm, such that $d^2(\mathbf{Z}) = \|\mathbf{Z}\|^2$;*
- *$v_{\mathbf{Z}} = \sum_{i=1}^{k} \mathrm{var}(\mathbf{Z}[i])$, is the total variance.*

When the number of independent dimensions in random vectors becomes large, euclidean distances can be predicted using the law-of-large-numbers. More formally, if the number of independent dimensions tends to infinity, the euclidean distance between $\underline{\text{independent}}$ random vectors $\mathbf{A}$ and $\mathbf{B}$ is [5]**:**

$$d^2(\mathbf{A} - \mathbf{B}) \overset{a.s.}{=} v_{\mathbf{A}} + v_{\mathbf{B}} + d^2(\boldsymbol{\mu}_{\mathbf{A}} - \boldsymbol{\mu}_{\mathbf{B}}), \tag{1}$$

where $\boldsymbol{\mu}_{\mathbf{A}}, \boldsymbol{\mu}_{\mathbf{B}}$ and $v_{\mathbf{A}}, v_{\mathbf{B}}$ represent the mean and average variance of their respective distributions; $\overset{a.s.}{=}$ denotes almost-surely equal; and $d^2(.)$ is the squared, ℓ^2 norm divided by the number of dimensions.

Note that Eq. (1) comes with a caveat. The law-of-large-numbers requires the number of independent random variables to be much larger than data variability. This caveat is usually satisfied given a sufficiently large number of independent dimensions; however, a pathological exception occurs if some dimensions have infinite variability. Thus, Eq. (1) requires the underlying distributions of $\mathbf{A}$ and $\mathbf{B}$ have finite fourth order moments in all dimensions [2].

To avoid the pathological exception discussed above, we define a high dimensional random vector as one whose: 1) Underlying distributions have finite fourth order moments; 2) Dimensions tend to infinity; 3) Has a finite number of dependent dimensions [5]. The multivariate probability density function associated with a high dimensional random vector is termed a high dimensional distribution.

Equation (1) leads to two main results. First, if $\mathbf{A}_1, \mathbf{A}_2$ are independent, identically distributed (i.i.d.), high dimensional random vectors, with average variance $v_{\mathbf{A}}$,

$$d^2(\mathbf{A}_1 - \mathbf{A}_2) \overset{a.s.}{=} 2v_{\mathbf{A}}. \tag{2}$$

Second, if $\mathbf{c}$ is the outcome of some random process that is $\underline{\text{independent}}$ of $\mathbf{A}$, we can set $\mathbf{B}$ in Eq. (1) to be a distribution of $\underline{\text{mean } \mathbf{c}}$ and variance zero. This yields:

$$d^2(\mathbf{A} - \mathbf{c}) \overset{a.s.}{=} v_{\mathbf{A}} + d^2(\boldsymbol{\mu}_{\mathbf{A}} - \mathbf{c}), \quad \forall \mathbf{c} \in \mathbb{R}^k. \tag{3}$$

Together, Eq. (1), Eq. (2) and Eq. (3), form the foundational distance relations we use for statistical analysis in high dimensions.

2.2 Mathematics meets Reality

Let the set of all objects be denoted:

$$s_o = \{\mathbf{a}_1, \mathbf{a}_2, \ldots\}.$$

Each object can be considered an outcome of some stochastic process. The stochastic processes are represented by a corresponding set of random vectors

$$S_0 = \{\mathbf{A}_1, \mathbf{A}_2, \ldots\}.$$

Statistical modeling can be interpreted as the attempt to make reasonable assumptions on the nature of the random vectors in S_0, such that the overall model becomes simple enough to be useful, yet remains complex enough to be realistic.

Many statistical models make the assumption that the random vectors in S_0 are independent, identically distributed (i.i.d.); i.e., they share a common probability distribution; and the outcome of each random vector has no impact on the outcome of any other random vector. Is this realistic?

2. More research may allow the constraint to be relaxed from fourth order to second order.

Recall that the vastness of high dimensional spaces make coincidental similarity rare. As a result, the outcomes of i.i.d. random vectors tend to be very unique. This uniqueness is so extreme that instances exhibit complete distinctiveness, with no two instances being more similar (closer) to each other, than they are to any other instance. This is proven in Eq. (2).

Natural data is certainly very unique. Often, the uniqueness of individual objects can be assumed, even when there is no mechanism that enforces uniqueness. Examples the reader may be familiar with include: finger-prints, snowflakes and human faces. However, it is much harder to accept a claim that natural data is completely distinctive. Surely humans resemble each other more than they do snowflakes.

Perhaps the previous example was too extreme and analysis needs to be restricted to a smaller set of objects, such as animals. However, the problem reoccurs. Animals can be divided into species, each of whose individuals is more similar to each other than they are to other species. Species can be further divided into sub-species whose members are more similar to each other than they are to other members of their parent species. In fact, it appears possible to iterate this process indefinitely, defining increasingly smaller groups based on ever finer definitions of similarity and difference. The result is a hierarchical pattern of similarity and difference that seems to be reflected in the hierarchical organization of semantics.

We posit that the hierarchy of similarity and differences arise from an innately hierarchical, generative process where dependencies between random vectors are caused by the sharing of common ancestors. To formalize this hypothesis, we propose a high dimensional generative process, governed by a few simple rules.

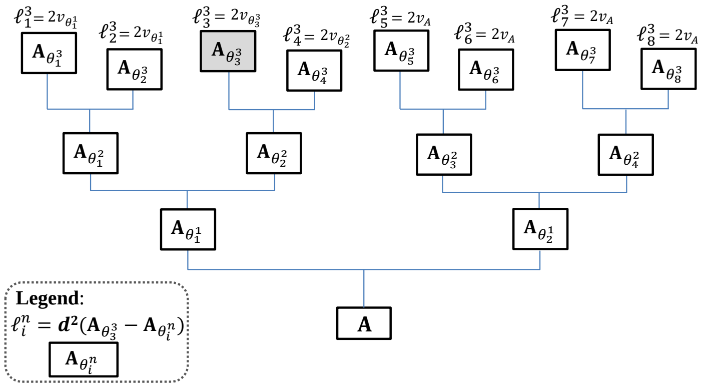
3 Hierarchical Generative Process

We define a hierarchical generative process as a stochastic model of data generation governed by the following rules:

1) The process starts with a single, parent data generator, known as the root generator;
2) A parent generator stochastically generates some number of child generators;
3) Child generators are conditionally independent given their parents;
4) Child generators themselves become parents, creating the next layer of generators. The process is iterated, creating multiple layers of generators;
5) Data-points are stochastic outcomes of the last layer of generators.

While the root generator only creates child generators directly, all data-points can be considered (indirect) stochastic outcomes of it. We use a random vector $\mathbf{A}$ to denote a data-point outcome by the root generator. The mean and average variance of $\mathbf{A}$ are denoted by $\boldsymbol{\mu}_{\mathbf{A}}, v_{\mathbf{A}}$ respectively. A specific instantiation of $\mathbf{A}$ is denoted by $\mathbf{a}$.

Random vector $\mathbf{A}_{\boldsymbol{\theta}^m}$ denotes a data-point outcome of an intermediate generator at the m^{th} level. $\mathbf{A}_{\boldsymbol{\theta}^m}$'s distribution parameters are denoted by $\boldsymbol{\theta}^m$, while its mean and average variance are denoted by $\boldsymbol{\mu}_{\boldsymbol{\theta}^m}, v_{\boldsymbol{\theta}^m}$ respectively. The root generator is considered level zero. Thus $\mathbf{A} = \mathbf{A}_{\boldsymbol{\theta}^0}$. To simplify notation, $\mathbf{A}_{\boldsymbol{\theta}^m}$ is sometimes used to reference the generator itself. When there is a need to distinguish between different generators at the m^{th} level, we add a subscript index and write $\mathbf{A}_{\boldsymbol{\theta}^m_k}$.



$\ell_1^3 = 2v_{\theta_1^1}$
$\ell_2^3 = 2v_{\theta_1^1}$
$\ell_3^3 = 2v_{\theta_3^3}$
$\ell_4^3 = 2v_{\theta_2^2}$
$\ell_5^3 = 2v_A$
$\ell_6^3 = 2v_A$
$\ell_7^3 = 2v_A$
$\ell_8^3 = 2v_A$
$\mathbf{A}_{\theta_1^3}$ $\mathbf{A}_{\theta_2^3}$ $\mathbf{A}_{\theta_3^3}$ $\mathbf{A}_{\theta_4^3}$ $\mathbf{A}_{\theta_5^3}$ $\mathbf{A}_{\theta_6^3}$ $\mathbf{A}_{\theta_7^3}$ $\mathbf{A}_{\theta_8^3}$
$\mathbf{A}_{\theta_1^2}$ $\mathbf{A}_{\theta_2^2}$ $\mathbf{A}_{\theta_3^2}$ $\mathbf{A}_{\theta_4^2}$
$\mathbf{A}_{\theta_1^1}$ $\mathbf{A}_{\theta_2^1}$
$\mathbf{A}$
Legend:
$\ell_i^n = \boldsymbol{d}^2(\mathbf{A}_{\theta_3^3} - \mathbf{A}_{\theta_i^n})$
$\mathbf{A}_{\theta_i^n}$

Fig. 1: Visualizing a hierarchical generative process. Generators are denoted $\mathbf{A}_{\boldsymbol{\theta}^n_i}$. Squared distance of $\mathbf{A}_{\boldsymbol{\theta}^3_3}$ to every other data-point is almost-surely two times the average variance of their most recent common ancestral generator. From Corollary 1, $v_{\mathbf{A}} > v_{\boldsymbol{\theta}^1_1} > v_{\boldsymbol{\theta}^2_2} > v_{\boldsymbol{\theta}^3_3}$. Thus, more closely related data-points are also geometrically closer to each other.

Any data-point from an M level hierarchical generative process is simultaneously the outcome of M ancestral generators. Thus it is simultaneously an instance of each member of the set,

$$\{\mathbf{A}, \mathbf{A}_{\boldsymbol{\theta}^1}, \mathbf{A}_{\boldsymbol{\theta}^2}, \ldots, \mathbf{A}_{\boldsymbol{\theta}^M}\}, \tag{4}$$

where $\mathbf{A}_{\boldsymbol{\theta}^i}$ denotes its i^{th} level ancestor. Each $\mathbf{A}_{\boldsymbol{\theta}^{i+1}}$ generator can be considered the result of its $\mathbf{A}_{\boldsymbol{\theta}^i}$ parent being constrained by some additional information.

3.1 Distances in High Dimension

A high dimensional hierarchical generative process occurs when every $\mathbf{A}_{\boldsymbol{\theta}^m}$ in the generative process is a high dimensional random vector. We show that this leads to the distinctive pattern of similarity and difference that is commonly found in nature.

Let $\mathbf{a}_1, \mathbf{a}_2$ denote a pair of data-points generated by a high dimensional, hierarchical generative process. Due to rule 3, $\mathbf{a}_1, \mathbf{a}_2$, can be considered to have been independently generated by their most recent common generator, whose associated random vector is denoted $\mathbf{A}_{\boldsymbol{\theta}^j}$. From Eq. (2) their distance from each other is

$$a.s. \quad d^2(\mathbf{a}_1 - \mathbf{a}_2) = 2v_{\boldsymbol{\theta}^j}. \tag{5}$$

where $a.s.$ denotes almost-surely. More formally:

Theorem 1. *(**Pairwise Distance**) The normalized squared distance between any two data-points created by a high dimensional hierarchical generative process is almost-surely $\sqrt{2v}$, where v is the average variance of their most recent common generator.*

Theorem 1 implies the pattern of distances illustrated in Fig. 1. Next, we show how this pattern can be interpreted.

In the hierarchical generative process, any intermediate level generator, $\mathbf{A}_{\boldsymbol{\theta}^m}$, is simultaneously a root generator for its own hierarchical generative process. Thus, the data-point generation by $\mathbf{A}_{\boldsymbol{\theta}^m}$ can be decomposed into two stochastic steps: First, by iteratively applying rules 1-4 of the hierarchical generative process, create a descendant generator j levels below $\mathbf{A}_{\boldsymbol{\theta}^m}$; Next, generate a data-point from that descendant generator. We use $\mathbf{A}_{\boldsymbol{\theta}^m \boldsymbol{\Theta}^j}$ to reference this two step process. The parameters of the descendant generator are denoted with random variable $\boldsymbol{\Theta}^j$, where the superscript j denotes the number of levels the descendant generator is below $\mathbf{A}_{\boldsymbol{\theta}^m}$. The descendant generator's mean and average variance, are denoted with random variables

$$\mathbf{M}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}, V_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}, \quad j \in \mathbb{Z}_0^+,$$

respectively. Note that $\mathbf{A}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}$ is just a different notation for $\mathbf{A}_{\boldsymbol{\theta}^m}$. Thus,

$$\mathbf{A}_{\boldsymbol{\theta}^m} = \mathbf{A}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}. \tag{6}$$

When a data-point is considered generated by $\mathbf{A}_{\boldsymbol{\theta}^m}$, its distribution parameter, $\boldsymbol{\mu}_{\boldsymbol{\theta}^m}$, needs to be generated a priori. Thus, the generation of $\boldsymbol{\mu}_{\boldsymbol{\theta}^m}$ is independent of $\mathbf{A}_{\boldsymbol{\theta}^m}$. Hence, from Eq. (3),

$$d^2(\mathbf{A}_{\boldsymbol{\theta}^m} - \boldsymbol{\mu}_{\boldsymbol{\theta}^m}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^m}. \tag{7}$$

Likewise, $\mathbf{A}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}$ is independent of the generation of $\boldsymbol{\mu}_{\boldsymbol{\theta}^m}$ and from Eq. (3),

$$d^2(\mathbf{A}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^m}) \overset{a.s.}{=} V_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} + d^2(\mathbf{M}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^m}). \tag{8}$$

Combining Eq. (6), Eq. (7) and Eq. (8) gives rise to the following theorem:

Theorem 2. *(**Mean-Variance Constraint**) The mean and variance of generators descended from $\mathbf{A}_{\boldsymbol{\theta}^m}$ follow the constraint:*

$$V_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} + d^2(\mathbf{M}_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^m}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^m}, \quad \forall j \in \mathbb{Z}_0^+. \tag{9}$$

Theorem 2 implies the average variance of descendant generators, $V_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j}$, is almost-surely less than or equal to their ancestor's, $v_{\boldsymbol{\theta}^m}$. Equality only occurs if the descendant generator's distribution parameters are trivially similar to that of their parents[3]. Thus:

Corollary 1. *Non-trivial[3] descendant generators of $\mathbf{A}_{\boldsymbol{\theta}^m}$, almost-surely have average variances less than $v_{\boldsymbol{\theta}^m}$. i.e.,*

$$V_{\boldsymbol{\theta}^m\boldsymbol{\Theta}^j} \overset{a.s.}{<} v_{\boldsymbol{\theta}^m}, \quad \forall j \in \mathbb{Z}^+. \tag{10}$$

Corollary 1 means generators lower in the hierarchy almost-surely have smaller average variance than their ancestors. Theorem 1 implies distances between data-points almost-surely depend on the average variance of their most recent shared generator. Thus, **the distance between data-points is almost-surely a function of how recently they shared a common generator.**

Using this understanding to interpret Fig. 1, we see that a subset of data-points sharing a common generator, have greater mutual similarity (lower mutual distance) to each other, relative to data-points outside the set. This process can be iterated. Hence, a sub-subset of those data points, sharing an even more recent generator, will have greater mutual similarity to each other, relative to other data-points in the original subset. Thus, the hierarchical generative process creates a corresponding hierarchical pattern of similarities that greatly resembles the natural similarity patterns discussed in Sec. 2.

3. A non-trivial descendant generator is one with a finite difference in either mean or average variance from the parent. Trivial descendants are the opposite. **If the distributions underlying generator creation are continuous, child generators are almost-always non-trivially different from their parents.**

3.2 Distinctive-Shells

In high dimensional hierarchical generative processes, distances between data-points are proxies for statistical dependencies caused by the sharing of generative processes. This statistical constraint can be leveraged to develop unique identifiers for instances of each generator. We term these identifiers, distinctive-shells.

Theorem 3. *Let $\mathbf{A}_{\boldsymbol{\theta}^n}$ denote the data-point outcomes of an n^{th} level generator, that is a non-trivial child of its immediate parent [3,4]. Instances of $\mathbf{A}_{\boldsymbol{\theta}^n}$ are termed α data-points. All other data-points are non-α data-points.*

- *α data-points almost-surely lie on the distinctive-shell s_α :*
$$s_\alpha = \mathcal{S}(\boldsymbol{\mu}_{\boldsymbol{\theta}^n}, \sqrt{v_{\boldsymbol{\theta}^n}}); \tag{11}$$
- *Non-α data-points almost-surely lie outside s_α. Their gap from s_α is given by*
$$\begin{aligned}\mathcal{G}(\mathbf{A}_{\boldsymbol{\theta}^c}, s_\alpha) =& d^2(\mathbf{A}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}) - v_{\boldsymbol{\theta}^n}, \\ \overset{a.s.}{=}& 2(v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n}),\end{aligned} \tag{12}$$
where $\mathcal{G}$ is a function of random variables and $\mathbf{A}_{\boldsymbol{\theta}^c}$ is the non-α data-point's most recent common ancestral generator with α.

Note that Corollary 1 shows $v_{\boldsymbol{\theta}^c}$ is almost-surely greater than $v_{\boldsymbol{\theta}^n}$. Thus, the gap is almost-surely positive for non-α data-points.

Proof. Replacing m by n in Eq. (7) yields

$$d^2(\mathbf{A}_{\boldsymbol{\theta}^n} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}) = v_{\boldsymbol{\theta}^n}. \tag{13}$$

This proves Eq. (11)'s constraint on α data-points. We now focus on proving Eq. (12).

Let $\mathbf{a}_{\overline{\alpha}}$ denote a non-α data-point. Based on rule 3, $\mathbf{a}_{\overline{\alpha}}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}^n}$ can be considered, independently generated by their most recent common generator, whose data-point outcomes are denoted

$$\mathbf{A}_{\boldsymbol{\theta}^c}, \quad c \in \mathbb{Z}_0^+, c < n.$$

Thus, from Eq. (3), the distance of a non-α data-point from $\boldsymbol{\mu}_{\boldsymbol{\theta}^n}$ is:

$$a.s. \quad d^2(\mathbf{a}_{\overline{\alpha}} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}) = v_{\boldsymbol{\theta}^c} + d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}). \tag{14}$$

If we consider $v_{\boldsymbol{\theta}^n}, \boldsymbol{\mu}_{\boldsymbol{\theta}^n}$ to be distribution parameters of some descendant generator of $\mathbf{A}_{\boldsymbol{\theta}^c}$, from Theorem 2:

$$a.s. \qquad v_{\boldsymbol{\theta}^n} + d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}) = v_{\boldsymbol{\theta}^c}. \tag{15}$$

Combining Eq. (14) and Eq. (15), we can see that the distance of $\mathbf{a}_{\overline{\alpha}}$ from shell s_α is

$$a.s. \quad d^2(\mathbf{a}_{\overline{\alpha}} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}) - v_{\boldsymbol{\theta}^n} = 2(v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n}), \tag{16}$$

thus proving Eq. (12).

□

If semantics are symbolic references to individual generators of a hierarchical generative process, Theorem 3 would represent the (to our knowledge) first, explicit mathematical representation of the constraints which make semantic classification possible, a hypothesis that we term shell theory.

This concludes the paper's formal proof. However, before ending the section, we believe it best to provide readers with an intuitive interpretation of shell theory.

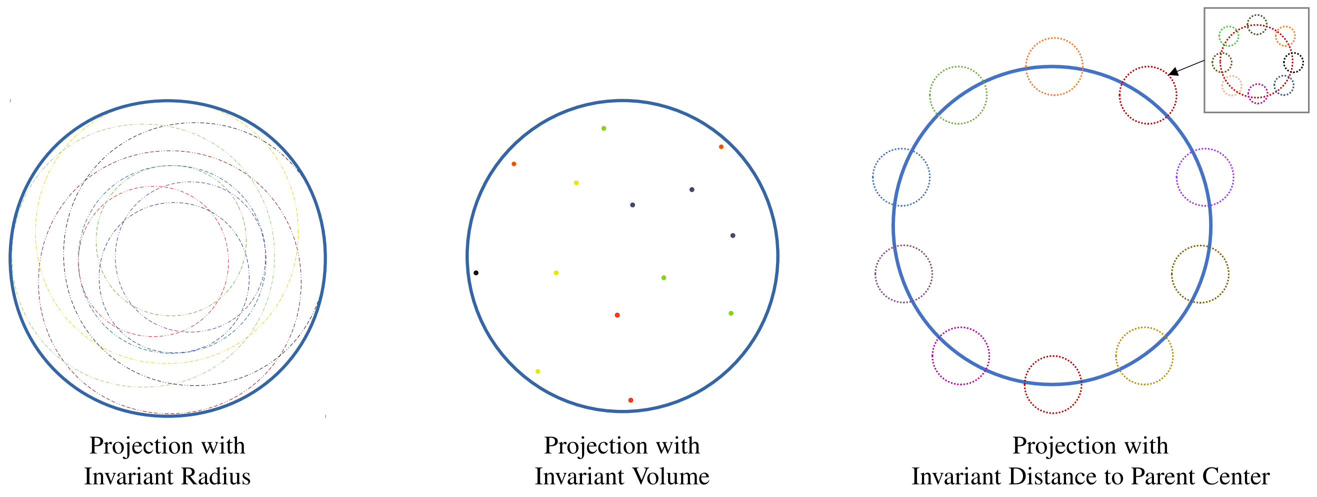


Fig. 2: Two dimensional projections of high dimensional distinctive-shells. Parent shells are in bold, while child shells are dotted. Each projection preserves a different attribute. **Radius invariant projection:** The distinctive-shell of each child generator has a smaller radius than its parent. **Volume invariant projection:** High dimensions cause small changes of radius to induce huge changes of volume. Thus, the distinctive-shells of child generators occupy an infinitesimally small percentage of their parent's distinctive-shell. As a result, independently generated instances of a parent will almost-never enter a child's distinctive-shell. **Projection with invariant distance to center:** For illustrative purposes, the previous two projections have taken liberties with the location of the child distinctive-shells. In reality, child distinctive-shells lie on their parent's distinctive-shell. This allows a parent distinctive-shell to serve as an identifier for almost-all instances of its child generators.

3.3 An Intuitive Derivation of Shell Theory

At the heart of shell theory lies the quirky relation between the radius and volumes of high dimensional hyper-spheres. Consider two k dimensional hyper-spheres, one of radius r and another of radius $r - \Delta r$, where Δr is small but positive. The ratio of their volumes is:

$$\frac{\text{volume of small sphere}}{\text{volume of big sphere}} = \left(\frac{r-\Delta r}{r}\right)^k = x^k, \qquad (17)$$

where x is less than ones. If $k \to \infty$, the above ratio tends to zero. This implies that the small change in radius has induced a huge change in volume. As a result, almost-all of a high dimensional hyper-sphere's volume will lie near its surface (outer-shell).

If we conceptualize a high dimensional sample space as a hyper-sphere, stochastically generated instances will almost-surely lie on the hyper-sphere's outer-shell, as this contains almost-all of the hyper-sphere's volume. It is this phenomenon that gives rise to the shell based constraints that shell theory derives its name.

The quirky relation between volume and radius also explains why distinctive-shells are distinctive. Recall that Corollary 1 and Eq. (11) show child generators have distinctive-shells of smaller radii than their parents; Eq. (17) shows this difference in radius restricts instances of child generators to an infinitesimal fraction of their parent's distinctive-shells. As a result, independently generated instances of a child's parent almost-never coincidentally fall within that child's distinctive-shell, causing the distinctiveness derived in Theorem 3. This is illustrated in Fig. 2 with a series of two dimensional projections of high dimensional distinctive-shells.

In summary: If the objects in our reality were the result of a high dimensional hierarchical generative process, they would represent a very sparse sampling of the huge number of potential objects. The sparsity of this sampling would mean that sets of instances that are related by some common ancestor, would be separated from other instances by wide gulfs of emptiness. This creates naturally occurring boundaries that can be exploited for machine learning. This is the challenge which we take up in the next sections.

4. If $\mathbf{A}_{\boldsymbol{\theta}^n}$ is a trivial child of its parent, its parent's generator is only "trivially-different" from $\mathbf{A}_{\boldsymbol{\theta}^n}$ and its parent's data-points should also be considered α data-points.

4 Normalization and Shells

Normalization is a commonly employed data pre-processor. It is widely accepted to significantly improve machine learning performance. However, the reason for this performance boost remains poorly understood. This section interprets normalization in terms of its impact on the distinctive-shells.

Before continuing, we provide a brief recap of Theorem 3 and its conclusions. $\mathbf{A}_{\boldsymbol{\theta}^n}$ denotes a generator of interest, also known as the α generator. Its outcomes are α data-points, which form a shell, $s_\alpha = \mathcal{S}(\boldsymbol{\mu}_{\boldsymbol{\theta}^n}, \sqrt{v_{\boldsymbol{\theta}^n}})$, centered on $\boldsymbol{\mu}_{\boldsymbol{\theta}^n}$ and with radius $\sqrt{v_{\boldsymbol{\theta}^n}}$.

Let $\mathbf{a}_{\overline{\alpha}}$ denote a non-α test data-point. Its most recent common ancestral generator with α is denoted $\mathbf{A}_{\boldsymbol{\theta}^c}, c < n$.

From Theorem 3, the gap between $\mathbf{a}_{\overline{\alpha}}$ and s_α is

$$\begin{aligned} \mathcal{G}(\mathbf{A}_{\boldsymbol{\theta}^c}, s_\alpha) &\overset{a.s.}{=} v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n} + d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^n}), \\ a.s. \quad &= 2(v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n}). \end{aligned} \qquad (18)$$

where because of Corollary 1, $v_{\boldsymbol{\theta}^c} \overset{a.s.}{>} v_{\boldsymbol{\theta}^n}$, this creates a finite gap that can be used to distinguish $\mathbf{a}_{\overline{\alpha}}$ from α data-points.

We interpret normalization in terms of its impact on the gap between $\mathbf{a}_{\overline{\alpha}}$ from s_α.

4.1 Shell-Normalization

In traditional normalization, the training data's mean is considered a normalization vector. This normalization vector is subtracted from each data-point which is then re-scaled to a unit-vector. This algorithm couples the normalization vector to the training data. However, the nature of the training data, and hence normalization vector, varies with the machine learning problem at hand. We use shell theory to study how such variations impact machine learning performance.

Let $\mathbf{m}$ denote the normalization vector, where,

$$\mathbf{m} \in \{\boldsymbol{\mu}_{\boldsymbol{\theta}^t} | t \in \mathbb{Z}_0^+, \quad t \le n\}.$$

Here, $\boldsymbol{\mu}_{\boldsymbol{\theta}^t}$ denotes the mean of the t^{th} level ancestral generator of α. The symbol, $\widetilde{\ }$, is used to denote post normalization outcomes. Thus, the normalization of a high dimensional random vector $\mathbf{B}$ is denoted

$$\widetilde{\mathbf{B}} = \frac{\mathbf{B} - \mathbf{m}}{\sqrt{d^2(\mathbf{B} - \mathbf{m})}}. \tag{19}$$

$\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^n}$ denotes post-normalized outcomes of the α generator. Its associated distinctive-shell is

$$\widetilde{s}_\alpha = \mathcal{S}(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}, \sqrt{\widetilde{v}_{\boldsymbol{\theta}^n}}),$$

where $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}, \widetilde{v}_{\boldsymbol{\theta}^n}$ are the mean and average variance of $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^n}$.

$\widetilde{\mathbf{a}_{\overline{\alpha}}}$ is the post-normalized non-α data-point. Its gap from the post-normalized shell $\widetilde{s}_\alpha$ shell is $\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha)$.

The effects of different normalization vectors, $\mathbf{m}$, on the gap $\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha)$ is summarized in two rules, whose derivations are presented in Appendix A.

First Normalization Rule: For normalization vector

$$\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^t}, \quad 0 \le t \le c,$$

the gap, $\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha)$, between $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ and shell $\widetilde{s}_\alpha$ increases as t increases, reaching a maximum at $\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^c}$.

Second Rule of Normalization: For normalization vector

$$\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^t}, \quad c \le t \le n,$$

the gap, $\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha)$, between $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ and shell $\widetilde{s}_\alpha$ decreases as t increases. When $\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^n}$, the gap becomes zero.

Together, these two rules suggest the optimal normalization vector for maximizing the gap is $\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^c}$, where $\boldsymbol{\mu}_{\boldsymbol{\theta}^c}$ is the mean of the most recent common ancestor for α and $\mathbf{a}_{\overline{\alpha}}$. This effectively couples the choice of normalization vector to the test data under consideration.

In this framework, changing the normalization vector has an effect analogous to adjusting the focal length on a zoom lens. Consider an example where the α class is Siamese cats. Normalizing with the mean of a very recent ancestor, like cats, "zooms in with the lens", magnifying fine details but losing the big picture. According to the first rule of normalization, the gap between other cat breeds and the Siamese cat's distinctive-shell widens, improving fine grained recognition. However, the second rule of normalization means both other cat breeds and dogs are now at nearly the same distance from the distinctive-shell, reducing the capacity for coarse-grained differentiation.

Similarly, normalizing with the mean of a distant ancestor, like animals, "zooms out the lens", compressing fine details but preserving the big picture. In this case, the gap of other cat breeds from the Siamese cat's distinctive-shell narrows, reducing the capacity for fine grained recognition. However, other cat breeds are now much closer to the distinctive-shell than dogs, thus enhancing coarse differentiation.

We term such shell guided normalization, shell-normalization.

Implications: Shell-normalization suggests a rule-of-thumb, in which **the normalization vector should be the mean of the test-data**. This differs from traditional normalization, which uses the training data's mean as the normalization vector.

From this perspective, traditional normalization is a special case that is only optimal if the training data's mean approximates the test data's mean. The success of traditional normalization is caused by the fact that normalization is typically employed for multi-class problems that coincidentally satisfy the above condition. This in turn suggests that the normalization process needs to be modified if it is to accommodate other machine learning problems.

4.2 Normalization and Noise

We have previously shown how the normalization vector can be chosen to maximize the gap. Interestingly, this procedure also has a noise cancellation property. We provide some preliminary analysis on this phenomenon in Appendix B. However, the topic requires a more complete analysis.

5 Semantic Manifolds

It is often hypothesized that instances of each semantic lie on a representative manifold. Shell theory suggests such semantic manifolds are actually distinctive-shells, thus providing us with the first, explicit parametrization of such manifolds. Explicit parametrization allows complex manifold discovery problems to be solved using simple shell fitting algorithms. The result is a fast and effective tool for machine learning.

5.1 Implementation

Shell theory assumes objects are represented by long attribute vectors. In practice, such attribute vectors are difficult to obtain. Thus, we use image features as proxies for attribute vectors, with each object being represented by the ResNet-50 feature [34] feature associated with its image. Where possible, the shell normalization pre-processor of Sec. 4.1 is applied to the features.

Given a set of features of a specific semantic, the semantic's distinctive-shell can be estimated using Algorithm 1. The next sections employ these distinctive-shells for: one-class learning; multi-class classification; and anomaly detection.

Algorithm 1: Shell-Fit

```
  Input: Features $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_N\}$
1 $\boldsymbol{\mu}'_\alpha = average(\mathbf{F})$ ;
2 $r'_\alpha = average(\{\|\mathbf{f}_i - \boldsymbol{\mu}'_\alpha\|\}_{i=1,\dots,N})$ ;
  Output: $s'_\alpha = \mathcal{S}(\boldsymbol{\mu}'_\alpha, r'_\alpha)$
```

6 One-Class Learning

One-class learning tries to identify objects of a specific class amongst all objects, by primarily learning from a training set containing only the objects of that class. Objects of the interest class are termed positive instances, other objects are termed negative instances.

As negative instances are not available in the training phase, it is difficult to define boundaries between positive and negative instances. Indeed, it is tempting to consider one-class learning an ill-posed problem. Yet, this cannot be true; humans are remarkably adept at one-class learning.

Shell theory provides a mathematically principled solution to this conundrum. Distinctive-shells can be learned from positive instances, with Theorem 3 guaranteeing the learned distinctive-shell will encapsulate almost-all positive instances and exclude almost-all negative ones. Finally, shell-normalization provides a data pre-processing framework for enhancing the effectiveness of distinctive-shells. The result is a one-class learning algorithm that is simple, fast and very effective.

6.1 Experimental Setup

The following data-sets are used for experimental evaluation:

- Fashion-MNIST [37] consists of simple black and white images which are rasterized into feature vectors. This data-set is likely not high dimensional, with Principal Component Analysis showing that on average, 50% of the variation in each class can be explained with just four dimensions. It is included in the evaluation because it is a common data-set for evaluating one-class learning and provides an interesting test case for the robustness of shell fitting when its assumptions are violated;
- STL-10 [38], is another widely used data-set. It consists of 10 commonly occurring classes, airplane, bird, cat, etc;
- Internet STL-10 is a data-set we created to simulate learning in the "wild". This data-set has the same 10 classes as STL-10. However, training images are crawled from the internet. Testing is performed on the original STL-10 data-set;
- MIT-Places-5 [39] is composed of the first five scene classes of the MIT-Places [39]. These are: abbey, airport terminal, alley, arch and amusement park. This data differs greatly from the object oriented classes, cat, human, car, etc, used to train deep-learned features [33], [34]. It is included to help assess the generalizability of shell fitting algorithms;
- Assira [40] consists of two easily confused classes [41], cats and dogs. This allows us to asses shell fittings reliability on fine-grained recognition.

When evaluating one-class learning algorithms on data-sets with multiple classes, we treat one class at a time as the positive class and the remainder as the negative class. Half the instances of the positive class are used as training data and the remaining instances (both positive and negative) are used as test data.

A one-class learner is expected to rank the test data based on their similarity to the positive class. The reliability of the rankings are using AUROC (Area Under Receiver Operating Characteristic curve). Each algorithm's average AUROC is tabulated in Table 1, with computational times presented in Table 3.

6.2 Discussion

We divide algorithms according to their inputs. This creates three groups:

Symbol	Algorithm Inputs
-I	raw <u>images</u>
-F	image <u>features</u>
-FN	<u>features normalized</u> with the mean of test data

Table 1 shows group -I has noticeably poorer performance than the other groups. This is surprising as group -I is expected to provide the best results.

Algorithms in group -I use deep-learning to jointly learn a semantic manifold and feature representations from positive instances. Such algorithms require significant parameter tuning. While we tuned the algorithms, parameter choices may still be sub-optimal.

Parameter settings aside, these results may indicate a more fundamental problem. The second rule of normalization in Sec. 4.1 shows that setting normalization vector as the positive class mean ($\mathbf{m} = \boldsymbol{\mu}_{\boldsymbol{\theta}^n}$) will reduce the gap between negative instance and the distinctive-shell to be zero, i.e. both positive and negative instances map onto the same distinctive-shell. This creates a degeneracy in which a manifold that fits the positive instances, also fits the negative instances. As algorithms in group -I have very large solution spaces, it is possible they are encountering this (or other) degeneracies.

Shell fitting is a manifold discovery approach which uses the last layer of a pre-trained network as a feature representation, this results in a fixed feature representation. Shell fitting and other algorithms that employ fixed feature representations, are grouped under -F. Table 1 shows the performance of algorithms in group -F are notably better performance than those of group -I. This may be due to their avoidance of the degeneracies.

Shell normalization is a pre-processor that shows how features can be adapted for improved performance. Algorithms whose input features have undergone shell normalization are grouped under -FN. Table 1 shows shell-normalization boost the performance of all algorithms, with many scores approaching perfection. To our knowledge, this is the first time such a phenomenon is reported.

If we focus on shell fitting's performance, we can see from Table 1 that shell fitting's accuracy is very similar to that of prior manifold discovery techniques. This suggests semantic manifolds are indeed distinctive-shells. As shell fitting is much simpler than manifold discovery, this perspective opens the possibility for significant gains in algorithmic efficiency, an issue that the following section discusses in detail.

Finally, Table 2 shows the result of repeating the STL-10 evaluation with different features. It is noticeable that the performance of all algorithms vary significantly with feature choice; however, the previously observed trends remain observable, with shell normalization pre-processor providing significant performance improvement and shell fitting having an accuracy that matches traditional manifold discovery.

6.3 Efficient Manifolds

In classic machine learning, semantic manifolds are complex entities which require large numbers of parameters to model faithfully. This has led to manifold representations that employ as many parameters as the availability of training data permits.

In such representations, parameters often increase linearly with quantity of training data. As a result, the time required to evaluate a test point also increases with the number of parameters, giving rise to a test time complexity of $O(N)$, where N is the number of training data-points. Such representations are also slow to train, as the pairwise comparison of data-points needed for manifold fitting tend to result in training time complexities of $O(N^2)$.

In shell theory, semantic manifolds are distinctive-shells. This suggests that increasing manifold complexity will only provide limited performance gains, as Theorem 3 has already guaranteed the distinctive-shell's distinctiveness. By decoupling model complexity from the availability of training data, a shell fitting approach provides a test time complexity of $O(1)$. In addition, distinctive-shells can be estimated without pairwise comparisons, reducing training complexity to $O(N)$. Table 3 shows the impact of these efficiency gains, while Table 1 shows that the gains do not come at the expense of performance.

6.4 Limitations

The previous results serve as a strong validation of shell theory. However we must make four caveats:

First, shell theory assumes data is high dimensional. This causes approximation noise on simple data-sets like Fashion MNIST. While naively applying shell theory yields respectable performance, additional research is needed to bridge the gap between low and high dimensions.

Second, the paper only provides a preliminary noise analysis. Better understanding of noise's impact on shell distinctiveness and normalization is critical for enhancing machine learning performance.

Third, the effectiveness of the distinctive-shell depends on feature quality. However, shell theory does not explain how feature quality can be ensured. This represents yet another gap in our knowledge that requires further investigation.

Fourth, the paper assumes each semantic is a reference to a single generator. In practice, a semantic may consist of a number of distantly related generators. For example, the word apple is used to refer to a fruit and an electronic device. A less obvious example would be the use of the term church to refer to both the outside and inside of christian religious buildings. In such cases, provided the number of unrelated generators encapsulated by the semantic is small, the semantics would still possess a "semi-distinctive-shell". This is because an instance that is unusually close to a semantic's generator mean would also be unusually close to the overall semantic mean. This paper does not consider this cases, which is more about how generative processes are represented in language, rather than about the generative processes themselves. However, in practical implementation, it is an important caveat readers should keep in mind.

	Average AUROC for each data-set				
	Fashion -MNIST	STL-10	Internet STL-10	MIT-Places	ASSIRA
Deep A.D. [42]-I	**0.935**	**0.730**	**0.717**	**0.722**	**0.888**
DSEBM [43]-I	0.884	0.571	0.560	0.613	0.516
DAGMM [44]-I	0.518	0.554	0.517	0.530	0.485
AD-GAN [45]-I	0.884	0.602	0.555	0.499	0.534
OC-VAE [46]-I	0.605	0.624	0.610	0.597	0.559
Shell [Ours]-F	**0.893**	0.874	**0.827**	0.793	0.815
OCSVM [47]-F	0.892	0.799	0.557	0.765	0.824
Mahalanobis [48]-F	**0.893**	**0.940**	0.819	**0.845**	**0.924**
Shell [Ours]-FN	0.921	0.992	**0.987**	0.981	**0.999**
OCSVM [Ours]-FN	**0.932**	**0.993**	**0.987**	**0.982**	**0.999**
Mahalanobis [Ours]-FN	0.892	0.988	0.926	0.936	0.987

TABLE 1: AUROC scores of one-class learners. The best algorithm in each group is highlighted in bold. Observe that shell learning is competitive with far more sophisticated manifold fitting techniques. In addition, our theory's suggested normalization pre-processor (denoted by the suffix -FN), gives a huge boost to one-class learning performance. These results provide empirical validation for our theory of distinctive-shells.

	Average AUROC on STL-10, using different features					
	ResNet -50 [34]	ResNet -50* [34]	VGG -16 [33]	ResNet -101 [34]	Conv. Auto	KAZE Feat. [49]
Shell-F	0.874	0.629	0.865	0.868	0.607	**0.530**
OCSVM-F	0.799	**0.634**	0.881	0.883	0.624	0.529
Mahalanobis-F	**0.940**	0.573	**0.909**	**0.938**	**0.668**	0.516
Shell-FN	0.992	0.885	0.992	0.995	0.775	0.728
OCSVM-FN	**0.993**	**0.897**	**0.993**	**0.996**	**0.803**	**0.737**
Mahalanobis-FN	0.988	0.695	0.987	0.991	0.756	0.536

TABLE 2: AUROC scores of one-class learners using different features. Evaluations are performed on STL-10. Features tested are: deep-learned features trained on imageNet (ResNet-50, ReseNet-101, VGG-16); features trained on tiny imageNet (ResNet-50*); features trained on STL-10 with a convolution autoencoder (Conv. Auto); hand-crafted features (KAZE Feat.). Shell normalization and shell fitting remain consistently effective across all features.

7 One-Class to Multi-Class

Multi-class classification is the classic problem of machine learning and is almost always formulated in terms of adversarially defined class boundaries. Shell learning suggests an intriguing alternative.

Theorem 3 predicts that a semantics' distinctive-shell encapsulates only instances of itself, while the previous section shows that each semantics' distinctive-shell can be estimated using instances of itself. This creates the possibility of a multi-class classifier that learns by incrementally fusing independently trained distinctive-shells that have been learned in a one-class manner.

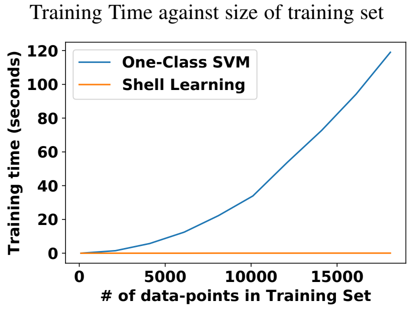
7.1 Experiment Setup and Discussion

We divide the algorithms used in the experiments into two groups: The first group consists of traditional multi-class classification algorithms like linear SVM [50], [51], kernel SVM [52] and PSDML (a multi-class manifold learner) [53], all of which are discriminative classifiers that learn partition boundaries between classes. These algorithms are denoted with the suffix -MC. The second group consists of multi-class classification algorithms created by fusing independently trained one-class learners. For these algorithms, we apply a traditional normalization pre-processor before conducting one-class learning. Note that, as discussed in Sec. 4.1, in the context of multi-class classification, shell-normalization is identical to traditional normalization. Such algorithms are denoted with the suffix -FN.

Shell theory suggests multiple distinctive-shells can be fused into a multi-class classifier by simply assigning test instances to their closest shell. It is unclear what the equivalent procedures for the other one-class learners; hence, we heuristically modify each algorithm to maximize performance. For one-class SVM [47], we divide scores by the number of training instances. This (mysteriously?) provides a great improvement to results. The scheme is denoted OCSVM*. For Mahalanobis [48], we find assigning each instance to its highest scoring class works best.

Algorithms are evaluated on the same five datasets used in the previous section, with mean accuracy and mean average precision of each algorithm tabulated in Table 4.

	Computational Time						
Algo	Shell Learning	One-Class SVM	Mahala. [48]	DSEBM [43]	Deep A.D. [42]	DAGMM [44]	AD-GAN [45]
Training Time	0.00301 Sec	0.578 Sec	0.615 Sec	165 Sec	191 Sec	53.713 Sec	2558 Sec
Testing Time	0.0999 Sec	3.524 Sec	0.953 Sec	1.33 Sec	983 Sec	1.203 Sec	2539 Sec
Device	CPU	CPU	CPU	GPU	GPU	GPU	GPU



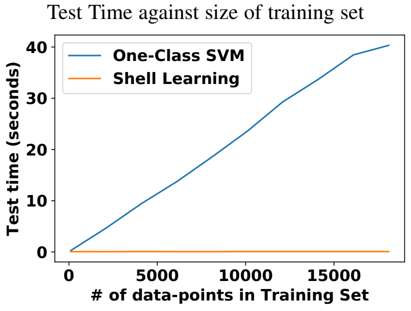


TABLE 3: **Top:** Training and test time of one-class learning algorithms. Timing is measured on STL-10, with each class divided into 650 training and 650 testing data-points. CPU refers to a single thread of the AMD Ryzen 7 2700 processor; GPU refers to a Nvidia 1080Ti graphics card. We observe that shell learning is an order of magnitude faster than most of its counterparts. **Bottom:** A plot of training and test time as training set size increases, with shell learning being compared to the classic one-class SVM algorithm. Shell learning is based on a new interpretation of semantic manifolds which leads to lower computational complexity. Section 6.3 discusses this issue in more detail.

Table 4 shows distinctive-shells are surprisingly good at multi-class classification. Compared to other one-class learning techniques, shell based multi-class classification shows a consistent 3-4% improvement in both accuracy and mean average precision, with some performance gains as high as 10-20%.

Indeed, for most data-sets, shell based classification is almost comparable to traditional multi-class classifiers. The one exception is Fashion-MNIST, which (as discussed in Sec. 6) may be inappropriate for our model.

These results act as additional empirical validation of shell theory; however, they also provide a new perspective to machine learning that we explore in the next section.

7.2 From Machines to Humans and Back Again

Shell theory may be the answer to one of the great riddles of our time: What is the mechanics of human learning? When teaching children a semantic class, we typically show them multiple exemplars of the class. There is notably less emphasis on explicitly showing what examples do not belong to the class, something which children are expected to infer for themselves. This mechanics differs greatly from the adversarial boundary training employed in classic machine learning but is remarkably similar to distinctive-shell fitting.

Accepting the hypothesis that humans learn through distinctive-shells may answer one question but it creates another puzzle. Learning distinctive-shells is less optimal than learning explicit adversarial boundaries. Why would humans evolve to learn or teach in a sub-optimal fashion? We believe this is because distinctive-shell learning has advantages which make it uniquely suited to unstructured environments:

- **Incremental learning:** As demonstrated in the experiments of the previous subsection, it is easy to incorporate new semantic concepts by learning their distinctive-shells. Thus, shell based learning can commence before all training classes are collected in one place. This is necessary for humans but is also important for machine learning, where such problems are termed life-long learning or never-ending learning [54], [55], [56];
- **Robustness to failure:** Shell theory allows each class to be trained independently. This allows the overall system to survive training inconsistencies that affect only a few classes, providing the robustness that humans need to function in the “wild”. Such robustness is also critical to machine learning problems that use noisy internet data;
- **Accommodate overlapping class membership:** Class memberships are often not mutually exclusive. When overlaps are known a priori, they can be handled within the classic partition boundary discovery framework. However, mistakenly considering overlapping classes to be non-overlapping can disastrously impact such classifiers. This problem is notably less acute in shell based learning, which need not assume mutually exclusive class membership.

In summary, shell theory may help explain how humans effortlessly accommodate challenging learning environments. By applying these lessons to machine learning, we may be able to make a generational leap in artificial intelligence capability.

8 Anomaly Detection

The term anomaly detection is used to refer to two different machine learning problems. Both problems share a common goal of differentiating instances of an interest class (normality) from other instances (anomalies). However, the problems make different assumptions regarding the information available.

The first problem, which is sometimes termed supervised anomaly detection, assumes the availability of training data that consists only of normal instances. In this paper, we refer to this problem as one-class learning, which is discussed in Sec. 6.

The second problem, which is sometimes termed unsupervised anomaly detection, assumes a clean set of normal data is not available for training. The task is to identify the anomalies in a data-set of mixed, normal and anomalous instances, without

	Mean accuracy on each data-set				
	Fashion -MNIST	STL-10	Internet STL-10	MIT-Places	ASSIRA
Shell-FN	0.617	**0.945**	**0.878**	**0.903**	**0.985**
OCSVM*-FN	0.579	0.917	0.862	0.865	0.984
Mahalanobis-FN	**0.677**	0.922	0.453	0.770	0.979
SVM(linear)-MC	**0.810**	0.944	0.880	0.926	**0.988**
SVM(RBF-kernel)-MC	0.809	**0.969**	**0.915**	**0.939**	0.987
PSDML-MC	0.680	0.961	0.858	0.920	0.987
	Mean average precision on each data-set				
	Fashion -MNIST	STL-10	Internet STL-10	MIT-Places	ASSIRA
Shell-FN	**0.730**	**0.991**	**0.970**	**0.981**	**0.999**
OCSVM*-FN	0.652	0.958	0.914	0.946	**0.999**
Mahalanobis-FN	0.598	0.923	0.768	0.827	0.991
SVM(linear)-MC	0.818	**0.990**	0.954	0.968	**0.999**
SVM(RBF-kernel)-MC	**0.823**	0.988	**0.955**	**0.973**	**0.999**
PSDML-MC	0.566	0.977	0.914	0.916	**0.999**

TABLE 4: Mean accuracy and mean average precision on multi-class classification problems. Suffix -FN indicates algorithms based on fusing independently trained one-class learners; suffix -MC denotes traditional multi-class classifiers. Shell learners are conceptually very different from traditional multi-class classifiers but attain a similar level of accuracy. This may represent the emergence of a new framework for machine learning.

any prior reference to clean normal data. The second problem is notably more difficult than the first. In this paper, the term anomaly detection refers exclusively to the second problem.

Traditional anomaly detection is based on the principle that normal instances will be related by a consistency that is absent from the anomalies. Thus, it is assumed that fitting a manifold to a data-set that consists of a mix of normal and anomalous instances will cause the anomalies (if present) to stand out as instances that will not fit the manifold.

Shell theory suggests the above approach may be flawed, since, for any given set of data-points, there will be a most recent common ancestor on whose distinctive-shell (manifold), all data-points would lie.

Shell theory also suggests an alternative framework, in which anomaly detection can be formulated in terms of robust distinctive-shell fitting. This can be implemented through iterative robust least squares, with an algorithm that alternates between estimating a distinctive-shell from hypothesized inliers, and re-estimating inliers based on the hypothesized distinctive-shell. Following the conventions of robust least squares, the threshold between inliers and outliers is set to a multiple of the estimated standard-deviation of inlier points from the shell, with the standard-deviation estimated as being 1.42 times the median absolute deviation. The process is summarized in Algorithm 2.

As in the one-class learning of the previous section, performance can be enhanced through shell-normalization; however, it is hard to know a priori what the normalization vector should be. In this section, we use the mean of Flickr11k [57] as the normalization vector. This implicitly assumes Flickr11k approximates a random sample of all images in the world, thus preventing the anomaly detector from being biased to any specific class. If we can estimate the mean of the most recent common ancestor of both the normal and anomalous data, we can use it as the normalization vector. This would specialize the distinctive-shell to the data-set at hand and is achieved through the re-normalization option of Algorithm 2. The impact of re-normalization is illustrated in Fig. 3.

8.1 Experiment Setup and Discussion

Anomaly detectors are evaluated on the data-sets: STL-10, MIT-Places-5 and Assira Dog vs Cat. For each data-set, we treat one class at a time as the normal class, with the other classes being anomalies. Anomalous data is mixed with the normal data. The anomaly detection algorithm is tasked with assigning each data instance an anomaly score which will separates that the normal and anomalous data. i.e. normal data and anomalous data instances should receive very different scores. The reliability of the anomaly score is measured by its receiver operating characteristics (ROC). The ROC for different percentages of anomalies is plotted in Fig. 4.

When interpreting results, readers should bear in mind that anomaly detection is only a meaningful problem if the normal class makes up a clear plurality of the data-set (i.e. is the largest subset by a significant margin). Further, the impact of anomaly percentages on normal class plurality depends on the number of classes per data-set. These are: 10, 5 and 2, for STL-10, MIT-Places-5 and Assira Dog vs Cat respectively.

When anomalies come from many different classes, the normal class may retain a clear plurality even if the percentage of anomalies is high. Thus, in data-sets like STL-10 and MIT-Places-5, the anomaly detection problem remains well defined, even if anomaly percentages exceed 50%.

Assira is an exceptional data-set because it comprises of only two classes. As a result, all anomalies come from a single class. This causes the normal class to loose plurality rapidly as the percentages of anomalies increases. Thus, when measured in terms of normal class plurality, 20% anomalies in Assira is equivalent to 50% anomalies in MIT-Places-5. Thus, when evaluated on Assira, all anomaly detection algorithms show sharp performance declines after the 20% anomaly threshold.

Focusing on shell based anomaly detection, we see that it maintains high ROC scores even when anomalies make up a large percentage of the data-set. This is very challenging, as attested to by the performance of one-class SVM, the classic baseline evaluating anomaly detection. Shell based anomaly detection is also simpler and faster than one-class SVM; this suggests that shell based techniques may serve as useful baselines in future anomaly detection evaluations. Qualitative evaluation is provided in Fig. 5.

9 General Discussion

Thus far, we have used shell theory to address specific machine learning problems. However, shell theory can also provide general insights, some of which are useful, others just quirky.

9.1 Statistical Maximum Distance of $\sqrt{2}$ for Unit-Normalized Data

The geometric maximum distance between two unit-vectors is 2. Our theory suggests the existence of a previously unknown, statistical maximum distance, that is substantially lower than the geometric maximum. This is illustrated in Fig. 6.

The proof is as follows. For unit-vector-normalized data,

$$\begin{aligned} &d^2(\mathbf{A}-\mathbf{0}) \overset{a.s.}{=} v_{\mathbf{A}} + d^2(\boldsymbol{\mu}_{\mathbf{A}}-\mathbf{0}) = 1, \\ &a.s. \quad \Rightarrow \quad v_{\mathbf{A}} \leq 1. \end{aligned} \tag{20}$$

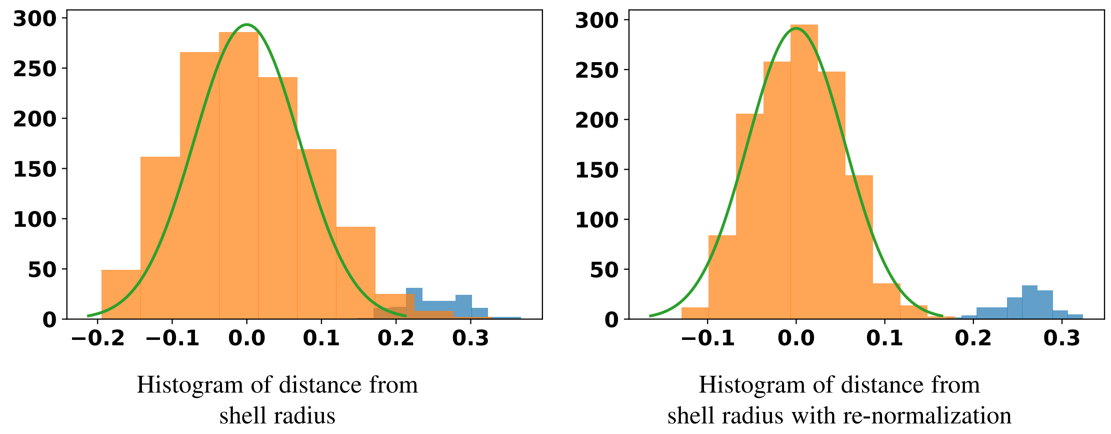


Fig. 3: Histogram of distances from inlier class mean; inliers are denoted with orange and anomalies with blue. **Left:** Interpreting the histogram as distance from distinctive-shell re-centers it. Inlier distances can be modeled as a zero mean, Gaussian perturbation of the shell. Anomalies can be identified as data points that cannot be explained with the Gaussian. **Right:** Iteratively re-normalizing the data based on hypothesized anomalies widens the gap between inliers and anomalies, enabling finer anomaly detection results.

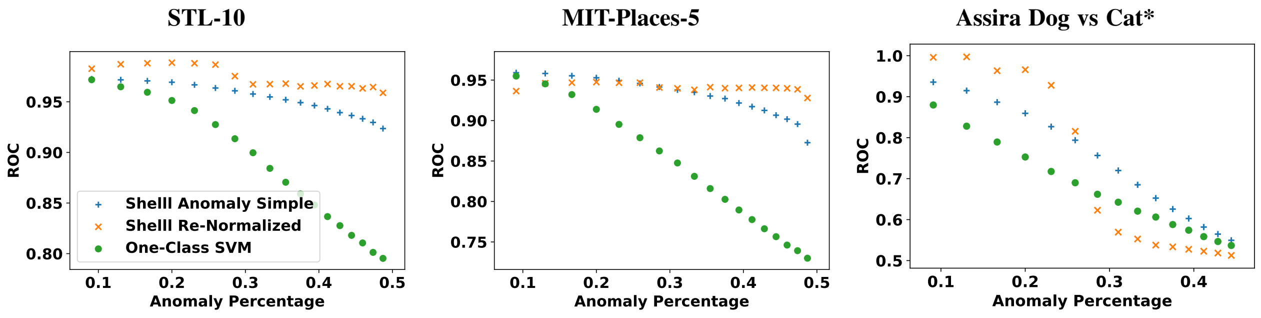


Fig. 4: ROC scores for anomaly detection algorithms. Shell fitting provides significant improvement over classic one-class SVM. *Assira is a two class data-set, in which the problem of anomaly detection is no longer well defined when anomalies exceed 20%. This is elaborated in Sec. 8.1.

Algorithm 2: Anomaly Detection

```
Input:
Features: $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n\}$;
Anomaly Threshold: thres = 2;
Normalization Vector: $\mathbf{m}$
Re-Normalize = True/ False
1  $\mathbf{F}_{inlier} = \mathbf{F}$;
2  $\mathbf{F}_{outlier} = \varnothing$;
3  for $i = 1$ to # iterations do
4      $\widetilde{\mathbf{F}}$ = Shell-Normalization($\mathbf{F}_{inlier}, \mathbf{m}$);
5      $s'_\alpha = \mathcal{S}(\boldsymbol{\mu}'_\alpha, r'_\alpha)$ = Shell-Fit($\widetilde{\mathbf{F}}$);
6      $e = \{e_i | e_i = \|\widetilde{\mathbf{f}}_i - \boldsymbol{\mu}'_\alpha\| - r'_\alpha\}$;
7      $e_{mad}$ = median(absolute($e$));
8      $\mathbf{F}_{inlier} = \{\mathbf{f}_i | e_i < 1.42 * thres * e_{mad}\}$;
9      $\mathbf{F}_{outlier} = \mathbf{F}^c_{inlier}$;
10     if Re-Normalize then
11         $\mathbf{m} = average(\mathbf{F}_{outlier})$;
Output: $\mathbf{F}_{inlier}, \mathbf{F}_{outlier}, e$
```

Hence, from Corollary 1, the average variance of some generator $\mathbf{A}_{\boldsymbol{\theta}^m}$ is almost-surely less than or equal to one. i.e.

$$a.s. \quad v_{\boldsymbol{\theta}^m} \leq 1. \tag{21}$$

From Corollary 1, the squared distance between any two data-points is almost-surely two times the average variance. of their most recent shared generator. Hence, from Eq. (21), the distance between two data-points $\mathbf{a}_1, \mathbf{a}_2$ is

$$a.s. \quad \sqrt{d^2(\mathbf{a}_1 - \mathbf{a}_2)} \leq \sqrt{2}. \tag{22}$$

9.2 Deep Learners are Easily Fooled

This paradox was raised by Nguyen *et al.* [58], who showed that deep-learned classifiers are robust in practice but vulnerable to small deliberate perturbations, which can fool the classifier into making ridiculous conclusions.

This paradox becomes comprehensible if we assume deep learners are representing semantic classes with distinctive-shells that are adjacent to each other. The adjacency of shells will not affect classification because the almost-surely convergence in high dimensions, means instances almost never jump the small gap between shells. However, such small gaps are vulnerable to deliberate perturbations, giving rise to the effect reported by the authors.

9.3 "Contrast-Loss"

"Contrast-loss" refers to proof by Argarwal *et al.* [6] which suggests machine learning in high dimensions is intrinsically

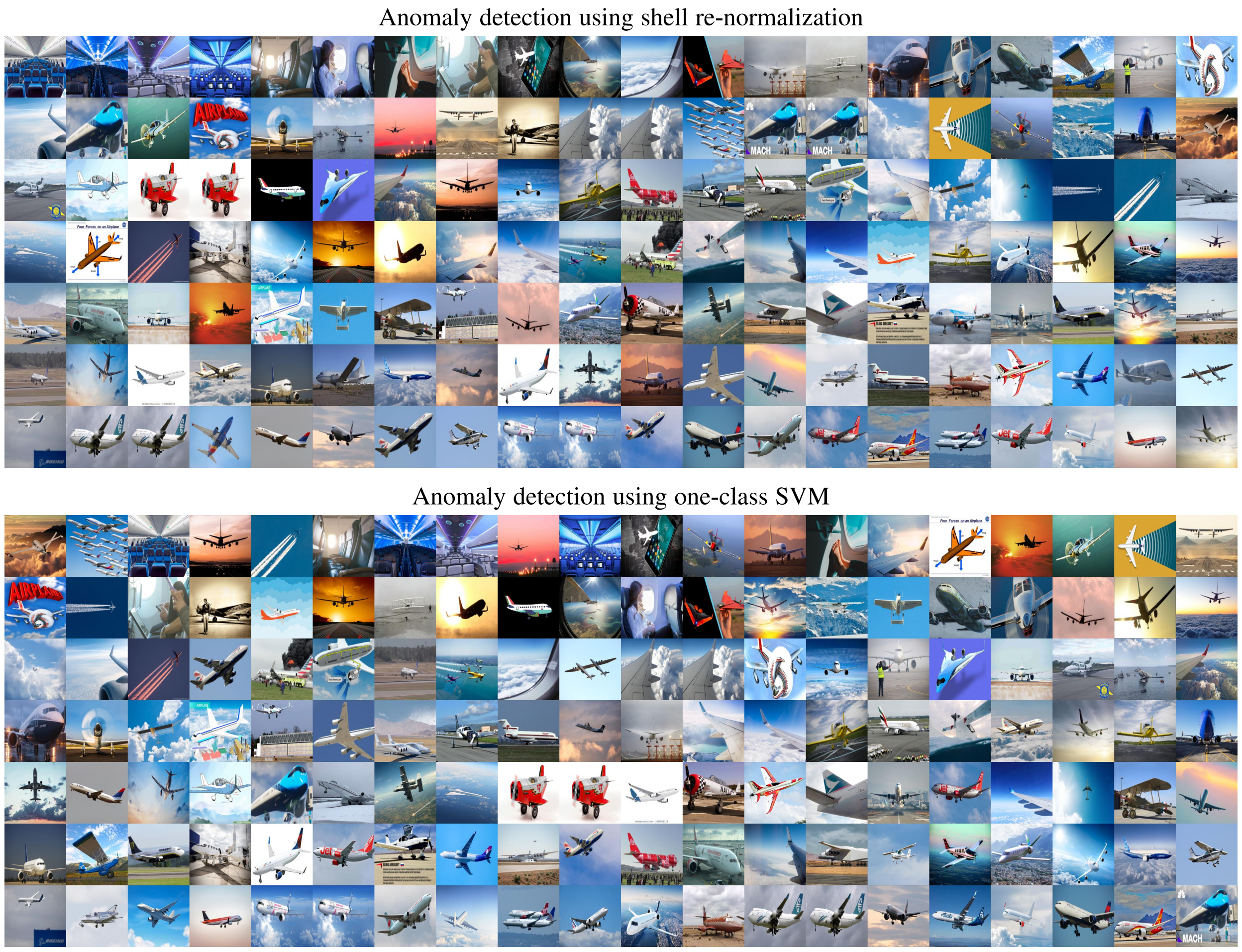

Anomaly detection using shell re-normalization
Anomaly detection using one-class SVM

Fig. 5: Anomaly based ranking of airplane images crawled from the internet based; anomaly scores decrease from left to right, top to bottom. Shell based anomaly detection provides rankings which are noticeably more consistent than that of one-class SVM.

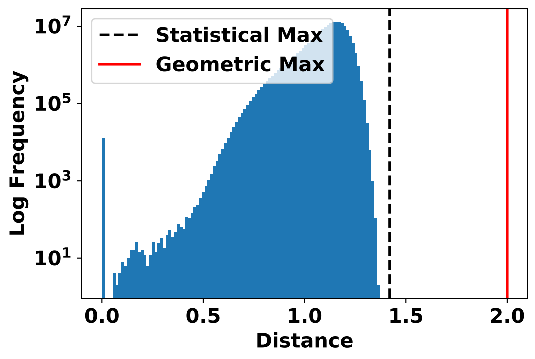


Fig. 6: Log histogram of pairwise distances between unit-vectored ResNet features descriptions [34] of STL-10 images [38]. The distances seem to abide by our predicted statistical maximum, which is much lower than the geometric maximum.

impossible because the distance between almost-all instances converge to a constant. This proof is widely accepted by the machine learning community [8]. However, it is inconsistent with empirical evidence which shows deep-learned features are both effective and high dimensional.

The hierarchy of dependencies proposed by shell theory resolves this paradox. In shell theory, the constraints used to prove "contrast-loss" are used to show that distances between high dimensional instances reflect shared dependencies (Theorem 1). This makes "contrast-loss" a special case of shell theory, which occurs in the unlikely event of all instances being statistically independent outcomes of a single generator.

10 Conclusion

This paper proposes a distance based technique that makes high dimensional statistical models possible. The technique is used to develop shell theory, a stochastic model of the generative processes that underlie natural data.

While the majority of the paper is dedicated to the mathematical formulation and empirical evaluation of shell theory, we believe its primary contribution lies not in theory itself but in the research philosophy the theory represents: Understand machine learning requires an understanding of the underlying patterns we seek to learn. As such, we invite readers to consider shell theory as a proof of concept, rather than a theory of everything; may it be the first step in a journey of a thousand miles. [5]

5. Code is available at: https://www.kind-of-works.com/

Acknowledgments

This research is supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant. The authors also gratefully acknowledge Lam Hoi Sze's contribution towards making the paper possible.

References

[1] A. Holster, *The Death of Science: A Companion Study to Martín López Corredoira's The Twilight of the Scientific Age*. Universal-Publishers, 2016.

[2] R. Wells Jr, "Complex manifolds and mathematical physics," *Bulletin of the American Mathematical Society*, vol. 1, no. 2, pp. 296–336, 1979.

[3] A. N. Gorban and I. Y. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2118, p. 20170237, 2018.

[4] S. Wenling, S. Kihyuk, A. Diogo *et al.*, "The extraordinary link between deep neural networks and the nature of the universe," *MIT Technology Review*, 2016.

[5] W.-Y. Lin, S. Liu, J.-H. Lai, and Y. Matsushita, "Dimensionality's blessing: Clustering images by underlying distribution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5784–5793.

[6] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[7] P. M. Domingos, "A few useful things to know about machine learning." *Commun. acm*, vol. 55, no. 10, pp. 78–87, 2012.

[8] https://en.wikipedia.org/wiki/Curse_of_dimensionality, accessed: 2019-11-15.

[9] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Reverse nearest neighbors in unsupervised distance-based outlier detection," *IEEE transactions on knowledge and data engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.

[10] N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic, "The role of hubness in clustering high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 739–751, 2014.

[11] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[12] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," *arXiv preprint arXiv:1703.04933*, 2017.

[13] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," *arXiv preprint arXiv:1710.05468*, 2017.

[14] M. Hutson, "Ai researchers allege that machine learning is alchemy," *Science*, vol. 360, no. 6388, p. 861, 2018.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 601–608.

[16] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 1605–1614.

[17] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros, "Unsupervised discovery of visual object class hierarchies," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[19] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[20] F. Dellaert, "The expectation maximization algorithm," Georgia Institute of Technology, Tech. Rep., 2002.

[21] G. E. Box and G. C. Tiao, "Multiparameter problems from a bayesian point of view," *The Annals of Mathematical Statistics*, vol. 36, no. 5, pp. 1468–1482, 1965.

[22] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[23] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[24] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," *CVPR*, 2021.

[25] G. Xuan, W. Zhang, and P. Chai, "Em algorithms of gaussian mixture model and hidden markov model," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 1. IEEE, 2001, pp. 145–148.

[26] J. Zhu, S. C. Hoi, and M. R. Lyu, "Nonrigid shape recovery by gaussian process regression," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 1319–1326.

[27] W.-Y. Lin, M.-M. Cheng, S. Zheng, J. Lu, and N. Crook, "Robust non-parametric data fitting for correspondence modeling," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2376–2383.

[28] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[29] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 305–314.

[30] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," *arXiv preprint arXiv:2009.09687*, 2020.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[32] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[36] C. Li, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," *Advances in neural information processing systems*, vol. 30, pp. 4088–4098, 2017.

[37] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[38] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.

[39] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 487–495.

[40] J. Elson, J. J. Douceur, J. Howell, and J. Saul, "Asirra: A captcha that exploits interest-aligned manual image categorization," in *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.

[41] O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman, "The truth about cats and dogs," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1427–1434.

[42] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018, pp. 9758–9769.

[43] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," *arXiv preprint arXiv:1605.07717*, 2016.

[44] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[45] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft, "Anomaly detection with generative adversarial networks," *https://openreview.net/forum?id=S1EfylZ0Z*, 2018.

[46] H. Khalid and S. S. Woo, "Oc-fakedect: Classifying deepfakes using one-class variational autoencoder," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 656–657.

[47] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class svm for learning in image retrieval." in *Proceedings of IEEE International Conference on Image Processing (ICIP)*. Citeseer, 2001, pp. 34–37.

[48] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in

Advances in Neural Information Processing Systems, 2018, pp. 7167–7177.

[49] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “Kaze features,” in *European Conference on Computer Vision*. Springer, 2012, pp. 214–227.

[50] M. A. Hearst, “Support vector machines,” *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18–28, Jul. 1998. [Online]. Available: https://doi.org/10.1109/5254.708428

[51] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[52] J. Platt *et al.*, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[53] P. Zhu, L. Zhang, W. Zuo, and D. Zhang, “From point to set: Extend the learning of distance metrics,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2664–2671.

[54] N. Kardan and K. O. Stanley, “Fitted learning: Models with awareness of their limits,” *arXiv preprint arXiv:1609.02226*, 2016.

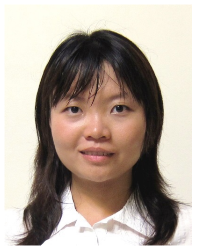
[55] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, 2019.

[56] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel *et al.*, “Never-ending learning,” *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[57] Y.-H. Kuo, H.-T. Lin, W.-H. Cheng, Y.-H. Yang, and W. H. Hsu, “Unsupervised auxiliary visual words discovery for large-scale image object retrieval,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 905–912.

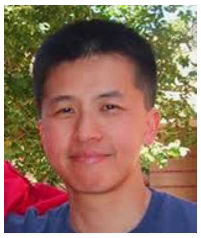
[58] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 427–436.

Wen-Yan Lin received his PhD degree from the National University of Singapore in 2011. He is currently an Assistant Professor with the Singapore Management University.



Siying Liu received the B. Eng and M. Eng degrees in Electrical Engineering from the National University of Singapore in 2006 and 2009, respectively. She obtained the PhD degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2016, supervised by Prof. Minh N. Do. She is currently a Research Scientist at the Institute for Infocomm Research, Singapore.

Changhao Ren received B.Eng. in electrical engineering (2019) and is pursuing M.S. in computer science at National University of Singapore. He is currently working as a research assistant at the Singapore University of Technology and Design.



Ngai-Man Cheung is an Associate Professor with Singapore University of Technology and Design (SUTD). He receives his Ph.D. degree in Electrical Engineering from University of Southern California (USC), Los Angeles, CA. His Ph.D. research focused on image and video coding, and the work was supported in part by NASA-JPL. He was a postdoctoral researcher with the Image, Video and Multimedia Systems group at Stanford University, Stanford, CA. He has also held research positions with IBM T. J. Watson Research Center, Hong Kong University of Science and Technology (HKUST), and Mitsubishi Electric Research Labs (MERL).

His research has resulted in 11 U.S. patents granted with several pending. Two of his inventions have been licensed to companies. One of his research results has led to a SUTD spinoff on AI for wound care. His research has also been featured in the National Artificial Intelligence Strategy. He has received several research recognitions, including recently the Best Paper Finalist at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2019, the Finalist of Super AI Leader (SAIL) Award at the World AI Conference (WAIC) 2019, and the Excellence in Research Award from SUTD. His research interests are Image and Signal Processing, Computer Vision and AI.

Hongdong Li is a Chief Investigator of the Australia ARC Centre of Excellence for Robotic Vision, professor in Computer Vision with the Australian National University. He joined the ANU from 2004 first as a postdoctoral fellow, then a senior research fellow until 2008. He was a Senior Scientist of NICTA in 2008-2010. He was a Visiting Professor with the Robotics Institutes, Carnegie Mellon University, doing a sabbatical in 2017-2018. His research interests include 3D computer vision, pattern recognition and machine learning, autonomous driving and mathematical optimization. He is an Associate Editor for IEEE Transactions on PAMI, and IVC, and served as Area Chair for recent years' CVPR, ICCV and ECCV conferences. He was the winner of the 2012 CVPR Best Paper Award, the 2017 Marr Prize (Honorable Mention), a finalist for the CVPR 2020 best paper award, ICPR Best student paper award, ICIP Best student paper award, DSTO Fundamental Contribution to Image Processing Prize at DICTA 2014, Best algorithm award in CVPR NRSFM Challenge 2017, and a Best Practice Paper (honourable mention) award at WACV 2020. He is a Co Program Chair for ACCV 2018 and ACCV 2022, Co-Publication Chair for IEEE ICCV 2019. His research is funded by Australia Research Council, CSIRO, Microsoft Research etc.

Yasuyuki Matsushita received his B.S., M.S. and Ph.D. degrees in EECS from the University of Tokyo in 1998, 2000, and 2003, respectively. From April 2003 to March 2015, he was with Visual Computing group at Microsoft Research Asia. In April 2015, he joined Osaka University as a professor. His research area includes computer vision, machine learning and optimization. He is/was an Editor-in-Chief for International Journal of Computer Vision and on editorial board of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), The Visual Computer journal, IPSJ Transactions on Computer Vision Applications (CVA), and Encyclopedia of Computer Vision. He served/is serving as a Program Co-Chair of PSIVT 2010, 3DIMPVT 2011, ACCV 2012, ICCV 2017, and a General Co-Chair for ACCV 2014 and ICCV 2021. He is a senior member of IEEE.

Appendix A Proof of Normalization Rules 1 and 2 in Sec. 4.1

Let $\mathbf{A}_{\boldsymbol{\theta}^t}$ is the generator whose mean is the normalization vector $\boldsymbol{\mu}_{\boldsymbol{\theta}^t}$. The normalization rules 1 and 2 derive from the following corollary:

Corollary 2. *After normalization with $\boldsymbol{\mu}_{\boldsymbol{\theta}^t}$, the distance of a non-$\alpha$ point, $\widetilde{\mathbf{a}_{\overline{\alpha}}}$, from shell $\widetilde{s}_\alpha$ is almost-surely:*

$$\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha) \overset{a.s.}{=} \begin{cases} \dfrac{2(v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n})}{v_{\boldsymbol{\theta}^t}} \overset{a.s.}{=} \dfrac{\mathcal{G}(\mathbf{A}_{\boldsymbol{\theta}^c}, s_\alpha)}{v_{\boldsymbol{\theta}^t}}, & \text{if } 0 \le t \le c \le n; \\ \dfrac{2(v_{\boldsymbol{\theta}^t} - v_{\boldsymbol{\theta}^n})}{v_{\boldsymbol{\theta}^t}} \overset{a.s.}{=} \dfrac{\mathcal{G}(\mathbf{A}_{\boldsymbol{\theta}^t}, s_\alpha)}{v_{\boldsymbol{\theta}^t}}, & \text{if } 0 \le c \le t \le n. \end{cases} \tag{23}$$

For unit-vector data, $1 \ge v_{\mathbf{A}}$. Further, from Corollary 1,

$$1 \ge v_{\mathbf{A}} > v_{\boldsymbol{\theta}^t|t=1} > v_{\boldsymbol{\theta}^t|t=2} > \ldots > v_{\boldsymbol{\theta}^n}.$$

Thus, the first case in Eq. (23) shows that as t increases from zero, the distance of $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ from shell $\widetilde{s}_\alpha$ increases, peaking at $t = c$.

If t increases past c, the distance of $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ from shell $\widetilde{s}_\alpha$, is defined by the second case in Eq. (23) and decreases to zero at $t = n$.

These findings are summarized by normalization rules one and two in Sec. 4.1. Their proof is as follows.

Proving the first case of Corollary 2: This proof requires the following Lemma.

Lemma 1. *Let $\mathbf{B}$ be the data-point outcome of some generator. $\mathbf{C}$ is the data-point outcome of an ancestral generator of $\mathbf{B}$ that is independent of $\mathbf{B}$. $\mathbf{z}$ is an instantiated vector that is independent of both $\mathbf{B}$ and $\mathbf{C}$. The squared distance of $\mathbf{B}$ and $\mathbf{C}$ from $\mathbf{z}$ is almost-surely equal to a shared constant, we denote as z. i.e.*

$$a.s. \quad d^2(\mathbf{B} - \mathbf{z}) \overset{a.s.}{=} d^2(\mathbf{C} - \mathbf{z}) \overset{a.s.}{=} z \tag{24}$$

Proof. From Eq. (3), we know that the random variable $d^2(\mathbf{C} - \mathbf{z})$ is almost-surely a constant. We denote that constant as z, such that

$$d^2(\mathbf{C} - \mathbf{z}) \overset{a.s.}{=} z.$$

As an instance of $\mathbf{B}$ can be considered an instance of $\mathbf{C}$ generated independently of $\mathbf{z}$, the above relation implies:

$$a.s. \quad d^2(\mathbf{B} - \mathbf{z}) \overset{a.s.}{=} z.$$

□

For the first case of Eq. (23), $\mathbf{A}_{\boldsymbol{\theta}^t}$ is ancestral to $\mathbf{A}_{\boldsymbol{\theta}^c}$ and $\mathbf{A}_{\boldsymbol{\theta}^n}$, from Eq. (3), and Lemma 1,

$$\begin{aligned} &d^2(\mathbf{A}_{\boldsymbol{\theta}^t} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^t}, \\ a.s. \quad &d^2(\mathbf{A}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^t}, \quad d^2(\mathbf{A}_{\boldsymbol{\theta}^n} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^t}. \end{aligned} \tag{25}$$

Thus the impact of this normalization on $\mathbf{A}_{\boldsymbol{\theta}^c}, \mathbf{A}_{\boldsymbol{\theta}^n}$ is to translate their instances by $\boldsymbol{\mu}_{\boldsymbol{\theta}^t}$, then scale the result by $\frac{1}{\sqrt{v_{\boldsymbol{\theta}^t}}}$. i.e.

$$\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^n} \overset{a.s.}{=} \frac{\mathbf{A}_{\boldsymbol{\theta}^n} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}}{\sqrt{v_{\boldsymbol{\theta}^t}}}, \quad \widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c} \overset{a.s.}{=} \frac{\mathbf{A}_{\boldsymbol{\theta}^c} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}}{\sqrt{v_{\boldsymbol{\theta}^t}}}. \tag{26}$$

As $\mathbf{a}_{\overline{\alpha}}$ is an instance of $\mathbf{A}_{\boldsymbol{\theta}^c}$,

$$a.s. \quad \widetilde{\mathbf{a}_{\overline{\alpha}}} = \frac{\mathbf{a}_{\overline{\alpha}} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}}{\sqrt{v_{\boldsymbol{\theta}^t}}}. \tag{27}$$

From Eq. (25), we know $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^n}$ data-points lie on shell

$$\widetilde{s}_\alpha = \mathcal{S}\left(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n} = \frac{\boldsymbol{\mu}_{\boldsymbol{\theta}^n} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}}{\sqrt{v_{\boldsymbol{\theta}^t}}}, \sqrt{\widetilde{v}_{\boldsymbol{\theta}^n}} = \sqrt{\frac{v_{\boldsymbol{\theta}^n}}{v_{\boldsymbol{\theta}^t}}}\right). \tag{28}$$

and $\widetilde{\mathbf{a}_{\overline{\alpha}}}$'s distance from $\widetilde{s}_\alpha$ is

$$a.s. \quad \mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha) = \frac{\mathcal{G}(\mathbf{A}_{\boldsymbol{\theta}^c}, s_\alpha)}{v_{\boldsymbol{\theta}^t}} \overset{a.s.}{=} \frac{2(v_{\boldsymbol{\theta}^c} - v_{\boldsymbol{\theta}^n})}{v_{\boldsymbol{\theta}^t}}. \tag{29}$$

This proves the first case of Corollary 2.

Proving the second case of Corollary 2: Here, $\mathbf{A}_{\boldsymbol{\theta}^t}$ is a descendant of $\mathbf{A}_{\boldsymbol{\theta}^c}$ but ancestor of $\mathbf{A}_{\boldsymbol{\theta}^n}$. To prove this case, we need the following Lemma.

Lemma 2. *Let $\mathbf{A}_{\boldsymbol{\theta}^t}$ be some intermediate generator. It's mean, one of it's descendant generator's mean and an independent point $\mathbf{z} \in \mathbb{R}^k$ almost- surely form a right-angled triangle:*

$$d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \mathbf{z}) \overset{a.s.}{=} d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^t} - \mathbf{z}) + d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}). \tag{30}$$

Proof. From Eq. (3) and Theorem 2,

$$\begin{aligned} &d^2(\mathbf{A}_{\boldsymbol{\theta}^t} - \mathbf{z}) \\ &\overset{a.s.}{=} V_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} + d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \mathbf{z}) \\ &\overset{a.s.}{=} v_{\boldsymbol{\theta}^t} - d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}) + d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \mathbf{z}) \end{aligned} \tag{31}$$

From Eq. (3),

$$d^2(\mathbf{A}_{\boldsymbol{\theta}^t} - \mathbf{z}) \overset{a.s.}{=} v_{\boldsymbol{\theta}^t} + d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^t} - \mathbf{z}). \tag{32}$$

Combining Eq. (31) and Eq. (32), yields

$$d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \mathbf{z}) \overset{a.s.}{=} d^2(\boldsymbol{\mu}_{\boldsymbol{\theta}^t} - \mathbf{z}) + d^2(\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j} - \boldsymbol{\mu}_{\boldsymbol{\theta}^t}).$$

□

Note that after normalization with $\boldsymbol{\mu}_{\boldsymbol{\theta}^t}$, from Eq. (3), we have the following constraints:

$$\begin{aligned} &\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^t} = \mathbf{0}; \\ &\widetilde{v}_{\boldsymbol{\theta}^n} + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}) = 1; \quad \text{because } d^2(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^n} - \mathbf{0}) \overset{a.s.}{=} 1 \\ &\widetilde{v}_{\boldsymbol{\theta}^c} + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) = 1; \quad \text{because } d^2(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c} - \mathbf{0}) \overset{a.s.}{=} 1 \end{aligned} \tag{33}$$

Replacing $\mathbf{A}_{\boldsymbol{\theta}^t}, \mathbf{z}$ in Lemma 2, with $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^t}, \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}$ respectively:

$$d^2(\widetilde{\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j}} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) \overset{a.s.}{=} d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^t} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) + d^2(\widetilde{\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j}} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^t}). \tag{34}$$

As $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}$ is an instance of $\widetilde{\mathbf{M}_{\boldsymbol{\theta}^t\boldsymbol{\Theta}^j}}$, from Eq. (33) and Eq. (34),

$$\begin{aligned} a.s. \quad d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) &= d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^t} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^t}) \\ &= d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}). \end{aligned} \tag{35}$$

As $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ is an instance of $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}$ that is independent of $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}$ ($\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}$ is their most recent common ancestor), combining Eq. (3), Eq. (33), Eq. (35), the distance of $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ from shell center $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}$ is almost-surely

$$\begin{aligned} d^2(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}) &\overset{a.s.}{=} \widetilde{v}_{\boldsymbol{\theta}^c} + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c} - \widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}) \\ &= \widetilde{v}_{\boldsymbol{\theta}^c} + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^c}) + d^2(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{\theta}^n}) \\ &= 2 - \widetilde{v}_{\boldsymbol{\theta}^n}. \end{aligned} \tag{36}$$

Combing the above results with Eq. (28), the distance of $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ to the shell $\widetilde{s}_\alpha$ is:

$$a.s. \quad \mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha) = 2(1 - \widetilde{v}_{\boldsymbol{\theta}^n}) = \frac{2(v_{\boldsymbol{\theta}^t} - v_{\boldsymbol{\theta}^n})}{v_{\boldsymbol{\theta}^t}}. \tag{37}$$

This proves the second case of Corollary 2.

APPENDIX B
IMPACT OF NORMALIZATION ON NOISE: PROOFS FOR SEC. 4.2

In machine learning, deviations from expected results are modeled as noise. The most commonly investigated noise model is sensor-noise. In this scenario, the data-points are not faithful representations of reality and the deviation is modeled as noise. In this paper, we assume data is captured accurately and sensor noise is negligible.

Instead, we consider shell learning to be a framework for predicting distances between data-points In this context, noise refers to deviations from these theoretical distance predictions. Noisy distances are distinguished by the notation d^2_N. Thus, the noisy distance between random vector $\mathbf{Z}$ and an independent vector $\mathbf{c}$ is denoted

$$d^2_N(\mathbf{Z}-\mathbf{c}) \overset{a.s.}{=} d^2(\mathbf{Z}-\mathbf{c}) + \mathcal{N}. \tag{38}$$

with the random variable $\mathcal{N}$ used to denote additive noise. $\mathcal{N}$ is assumed to be independent of the hierarchical generative process and have zero mean.

Note that as we do not consider sensor noise, there is no noisy version of $\mathbf{Z}$. However, since normalization involves dividing data-points by their magnitude, it is impacted by noise in the distance estimate. As defined in Sec. 4.2, the outcome of noisy normalization is denoted by appending $\tilde{'}$ to instances or random vectors.

The impact of noise on distinctive-shells must be studied with respect to the gap (distances of points from distinctive-shells). In theory, there are four possible cases: Large noise with small gap; large noise with large gap; small noise with large gap; and small noise with small gap. Of these, only the last case is important. In the first two cases where noise is large, it would be more reasonable to consider an alternative model. In the third case, the impact of noise would be too small to interfere with the gap and is unlikely to affect machine learning performance. This, leaves the last case, which we consider below.

We show that in this case, if normalization is performed optimally, it will cancel out the first order noise terms. This is beneficial to fine grained recognition based on small differences in the gap. More formally:

Corollary 3. *Let $\widetilde{s}_\alpha$ denote a post-normalized distinctive-shell; $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ is a similarly normalized non-α point; and $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}$ the normalized, most recent common ancestor between $\mathbf{a}_{\overline{\alpha}}$ and α. The post-normalization gap between $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ and $\widetilde{s}_\alpha$ is $\mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha)$.*

If the gap between $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ and $\widetilde{s}_\alpha$ is small and normalization is optimal, the first order Taylor approximation of the gap in the noisy case is

$$a.s. \quad \mathcal{G}(\widetilde{\mathbf{A}}'_{\boldsymbol{\theta}^c}, \widetilde{s}'_\alpha) \overset{a.s.}{\approx} \mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \widetilde{s}_\alpha), \tag{39}$$

with all first order terms being zero.

Thus, in this case, when noise is small, we can approximate the noisy gap with the noise free gap.

Below, we show how Corollary 3 is derived.

Proof of Corollary 3: Let $\mathbf{B}$ be the outcome of some generator that is independent of $\mathbf{m}$. We use m to denote the expected squared distance of $\mathbf{B}$ to normalization vector $\mathbf{m}$, i.e.

$$d^2(\mathbf{B}-\mathbf{m}) \overset{a.s.}{=} m.$$

Th corresponding noisy distance is:

$$d^2_N(\mathbf{B}-\mathbf{m}) \overset{a.s.}{=} d^2(\mathbf{B}-\mathbf{m}) + \mathcal{N}. \tag{40}$$

where, as mentioned earlier, random variable $\mathcal{N}$ denotes noise.

The normalization of $\mathbf{B}$ in the presence of noise is:

$$\widetilde{\mathbf{B}}' = \frac{\mathbf{B}-\mathbf{m}}{\sqrt{d^2_N(\mathbf{B}-\mathbf{m})}}. \tag{41}$$

In the noiseless case, the normalized instances, $\widetilde{\mathbf{B}}$, are encompassed by the distinctive-shell, $\widetilde{s}_\mathbf{B}$, where

$$\widetilde{s}_\mathbf{B} = \mathcal{S}\left(\widetilde{\boldsymbol{\mu}}_\mathbf{B} = \frac{\boldsymbol{\mu}_\mathbf{B}-\mathbf{m}}{\sqrt{m}}, \sqrt{\widetilde{v}_\mathbf{B}} = \sqrt{\frac{v_\mathbf{B}}{m}}\right). \tag{42}$$

We use $\tilde{s}'_\mathbf{B}$ to denote the distinctive-shell after noisy normalization. $\tilde{s}'_\mathbf{B}$'s center is the mean of $\widetilde{\mathbf{B}}'$, denoted $\widetilde{\boldsymbol{\mu}}'_\mathbf{B}$. $\tilde{s}'_\mathbf{B}$'s radius is the expected distance of $\widetilde{\mathbf{B}}'$ from $\widetilde{\boldsymbol{\mu}}'_\mathbf{B}$. i.e.

$$\tilde{s}'_\mathbf{B} = \mathcal{S}\left(\widetilde{\boldsymbol{\mu}}'_\mathbf{B}, \quad \tilde{r}'_\mathbf{B} = \mathbb{E}\left(\sqrt{d^2_N(\widetilde{\mathbf{B}}' - \widetilde{\boldsymbol{\mu}}'_\mathbf{B})}\right)\right). \tag{43}$$

where $\tilde{r}'_\mathbf{B}$ is used to denote the radius of shell $\tilde{s}'_\mathbf{B}$.

Next, we show that the conditions stated in corollary 3 mean that $\tilde{s}'_\mathbf{B}$ is approximately equal to $\widetilde{s}_\mathbf{B}$ and use this result to show the noisy distance of a test point, $\widetilde{\mathbf{a}_{\overline{\alpha}}}'$ from $\tilde{s}'_\mathbf{B}$ is approximately the same as the distance of $\widetilde{\mathbf{a}_{\overline{\alpha}}}$ from $\widetilde{s}_\mathbf{B}$.

In the following Lemma 3, we prove that when noise is small, the centers of $\tilde{s}'_\mathbf{B}$ and $\widetilde{s}_\mathbf{B}$ are approximately the same:

Lemma 3. *The first order Taylor series approximation of $\widetilde{\boldsymbol{\mu}}'_\mathbf{B}$ is almost-surely its noise free counterpart $\widetilde{\boldsymbol{\mu}}_\mathbf{B}$, i.e.*

$$a.s. \quad \widetilde{\boldsymbol{\mu}}'_\mathbf{B} \approx \widetilde{\boldsymbol{\mu}}_\mathbf{B}, \tag{44}$$

with all first order noise terms being zero.

Thus, if noise is small, we can approximate $\widetilde{\boldsymbol{\mu}}'_\mathbf{B}$ with $\widetilde{\boldsymbol{\mu}}_\mathbf{B}$.

Proof. As $\mathbb{E}(\mathcal{N}) = 0$ and $\mathcal{N}$ is independent of $\mathbf{B}$,

$$\begin{aligned}\widetilde{\boldsymbol{\mu}}'_\mathbf{B} &= \mathbb{E}(\widetilde{\mathbf{B}}')\\ &= \mathbb{E}\left(\frac{\mathbf{B}-\mathbf{m}}{\sqrt{d^2_N(\mathbf{B}-\mathbf{m})}}\right)\\ &\overset{a.s.}{=} \mathbb{E}\left(\frac{\mathbf{B}-\mathbf{m}}{\sqrt{m+\mathcal{N})}}\right)\\ &\approx \mathbb{E}\left(\frac{\mathbf{B}-\mathbf{m}}{\sqrt{m}}\left(1-\frac{\mathcal{N}}{2m}\right)\right)\\ &= \mathbb{E}\left(\frac{\mathbf{B}-\mathbf{m}}{\sqrt{m}}\right)\\ &= \mathbb{E}(\widetilde{\mathbf{B}}) = \widetilde{\boldsymbol{\mu}}_\mathbf{B}\end{aligned} \tag{45}$$

□

To derive an expression for the radius of $\tilde{s}'_\mathbf{B}$, we take a slightly circuitous route and first develop Lemma 4 to relate noisy normalized instances of ancestral generators of $\mathbf{B}$ to $\widetilde{\boldsymbol{\mu}}'_\mathbf{B}$. We reuse this Lemma later to simplify the subsequent proofs.

Lemma 4. *Let $\mathbf{c}$ be an instantiated data-point of generator $\mathbf{C}$, which is also $\mathbf{c}$'s most recent common ancestral generator with $\mathbf{B}$. $c_\mathbf{B}$ is used to denote the value that $d^2(\mathbf{C}-\boldsymbol{\mu}_\mathbf{B})$ converges to, i.e.*

$$d^2(\mathbf{C}-\boldsymbol{\mu}_\mathbf{B}) \overset{a.s.}{=} c_\mathbf{B}. \tag{46}$$

The above is the theoretically predicted distance of $\mathbf{c}$ *from the center of shell* $s_{\mathbf{B}}$*. We will prove that the corresponding theoretically predicted, normalized distance of* $\widetilde{\mathbf{c}}$ *from the center of the normalized shell* $\tilde{s}_{\mathbf{B}}$ *is:*

$$d^2(\widetilde{\mathbf{C}} - \widetilde{\boldsymbol{\mu}}_{\mathbf{B}}) \overset{a.s.}{=} \frac{c_{\mathbf{B}}}{m}. \tag{47}$$

Further, we show that given the conditions of Corollary 3, the first order Taylor approximation of the squared distance of $\widetilde{\mathbf{c}}'$ *from to* $\widetilde{\boldsymbol{\mu}}'_{\mathbf{B}}$ *is:*

$$a.s. \quad d^2(\widetilde{\mathbf{c}}' - \widetilde{\boldsymbol{\mu}}'_{\mathbf{B}}) \approx d^2(\widetilde{\mathbf{c}} - \widetilde{\boldsymbol{\mu}}_{\mathbf{B}}) = \frac{c_{\mathbf{B}}}{m}. \tag{48}$$

This means that, when noise is small and the conditions in Corollary 3 are true, the noisy distance of $\mathbf{c}$ *from distinctive-shell centers is approximately the noiseless distance.*

Proof. The conditions in Corollary 3 focus on the case where normalization is optimal and the gap is small. From Sec. 4.2 we know that a small gap implies that $\boldsymbol{\mu}_{\mathbf{C}} \to \boldsymbol{\mu}_{\mathbf{B}}$. While optimal normalization implies $\mathbf{m} = \boldsymbol{\mu}_{\mathbf{C}} \to \boldsymbol{\mu}_{\mathbf{B}}$ and that $\mathbf{m}$ is independent of $\mathbf{C}$.

As $\mathbf{m}$ is independent of $\mathbf{C}$, from Lemma 1,

$$d^2(\mathbf{C} - \mathbf{m}) \overset{a.s.}{=} m. \tag{49}$$

Fusing Eq. (49) and the normalization definition in Eq. (19), we have

$$a.s. \quad d^2(\widetilde{\mathbf{C}} - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}) \overset{a.s.}{=} \frac{d^2(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}})}{m} \overset{a.s.}{=} \frac{c_{\mathbf{B}}}{m},$$

thus proving Eq. (47).

The noisy distance of $\mathbf{c}$ from distinctive shell center $\boldsymbol{\mu}_{\mathbf{B}}$ and normalization vector $\mathbf{m}$ are:

$$d_N^2(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}}) = c_{\mathbf{B}} + \mathcal{N}_1, \quad d_N^2(\mathbf{C} - \mathbf{m}) = m + \mathcal{N}_2, \tag{50}$$

respectively. The corresponding normalized distance from normalized shell center is given by

$$d_N^2(\widetilde{\mathbf{C}}' - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}').$$

Using Eq. (50) with Lemma 3, the first order Taylor approximation of $d_N^2(\widetilde{\mathbf{C}}' - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}')$ is:

$$\begin{aligned}
&d_N^2(\widetilde{\mathbf{C}}' - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}') \\
\overset{a.s.}{\approx}\, &d_N^2(\widetilde{\mathbf{C}}' - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}) \\
\overset{a.s.}{=}\, &d_N^2\left(\frac{\mathbf{C} - \mathbf{m}}{\sqrt{m + \mathcal{N}_1}} - \frac{\boldsymbol{\mu}_{\mathbf{B}} - \mathbf{m}}{\sqrt{m}}\right) \\
\overset{a.s.}{=}\, &\frac{d_N^2\left(\mathbf{C} - \mathbf{m} - (\boldsymbol{\mu}_{\mathbf{B}} - \mathbf{m})\sqrt{\frac{m+\mathcal{N}_1}{m}}\right)}{m + \mathcal{N}_1} \\
\overset{a.s.}{\approx}\, &\frac{d_N^2\left(\mathbf{C} - \mathbf{m} - (\boldsymbol{\mu}_{\mathbf{B}} - \mathbf{m})(1 + \frac{\mathcal{N}_1}{2m})\right)}{m + \mathcal{N}_1} \\
\overset{a.s.}{=}\, &\frac{d_N^2\left(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}} - (\boldsymbol{\mu}_{\mathbf{B}} - \mathbf{m})(\frac{\mathcal{N}_1}{2m})\right)}{m + \mathcal{N}} \\
\overset{a.s.}{=}\, &\frac{d_N^2(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}}) + \left(\frac{\mathcal{N}_1}{m}\right)^2 k_1 + \left(\frac{\mathcal{N}_1}{m}\right)^4 k_2}{m + \mathcal{N}_1} \\
\overset{a.s.}{\approx}\, &\frac{d_N^2(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}})}{m + \mathcal{N}_1} \\
\overset{a.s.}{=}\, &\frac{c_{\mathbf{B}} + \mathcal{N}_2}{m + \mathcal{N}_1},
\end{aligned} \tag{51}$$

where k_1 and k_2 are coefficients for the higher order terms.

Note that as $\mathbf{m} \to \boldsymbol{\mu}_{\mathbf{B}}$, $d_N^2(\mathbf{C} - \mathbf{m}) \to d_N^2(\mathbf{C} - \boldsymbol{\mu}_{\mathbf{B}})$ and thus $\mathcal{N}_2 \to \mathcal{N}_1$. This allows the above expression to be further simplified to

$$\begin{aligned}
&d_N^2(\widetilde{\mathbf{C}}' - \widetilde{\boldsymbol{\mu}_{\mathbf{B}}}') \\
\overset{a.s.}{\approx}\, &\frac{c_{\mathbf{B}} + \mathcal{N}_2}{m + \mathcal{N}_1} \\
\approx\, &\left(\frac{c_{\mathbf{B}}}{m} + \frac{\mathcal{N}_2}{m}\right)\left(1 - \frac{\mathcal{N}_1}{m}\right) \\
\approx\, &\frac{c_{\mathbf{B}}}{m} + \frac{\mathcal{N}_1}{m}\left(\frac{\mathcal{N}_2}{\mathcal{N}_1} - \frac{c_{\mathbf{B}}}{m}\right) \\
\approx\, &\frac{c_{\mathbf{B}}}{m}.
\end{aligned} \tag{52}$$

Combining Eq. (52) with Eq. (47), we get:

$$a.s. \quad d^2(\widetilde{\mathbf{c}}' - \widetilde{\boldsymbol{\mu}}'_{\mathbf{B}}) \approx d^2(\widetilde{\mathbf{c}} - \widetilde{\boldsymbol{\mu}}_{\mathbf{B}}) = \frac{c_{\mathbf{B}}}{m},$$

thus proving Eq. (48).

□

Note that we can set $\mathbf{C} = \mathbf{B}$ in Lemma 4. This means

$$d_N^2(\widetilde{\mathbf{B}}' - \widetilde{\mu}'_{\mathbf{B}}) = d^2(\widetilde{\mathbf{B}} - \widetilde{\boldsymbol{\mu}}_{\mathbf{B}}). \tag{53}$$

Thus, the radius of distinctive shell $\tilde{s}'_{\mathbf{B}}$ is:

$$\tilde{r}'_{\mathbf{B}} \approx \mathbb{E}\left(\sqrt{d^2(\widetilde{\mathbf{B}} - \widetilde{\boldsymbol{\mu}}_{\mathbf{B}})}\right) \overset{a.s.}{=} \sqrt{\widetilde{v_{\mathbf{B}}}}. \tag{54}$$

Combining this result with Lemma 3, the first-order approximation Taylor approximation of $\tilde{s}'_{\mathbf{B}}$ is

$$a.s. \quad \tilde{s}'_{\mathbf{B}} \approx \tilde{s}_{\mathbf{B}}. \tag{55}$$

Given this expression for $\tilde{s}'_{\mathbf{B}}$, the gap between a point $\widetilde{\mathbf{c}}'$ in Lemma 4 and distinctive-shell $\tilde{s}'_{\mathbf{B}}$ is:

$$a.s. \quad d^2(\widetilde{\mathbf{c}}' - \widetilde{\boldsymbol{\mu}}'_{\mathbf{B}}) - (\tilde{r}'_{\mathbf{B}})^2 \approx \frac{c_{\mathbf{B}}}{m} - \widetilde{v_{\mathbf{B}}}. \tag{56}$$

This can also be expressed in terms of random vectors as:

$$\mathcal{G}(\widetilde{\mathbf{C}}', \tilde{s}'_{\mathbf{B}}) \overset{a.s.}{\approx} \frac{c_{\mathbf{B}}}{m} - \widetilde{v_{\mathbf{B}}}. \tag{57}$$

In the noiseless case, Eq. (42) and Eq. (47) mean the gap from $\widetilde{\mathbf{c}}$ from $\tilde{s}_{\mathbf{B}}$ is

$$\mathcal{G}(\widetilde{\mathbf{C}}, \tilde{s}_{\mathbf{B}}) \overset{a.s.}{=} \frac{c_{\mathbf{B}}}{m} - \widetilde{v_{\mathbf{B}}}. \tag{58}$$

From the above two equations, the first order Taylor approximation of $\mathcal{G}(\widetilde{C}', \tilde{s}'_{\mathbf{B}})$ is:

$$\mathcal{G}(\widetilde{\mathbf{C}}', \tilde{s}'_{\mathbf{B}}) \approx \mathcal{G}(\widetilde{\mathbf{C}}, \tilde{s}_{\mathbf{B}}). \tag{59}$$

Let the α generator be denoted $\mathbf{A}_{\boldsymbol{\theta}^n}$, a non-$\alpha$ point $\mathbf{a}_{\overline{\alpha}}$, and their most recent common ancestor with α is $\mathbf{A}_{\boldsymbol{\theta}^c}$. Mapping $\mathbf{B}$ to $\mathbf{A}_{\boldsymbol{\theta}^n}$, $\mathbf{c}$ to $\mathbf{a}_{\overline{\alpha}}$ and $\mathbf{C}$ to $\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}$. Hence, given the conditions stated in Corollary 3,

$$a.s. \quad \mathcal{G}(\widetilde{\mathbf{A}}'_{\boldsymbol{\theta}^c}, \tilde{s}'_{\alpha}) \overset{a.s.}{\approx} \mathcal{G}(\widetilde{\mathbf{A}}_{\boldsymbol{\theta}^c}, \tilde{s}_{\alpha}). \tag{60}$$

This proves Corollary 3.