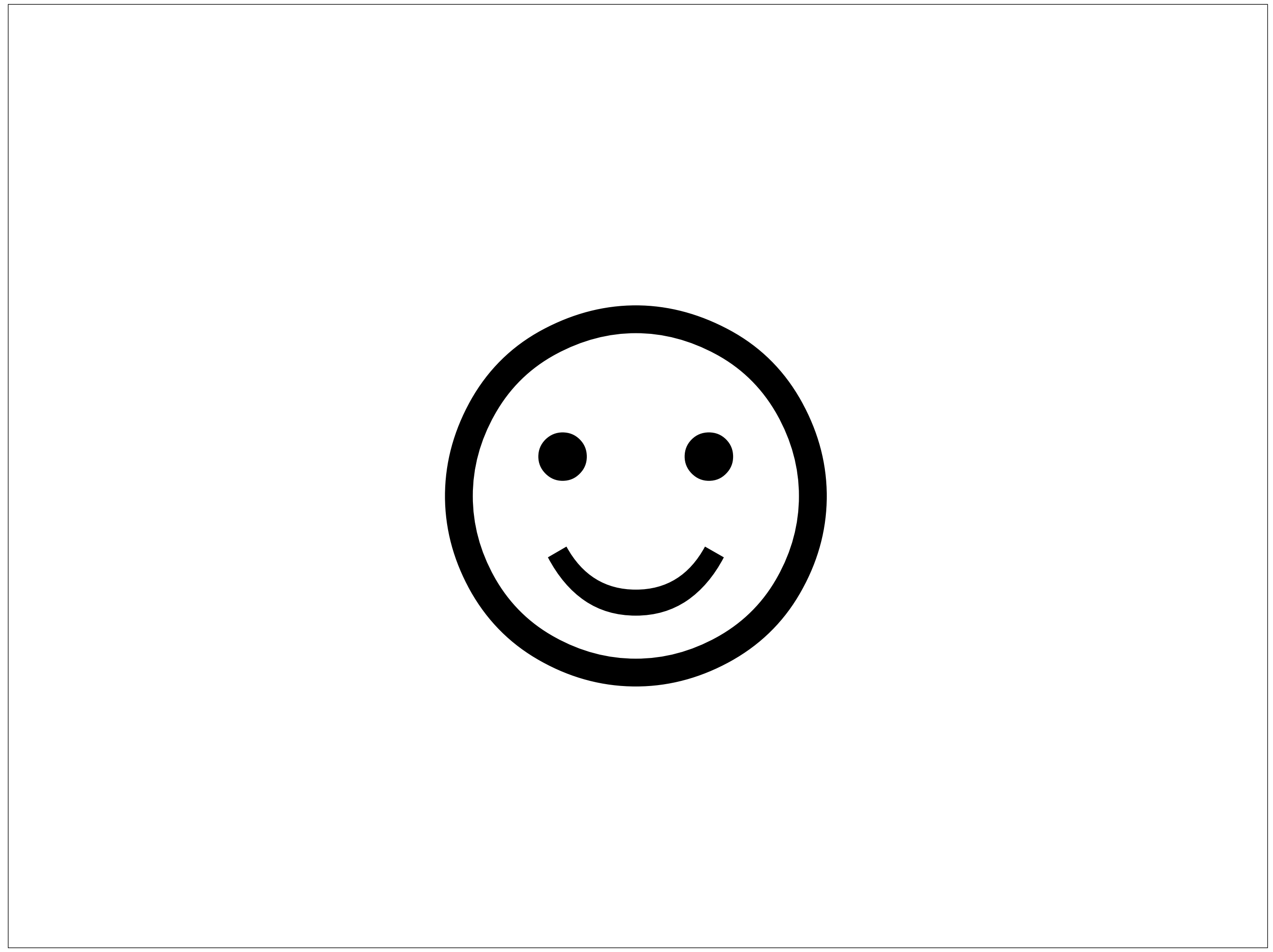
HIERARCHICAL SHAPE RETRIEVAL on a UNIT HYPERSPHERE

Yixin Lin
Glizela Taino
Mark Moyou
Adrian M. Peter



Zela
Can you recognize what this is?

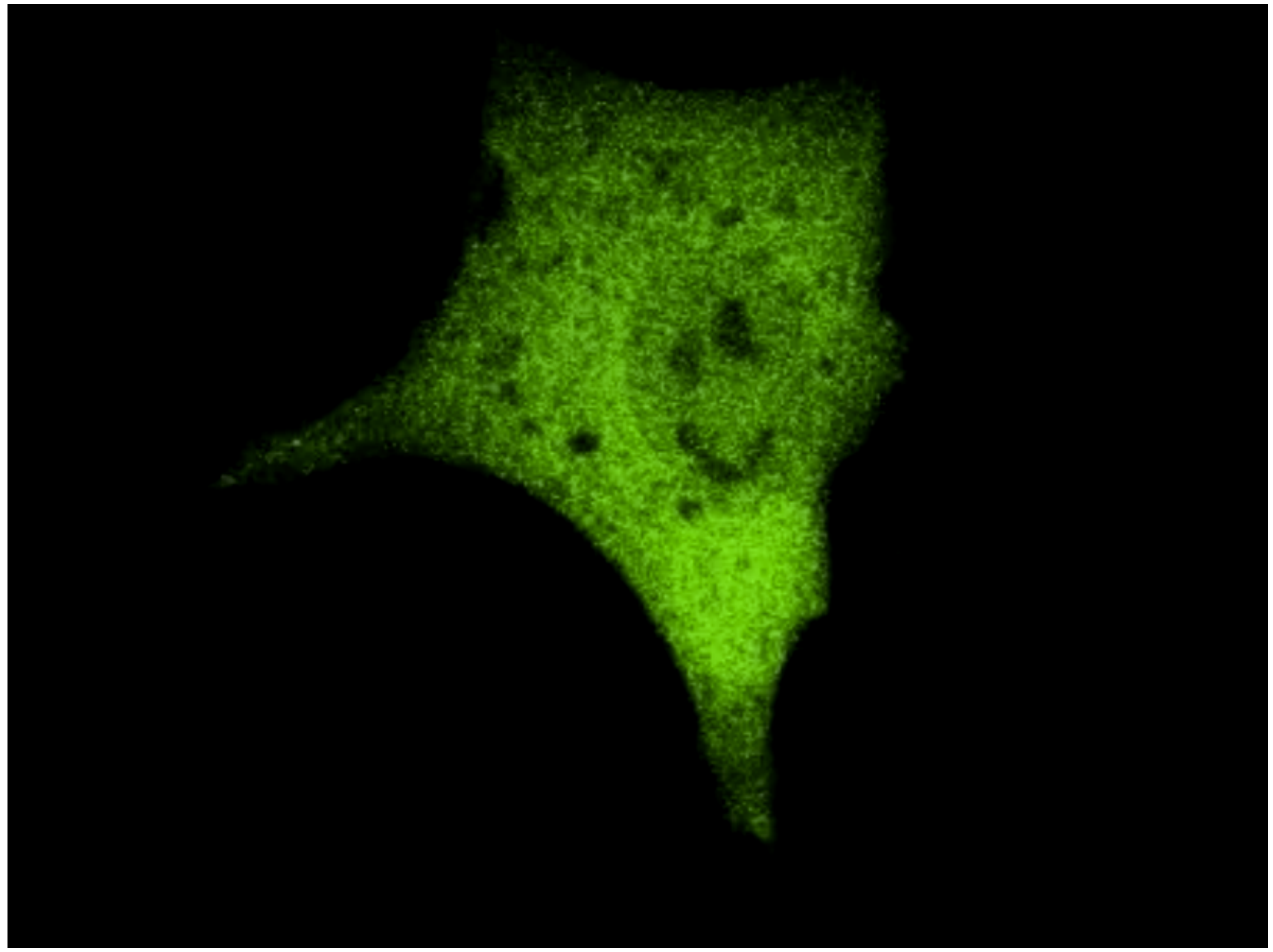
A smiley face, of course.

Zela
What about this?

Of course. This is a famous symbol from the novel Watchmen, a smiley face pin defaced by a drop of blood. You can still recognize it despite the color change, rotation, scaling, and of course the blood.

Yixin
What about this?

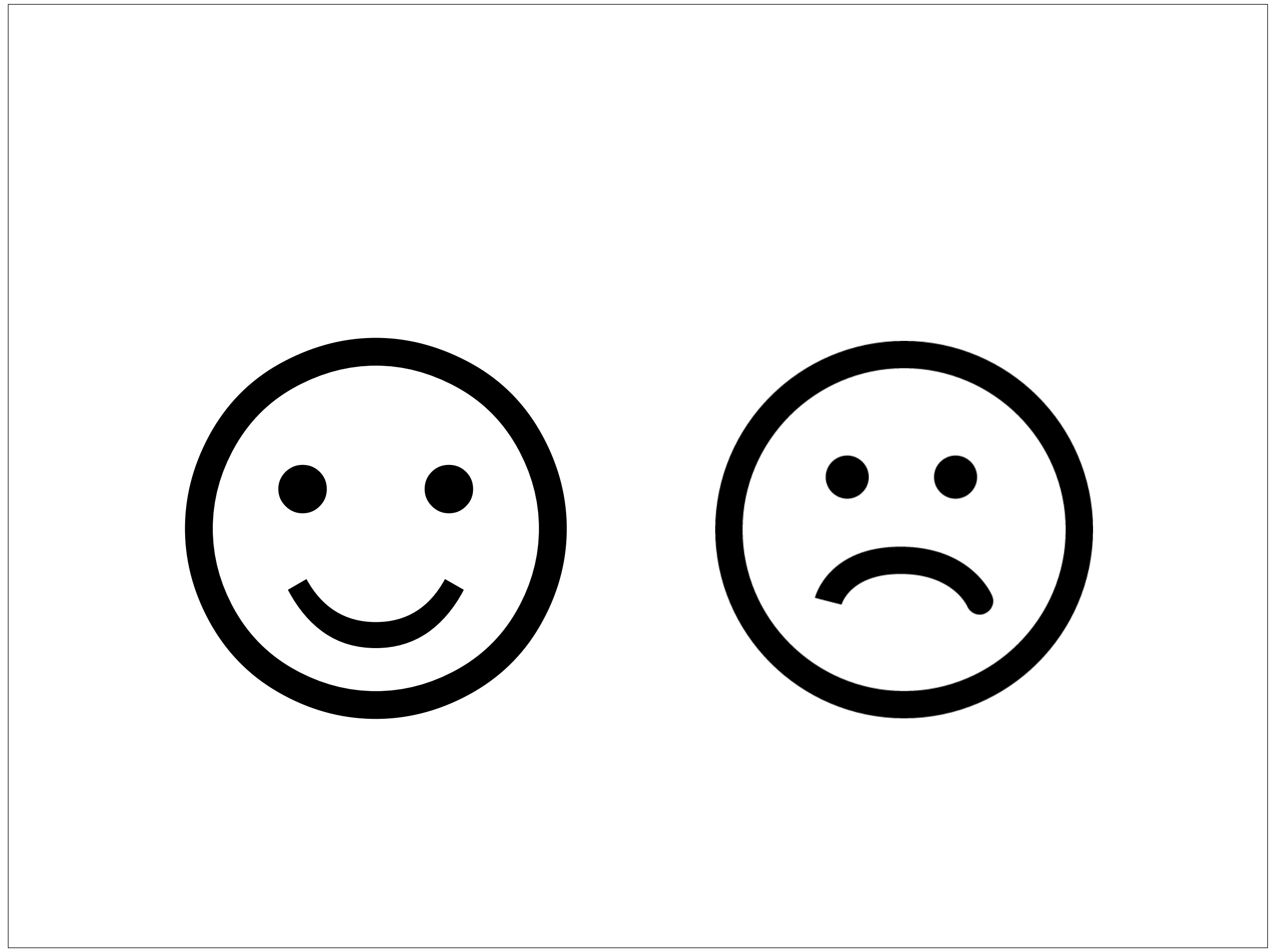
This is an artificial nanostructure called scaffolded DNA origami. It's at a scale millions of times smaller than the pin in the last slide. But despite the distortion from the electron microscope… and from being made out of nucleic acid, you can still tell what it is!



Yixin
This is a cancer cell literally laughing at human beings. Again a completely different scale, extreme color change, texture distortion, and warping of the image. Again, we can tell exactly how smug it's being.

Zela
Here’s a smile in the sky.

Zela
Here’s a smile from outer space, viewed by the Hubble space telescope.



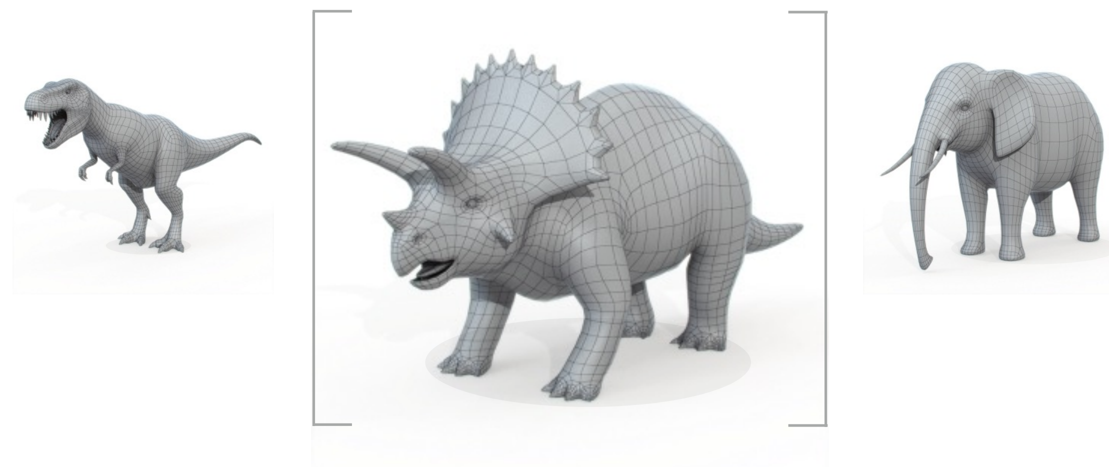
Zela
And yet when we see an unhappy face that looks almost exactly identical to a happy one, we can immediately tell it apart.

What is it about our understanding of these images that let us understand meaning? It’s certainly not color, nor scale, nor texture. What allows us see through any amount of distortions, imperfections, or warping?

It’s shape. If you don’t understand shape, you’re almost blind.

WHAT IS

SHAPE RETRIEVAL

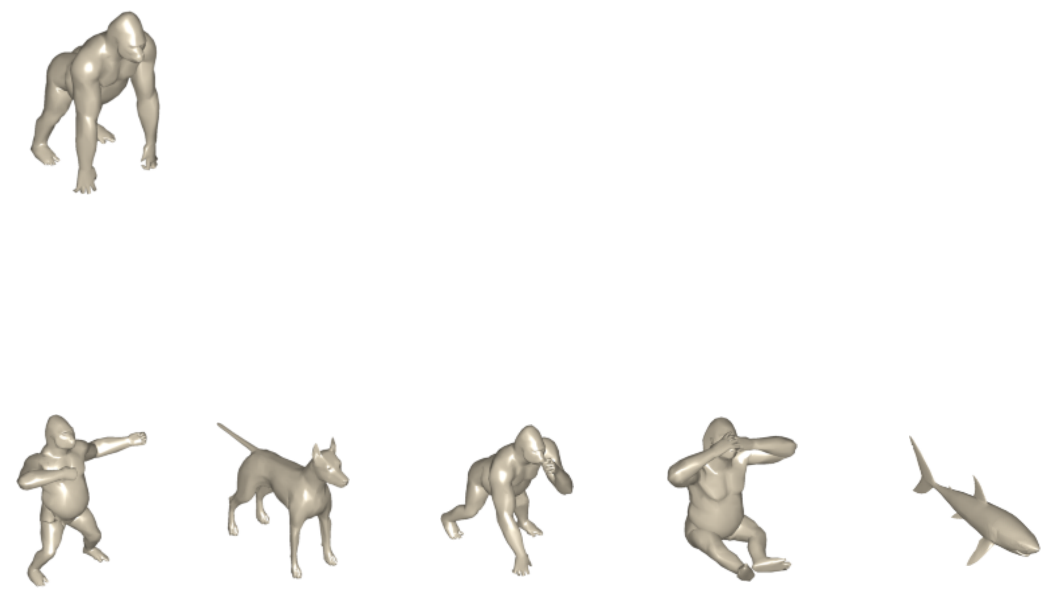


Yixin

Our goal is to allow computers to understand shape like humans naturally do, to see through all the distortions to capture the essence of an object, an important piece of the quest to make machines more intelligent. The question of shape retrieval is: given a shape, how can we find the most similar shapes from our database?

WHAT IS

SHAPE RETRIEVAL



Yixin

Our goal is to allow computers to understand shape like humans naturally do, to see through all the distortions to capture the essence of an object, an important piece of the quest to make machines more intelligent. The question of shape retrieval is: given a shape, how can we find the most similar shapes from our database?

3D
ANIMATION

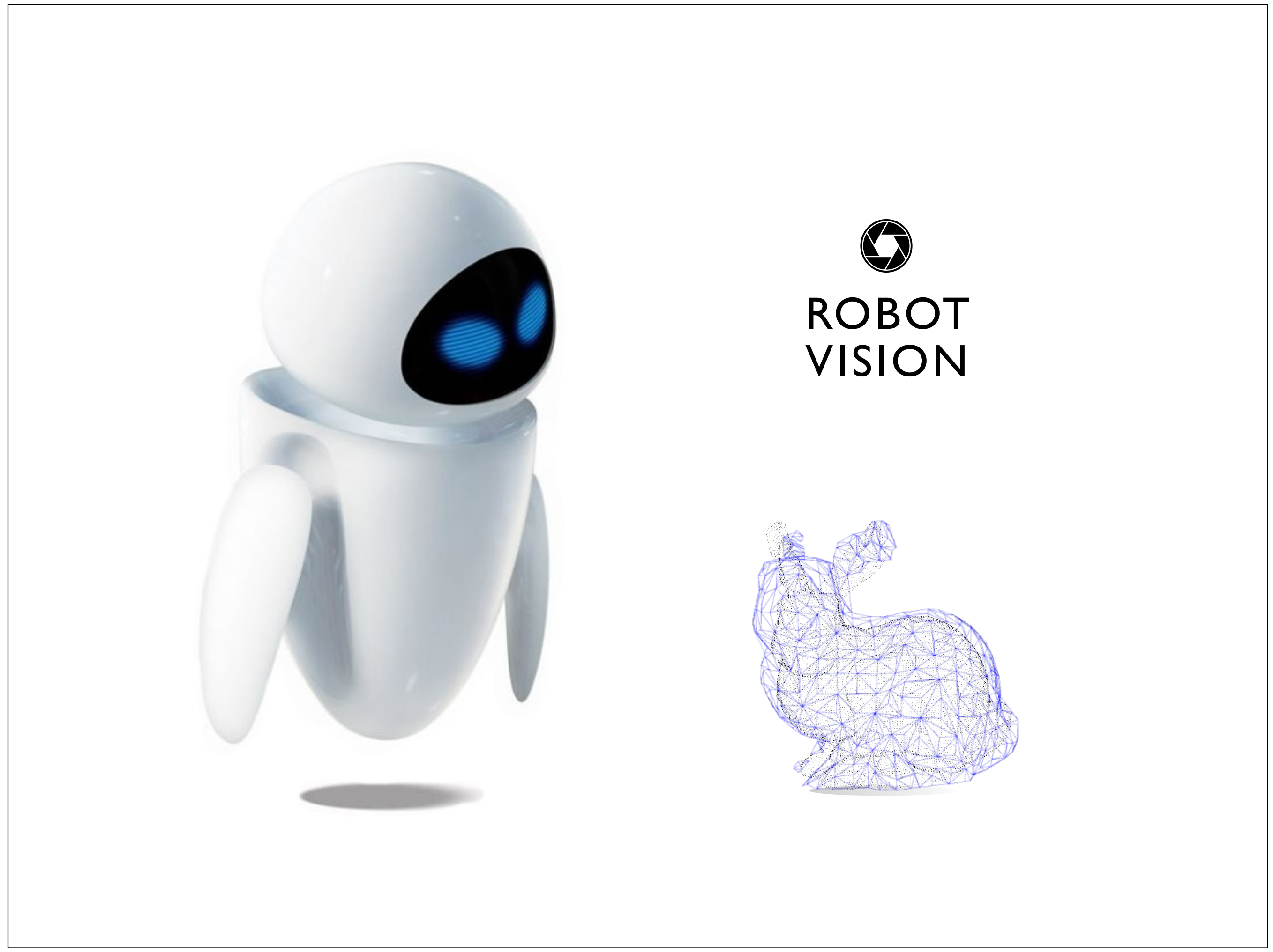
Yixin

One potential application is for use in 3D design and animation. Every billion dollar animated film has millions of 3D shapes that let the story come to life. If you can instantly find the similar ones among those millions of shapes, you can accelerate that creative process tremendously.

3D ANIMATION

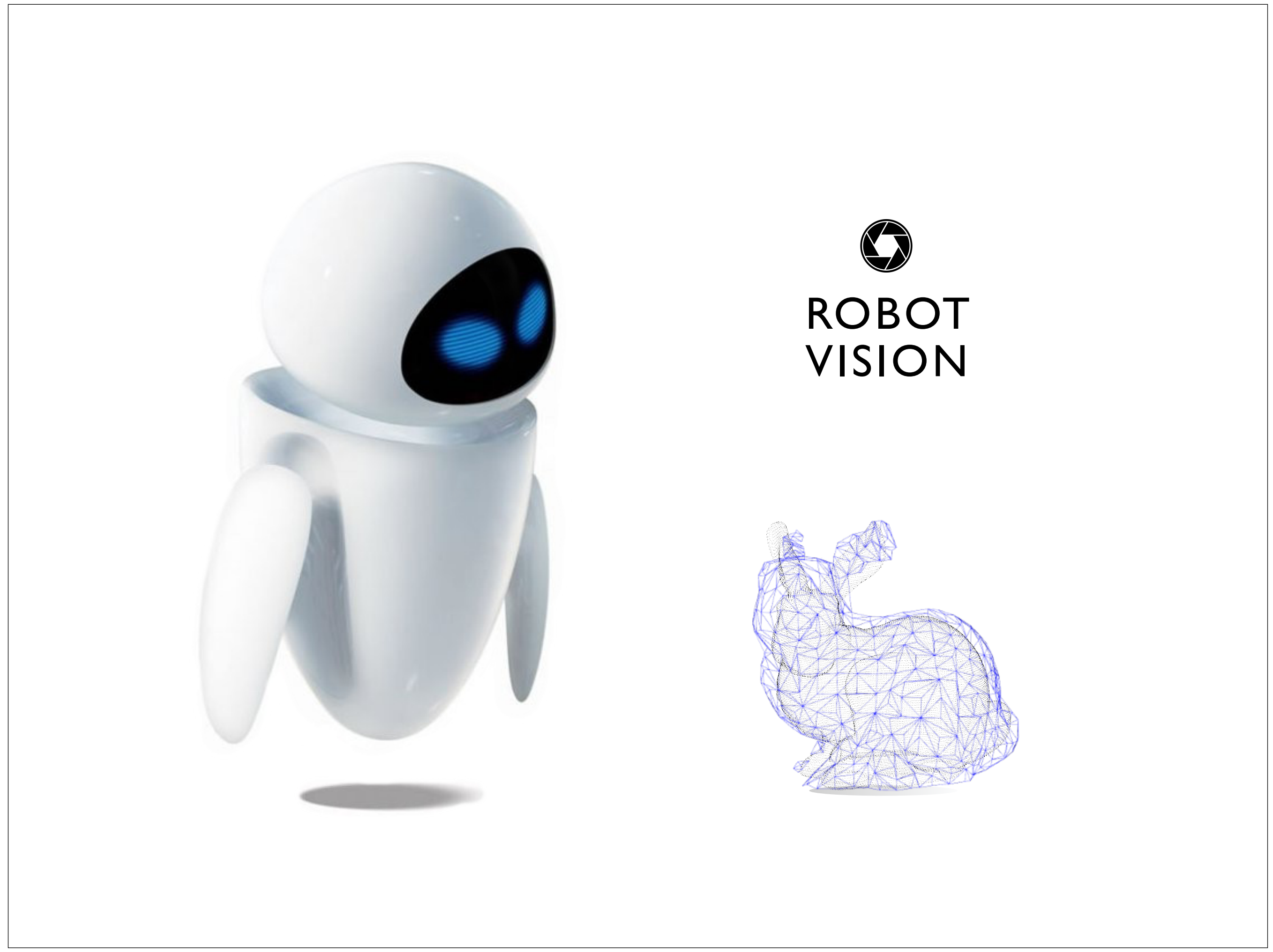
Yixin

One potential application is for use in 3D design and animation. Every billion dollar animated film has millions of 3D shapes that let the story come to life. If you can instantly find the similar ones among those millions of shapes, you can accelerate that creative process tremendously.



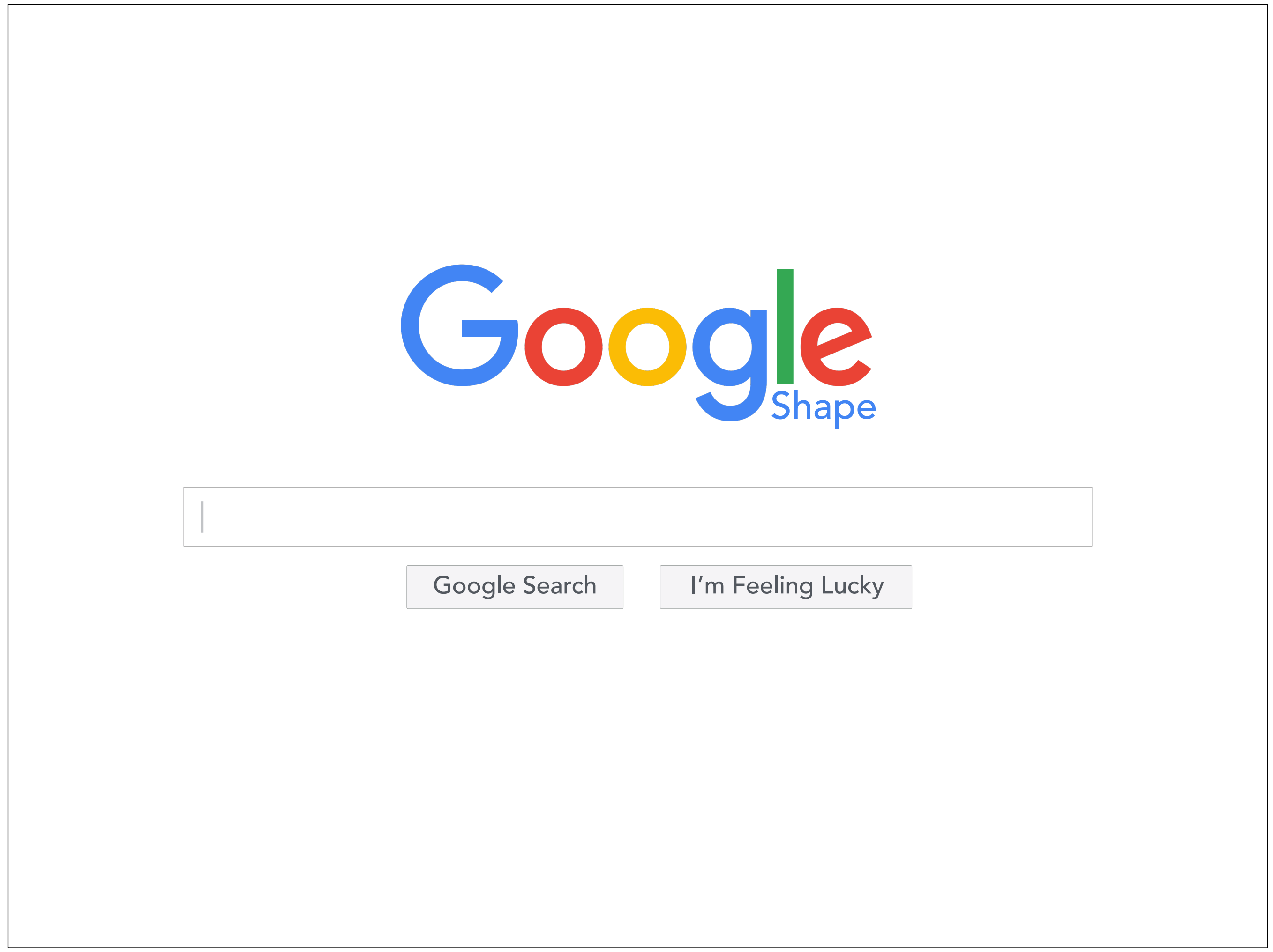
Zela

Even more exciting are the potentials for robot and computer vision. It's fairly simple to create point clouds of the surrounding world: all you need is a pair of stereoscopic cameras or more complicated equipment like LIDAR. If a robot can use depth and 3D shape in addition to traditional 2D image methods, it can understand the world around it in a more human way. Imagine if a robot can instantly bring up all similar objects from a cloud database: how much better would it understand its environment?



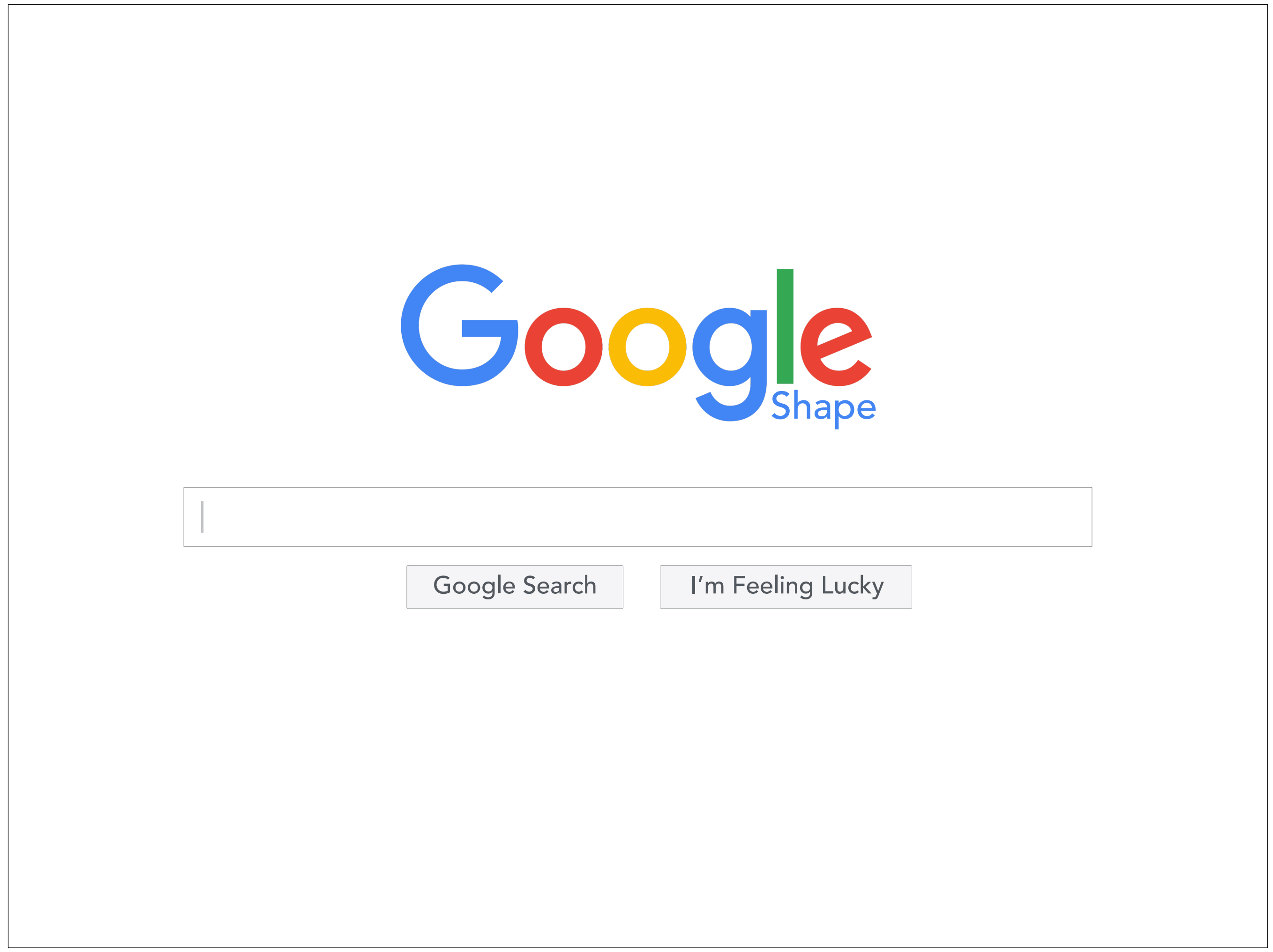
Zela

Even more exciting are the potentials for robot and computer vision. It’s fairly simple to create point clouds of the surrounding world: all you need is a pair of stereoscopic cameras or more complicated equipment like LIDAR. If a robot can use depth and 3D shape in addition to traditional 2D image methods, it can understand the world around it in a more human way. Imagine if a robot can instantly bring up all similar objects from a cloud database: how much better would it understand its environment?



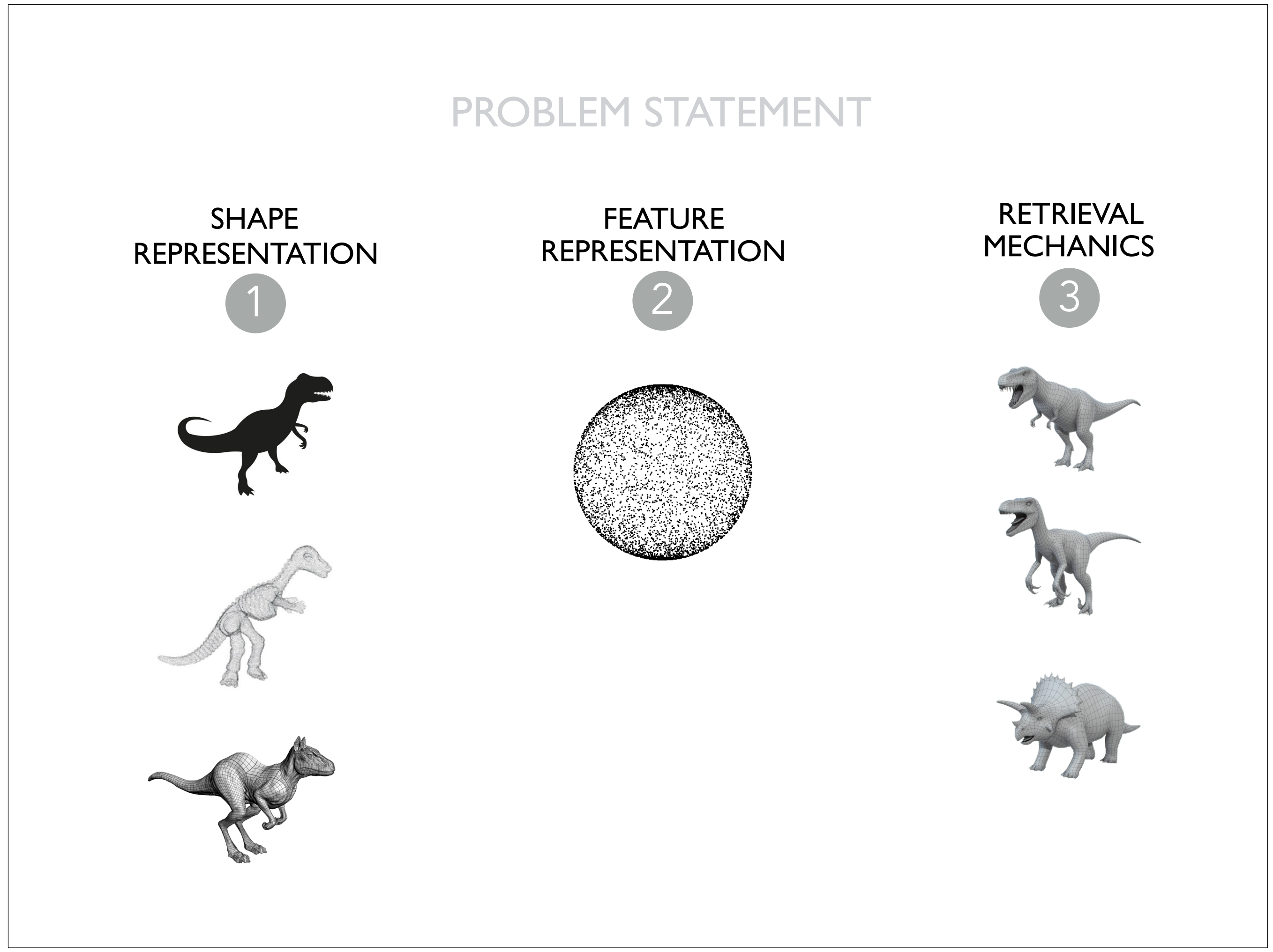
Zela

Ultimately, understanding and retrieving shapes is one of those fundamental problems in computer vision that, if solved, would let you build a search platform that helps solve problems we didn’t even know existed. The goal is to let another dimension of the world be accessible and retrievable by anyone.



Zela

Ultimately, understanding and retrieving shapes is one of those fundamental problems in computer vision that, if solved, would let you build a search platform that helps solve problems we didn’t even know existed. The goal is to let another dimension of the world be accessible and retrievable by anyone.



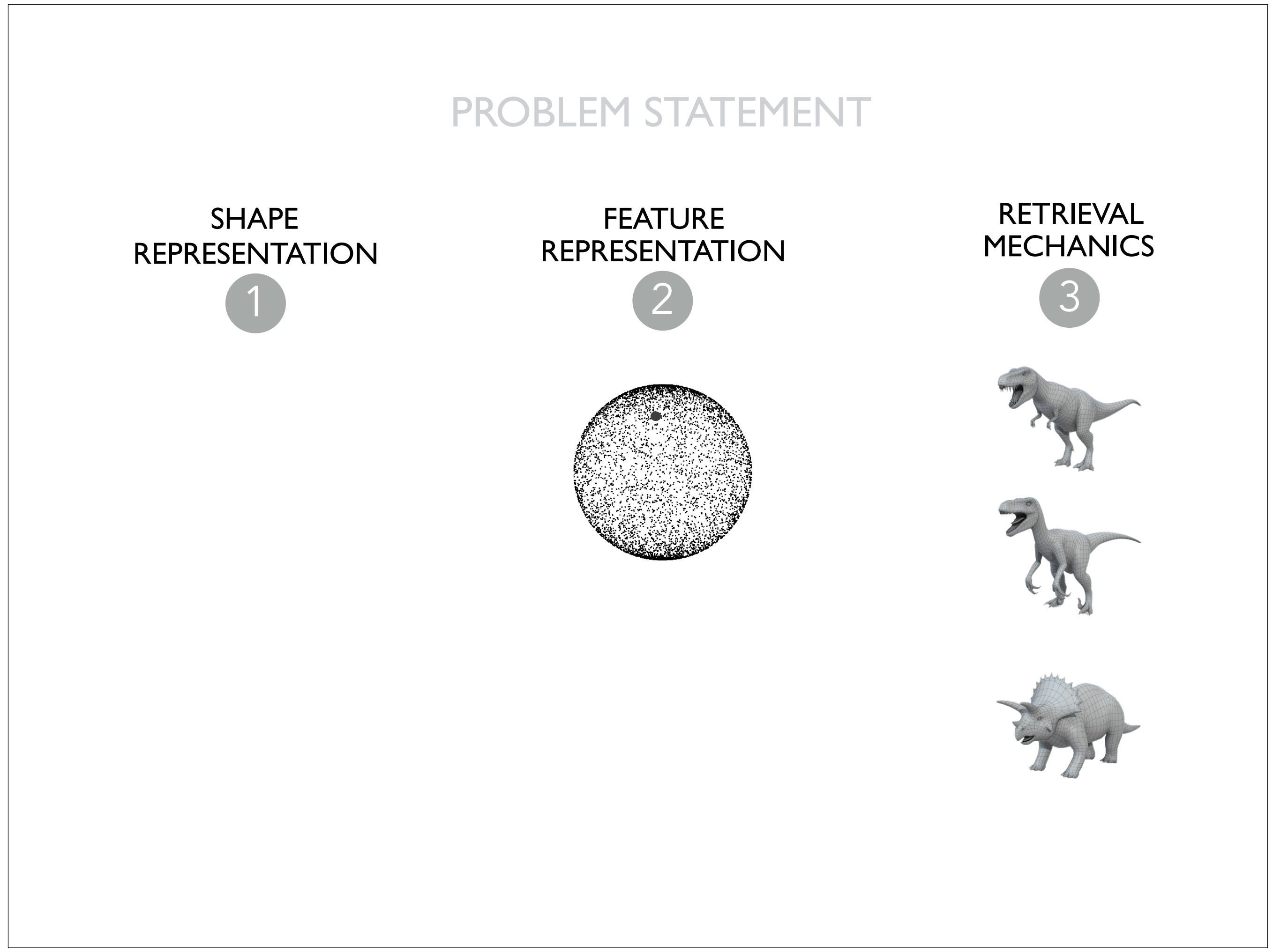
Yixin

- Shape representation
    - 2d points
    - 3d point cloud
    - 3d mesh: add polygons between the points
- Turn this into feature representation
    - Capture the essence of the data
- With this nice feature representation, we can bring all of our machine learning techniques: we'll be using unsupervised hierarchical clustering
- Our main work is retrieval: given a completely new query shape, what are the ones closest to it?

- Shape representation given as 2D point sets, 3D point sets, or 3D meshes
- Unsupervised data —> have to classify
- From SR we transform raw data into a FR that can be utilized in Shape Classification
- Not focused on feature representation. Focus on retrieval.

Transition: Important point, SR to FR onto Hypersphere.

** Show 2D Shape, bunny and dinosaur.



Yixin

- Shape representation
    - 2d points
    - 3d point cloud
    - 3d mesh: add polygons between the points
- Turn this into feature representation
    - Capture the essence of the data
- With this nice feature representation, we can bring all of our machine learning techniques: we’ll be using unsupervised hierarchical clustering
- Our main work is retrieval: given a completely new query shape, what are the ones closest to it?

- Shape representation given as 2D point sets, 3D point sets, or 3D meshes
- Unsupervised data —> have to classify
- From SR we transform raw data into a FR that can be utilized in Shape Classification
- Not focused on feature representation. Focus on retrieval.

Transition: Important point, SR to FR onto Hypersphere.

** Show 2D Shape, bunny and dinosaur.

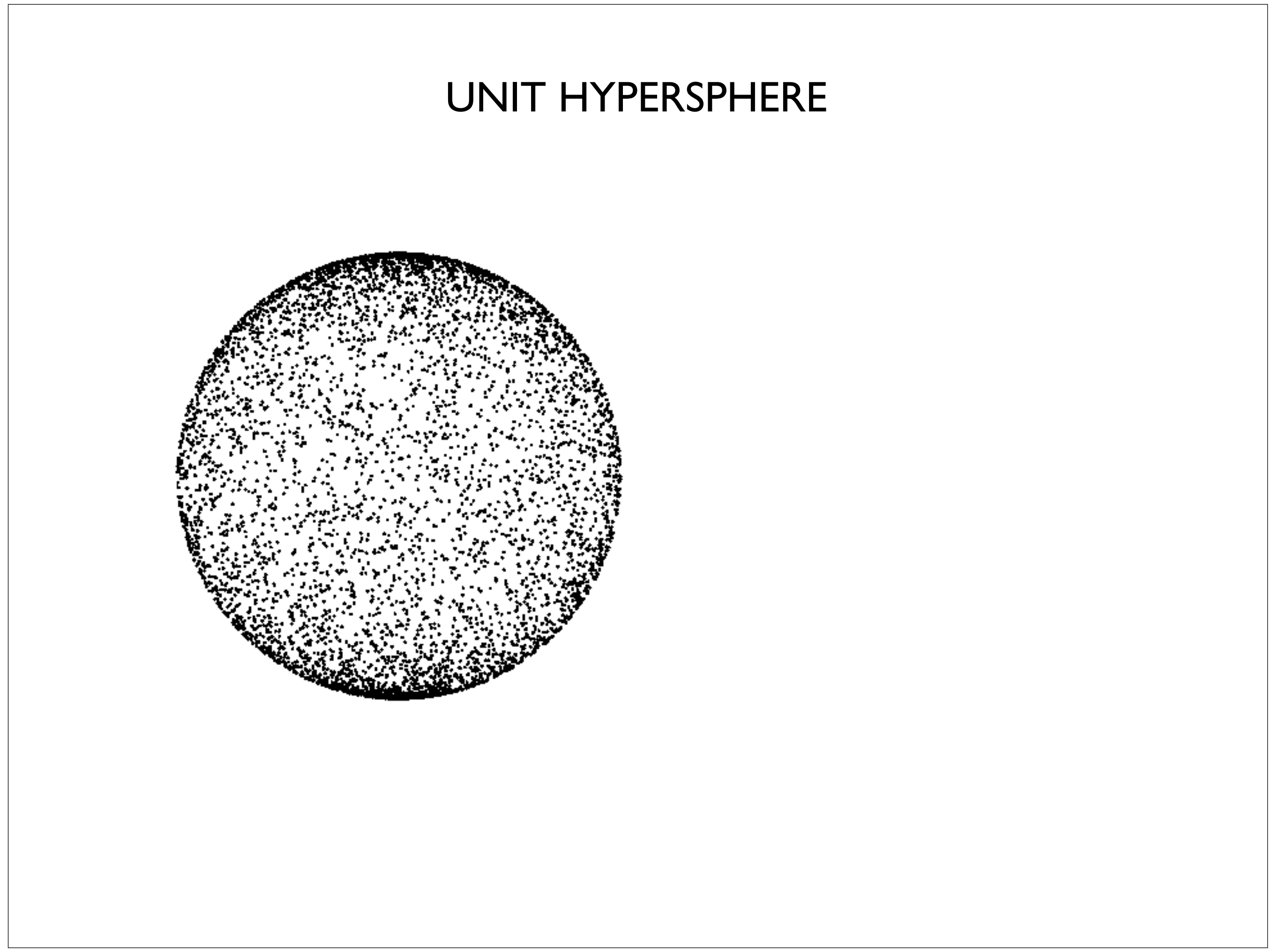
UNIT HYPERSPHERE

Yixin

- Nice properties of our feature representation:
  - Isometry invariance: any distance-preserving transformation maps
  - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
  - Can't simply take the Euclidean mean
  - Lets you take means by estimating what it is on a "flat" tangent space, remapping it back to get a new estimate, and iterate
  - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

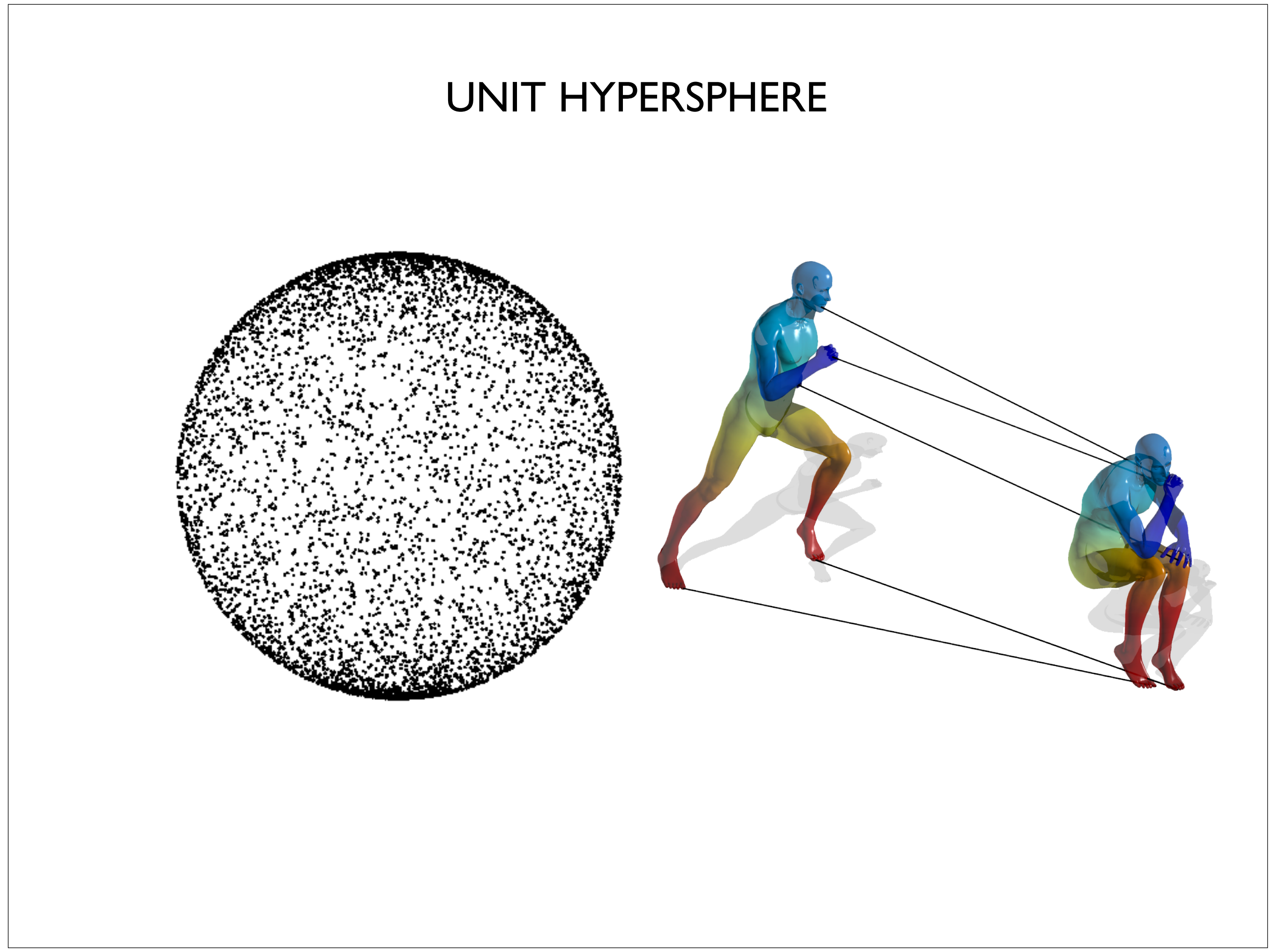


Yixin

- Nice properties of our feature representation:
  - Isometry invariance: any distance-preserving transformation maps
  - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
  - Can’t simply take the Euclidean mean
  - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
  - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin



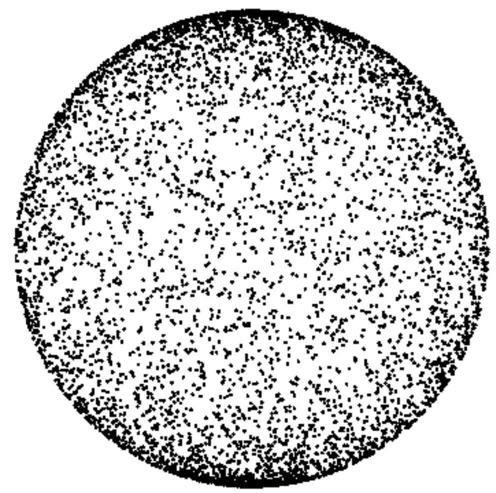
Yixin

- Nice properties of our feature representation:
  - Isometry invariance: any distance-preserving transformation maps
  - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
  - Can’t simply take the Euclidean mean
  - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
  - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

UNIT HYPERSPHERE

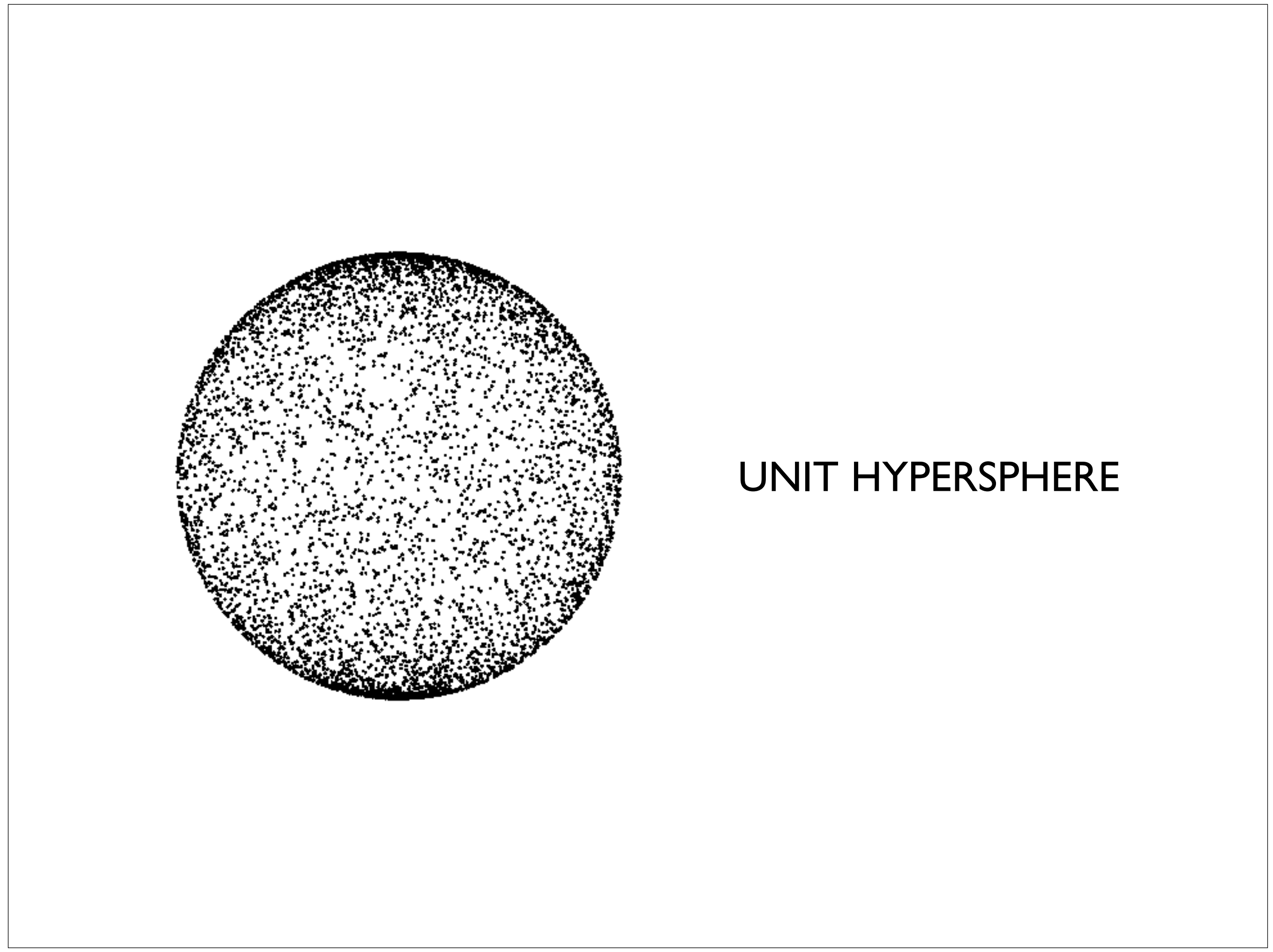


Yixin

- Nice properties of our feature representation:
    - Isometry invariance: any distance-preserving transformation maps
    - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
    - Can’t simply take the Euclidean mean
    - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
    - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

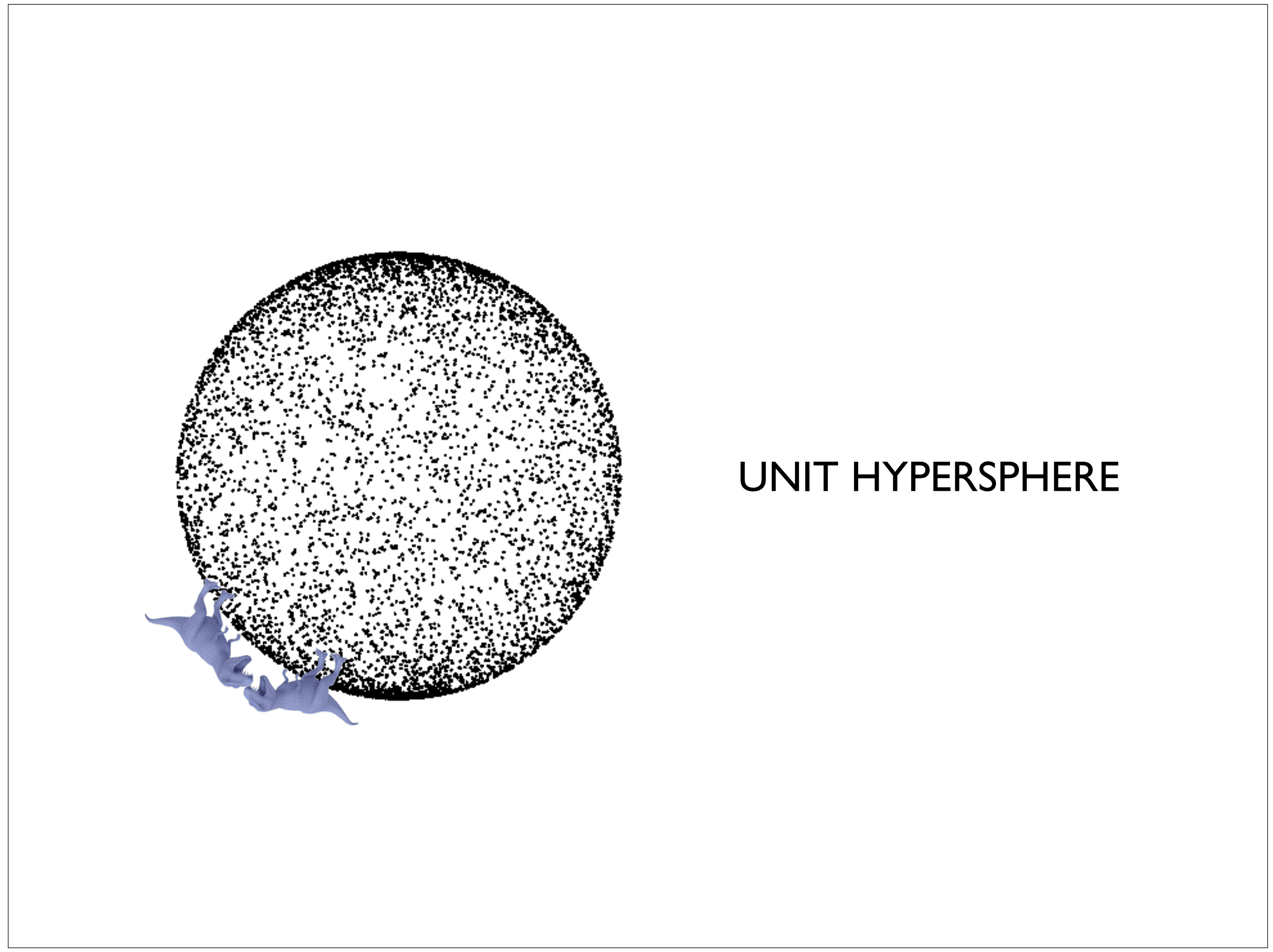


Yixin

- Nice properties of our feature representation:
    - Isometry invariance: any distance-preserving transformation maps
    - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
    - Can’t simply take the Euclidean mean
    - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
    - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

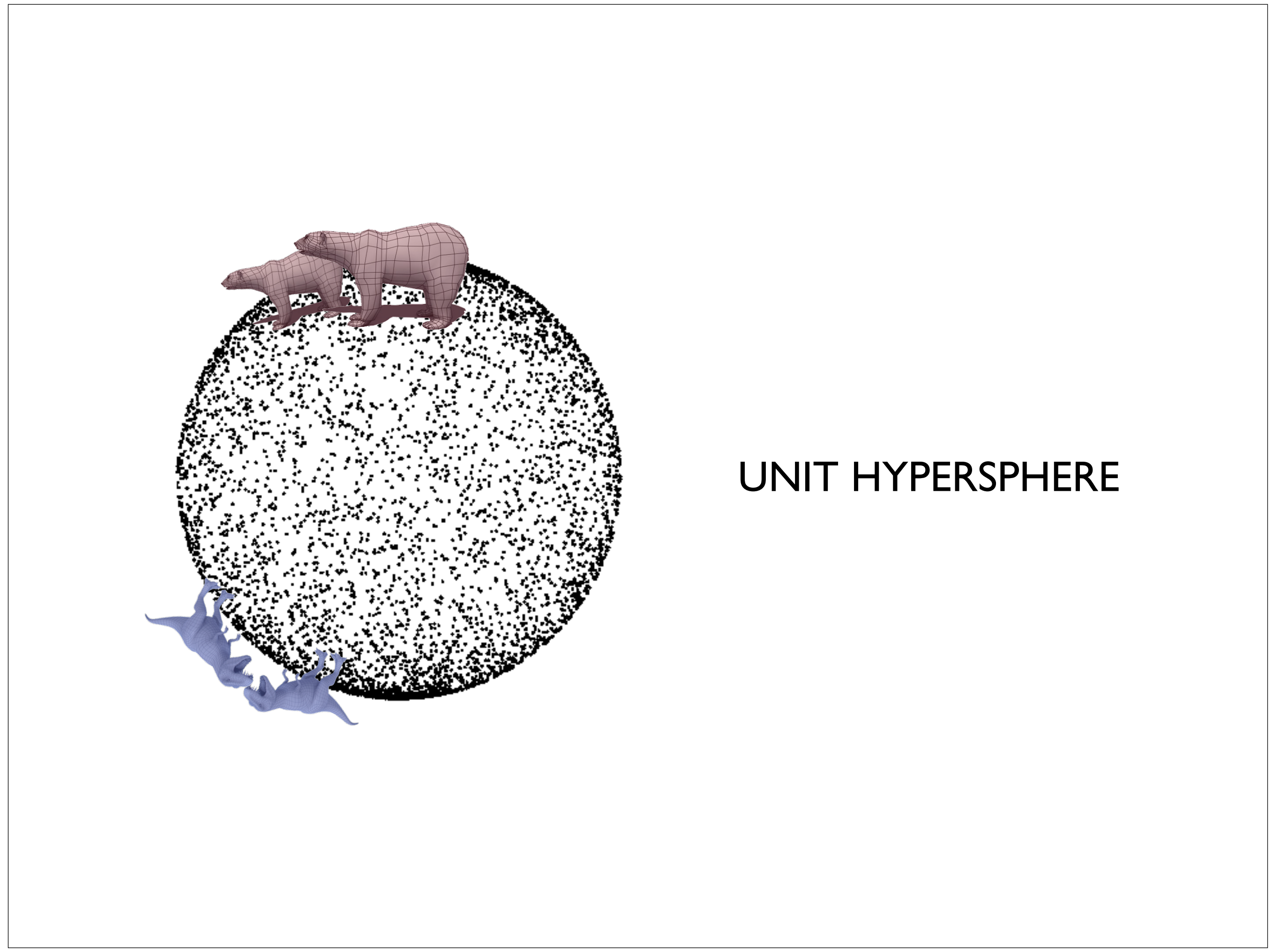


Yixin

- Nice properties of our feature representation:
  - Isometry invariance: any distance-preserving transformation maps
  - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
  - Can’t simply take the Euclidean mean
  - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
  - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

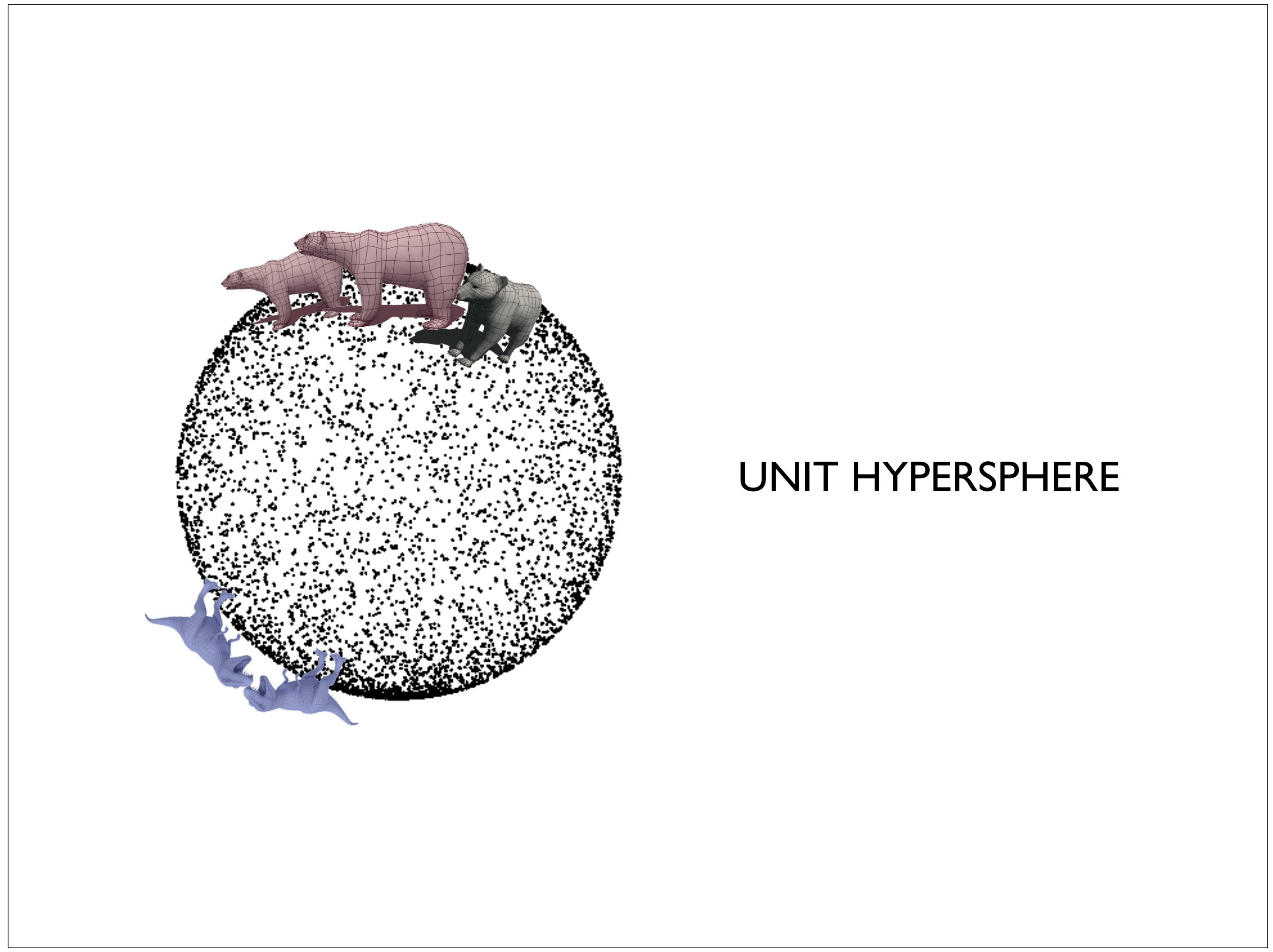


Yixin

- Nice properties of our feature representation:
    - Isometry invariance: any distance-preserving transformation maps
    - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
    - Can't simply take the Euclidean mean
    - Lets you take means by estimating what it is on a "flat" tangent space, remapping it back to get a new estimate, and iterate
    - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

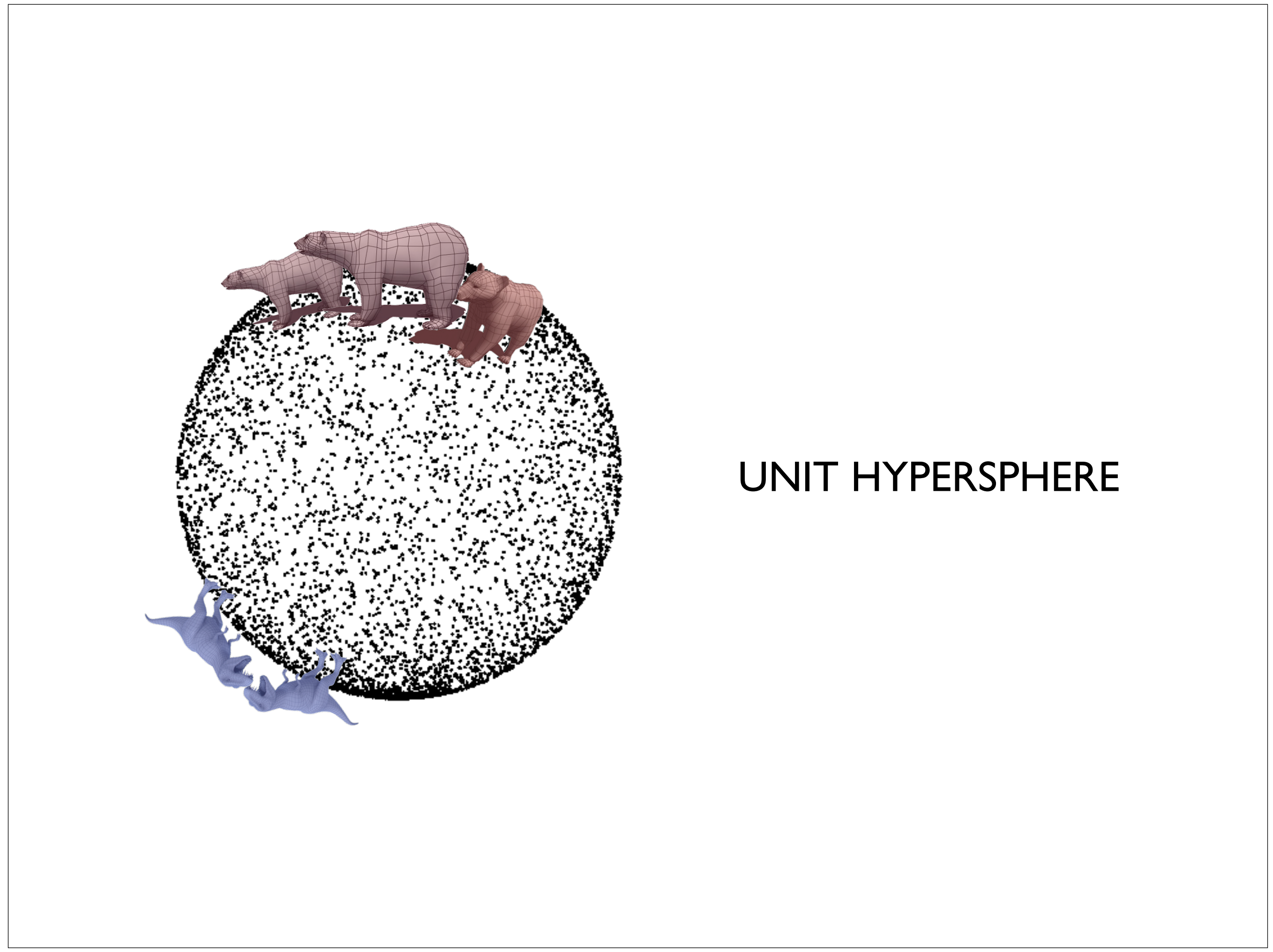


Yixin

- Nice properties of our feature representation:
    - Isometry invariance: any distance-preserving transformation maps
    - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
    - Can’t simply take the Euclidean mean
    - Lets you take means by estimating what it is on a “flat” tangent space, remapping it back to get a new estimate, and iterate
    - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin



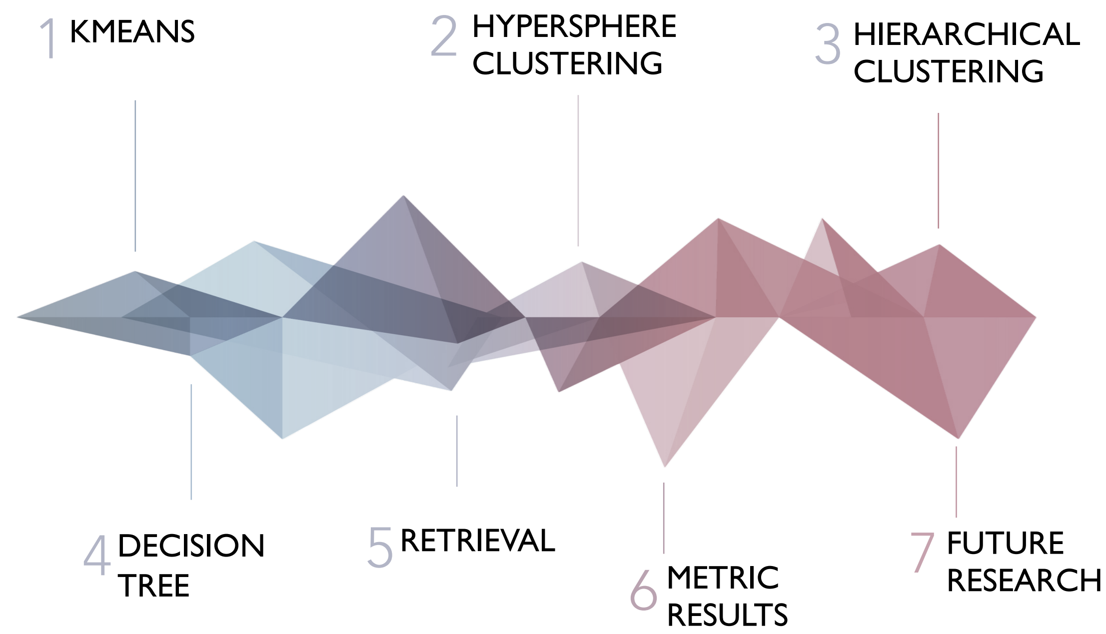
Yixin

- Nice properties of our feature representation:
  - Isometry invariance: any distance-preserving transformation maps
  - Brings us to unit hypersphere geometry, simple norm
- Distances between points signify similarity
- Form cluster representations of shape classes, e.g. bears, dinosaurs
- Given a new query, we can classify it based on our previous clusters
- Karcher mean
  - Can't simply take the Euclidean mean
  - Lets you take means by estimating what it is on a "flat" tangent space, remapping it back to get a new estimate, and iterate
  - Lets you take means on surfaces

** Add query shape animation of bear to solidify objective of query shape belonging to a cluster

Yixin

ROADMAP FOR RETRIEVAL



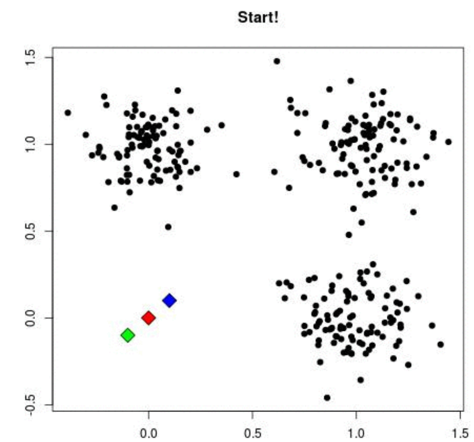
- Zela
-

UNDERSTANDING KMEANS

KMEANS CLUSTERING

EUCLIDEAN SPACE

1: generate an initial set of *k* means
2: **while** not converged **do**
3: assign points to nearest centroid
4: calculate new centroids of the points in the cluster
5: **end while**



Zela

Kmeans clustering is a two staged iterative process. The first stage is the assignment stage. Each observation is assigned to the nearest centroid. Then the centroids are updated to be placed in the middle of a cluster, or attempts to minimize its distance between itself and all members of its class. So then with the new updated centroid, the process repeats until centroids converge to a local minimum distance between all members in its class.

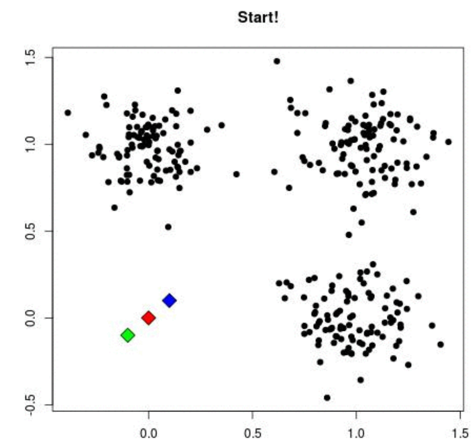
Kmeans clusters on flat surface. In our problem we need to cluster on a spherical manifold.

UNDERSTANDING KMEANS

KMEANS CLUSTERING

EUCLIDEAN SPACE

1: generate an initial set of *k* means
2: **while** not converged **do**
3: assign points to nearest centroid
4: calculate new centroids of the points in the cluster
5: **end while**



Zela
Kmeans clustering is a two staged iterative process. The first stage is the assignment stage. Each observation is assigned to the nearest centroid. Then the centroids are updated to be placed in the middle of a cluster, or attempts to minimize its distance between itself and all members of its class. So then with the new updated centroid, the process repeats until centroids converge to a local minimum distance between all members in its class.

Kmeans clusters on flat surface. In our problem we need to cluster on a spherical manifold.

HYPERSPHERE
CLUSTERING

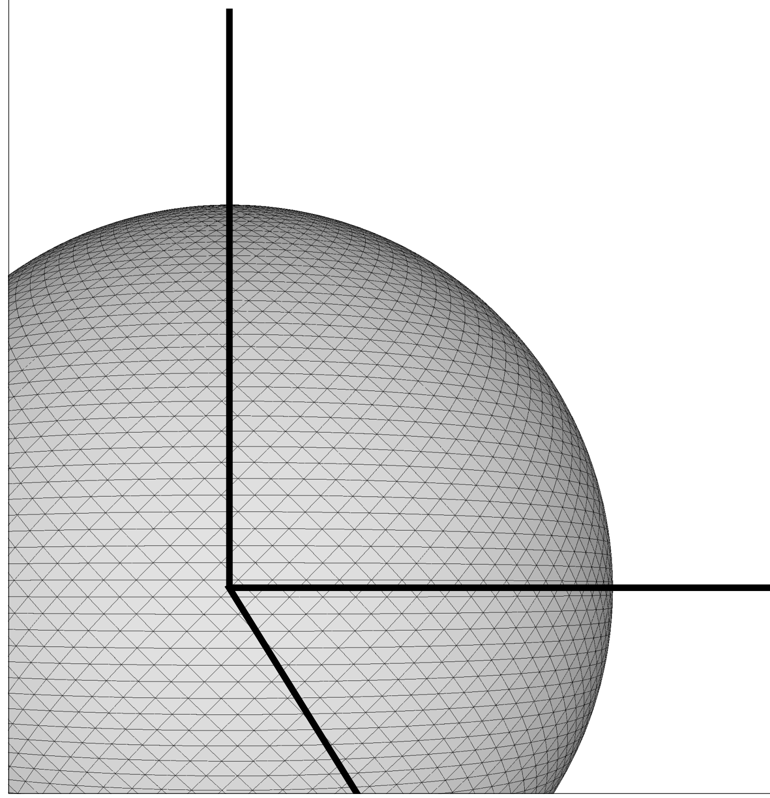
SPHERICAL KMEANS

Zela

- Same as k-means on a curved manifold, but we need a different way of computing the distances and means
- Our new distance metric is cosine similarity: it’s very easy to take the distance between two vectors which lie on the hypersphere; it’s just the angle between the two vectors
- Second, there exists a very nice approximation for the Karcher mean on a hypersphere which we used instead
- This second method finds a centroid using the normalized gravity center. It finds resultant of two vectors and then normalizes to map back onto the unit hypersphere

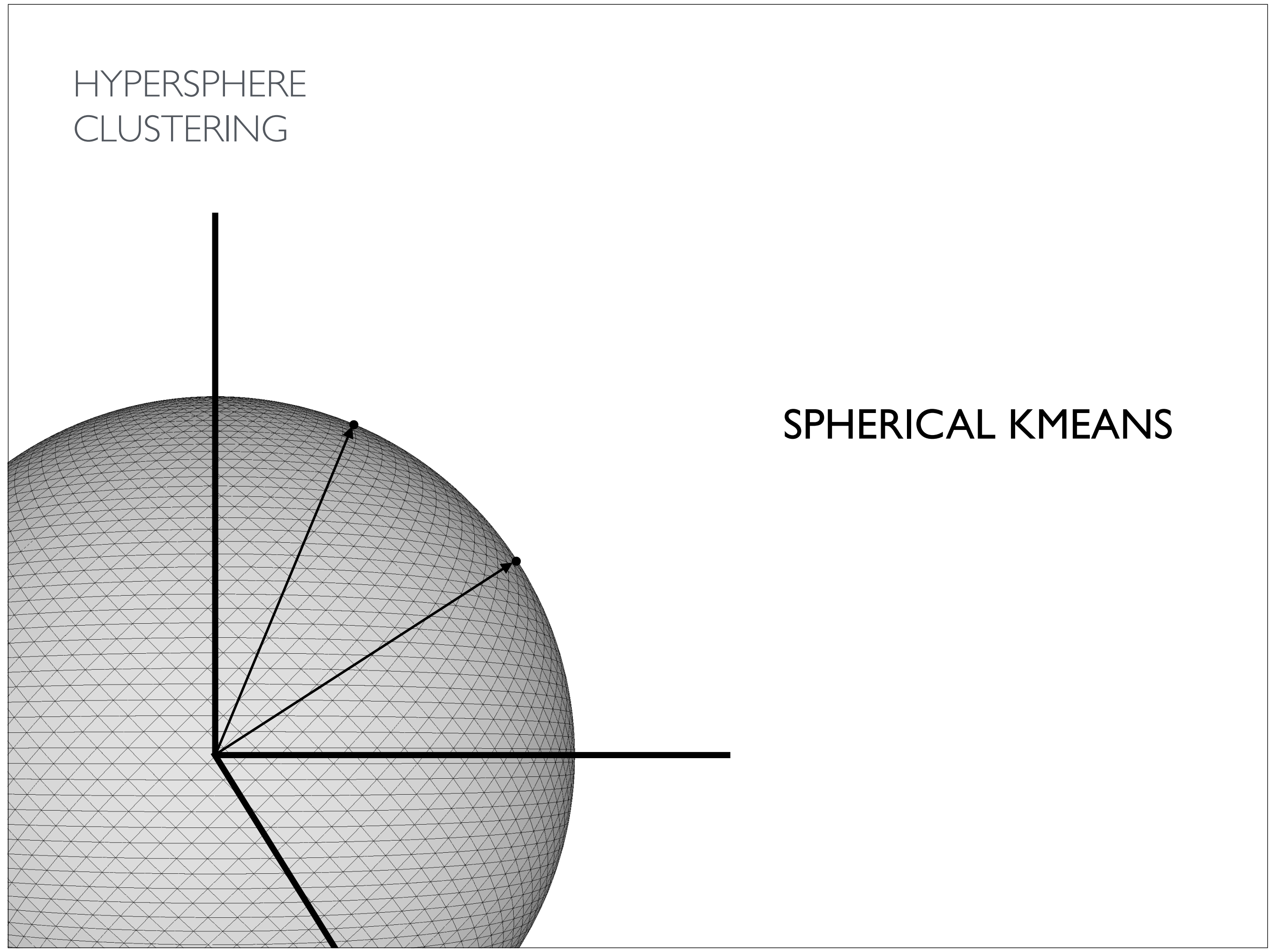
HYPERSPHERE
CLUSTERING

SPHERICAL KMEANS



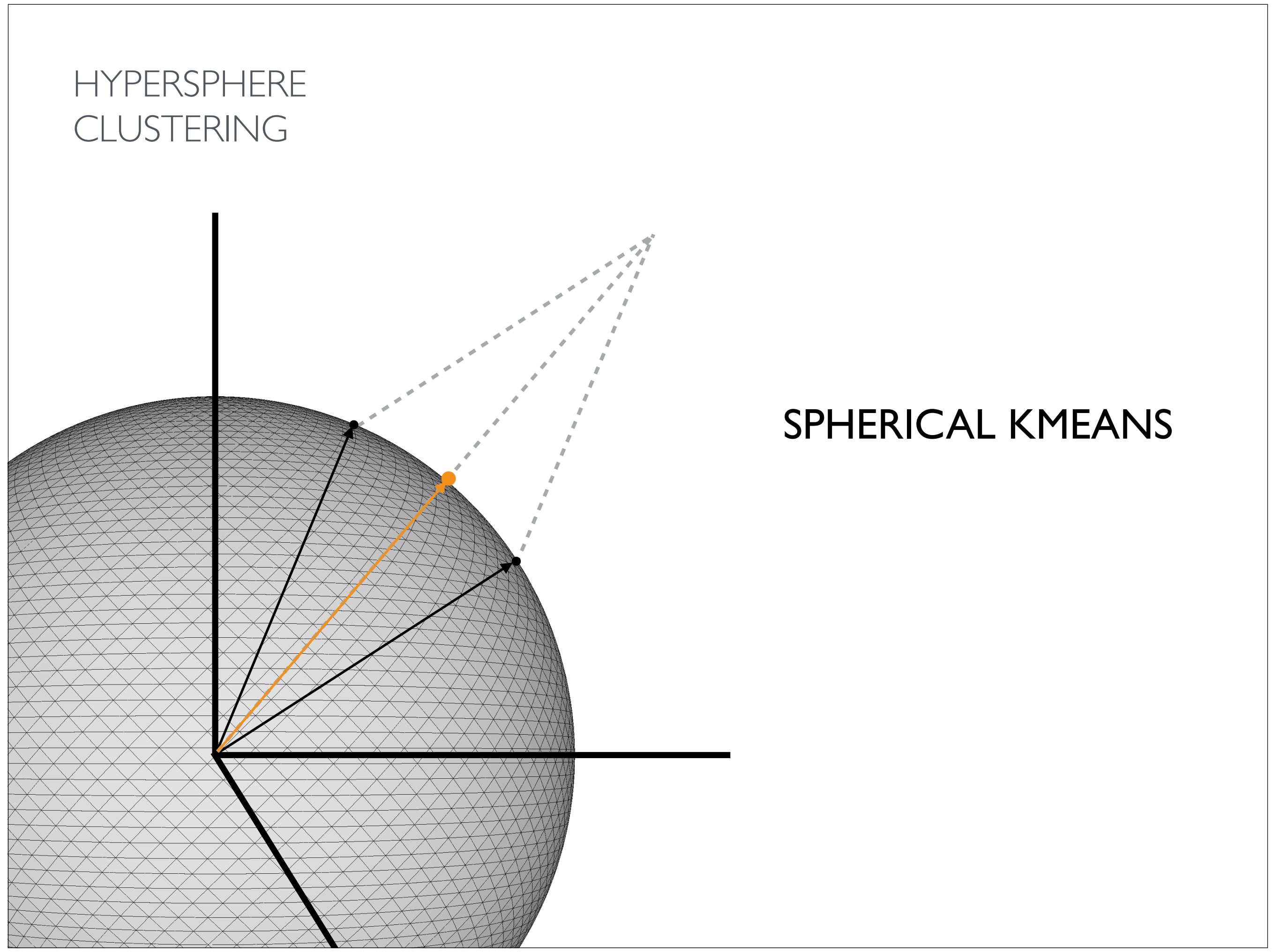
Zela

- Same as k-means on a curved manifold, but we need a different way of computing the distances and means
- Our new distance metric is cosine similarity: it’s very easy to take the distance between two vectors which lie on the hypersphere; it’s just the angle between the two vectors
- Second, there exists a very nice approximation for the Karcher mean on a hypersphere which we used instead
- This second method finds a centroid using the normalized gravity center. It finds resultant of two vectors and then normalizes to map back onto the unit hypersphere



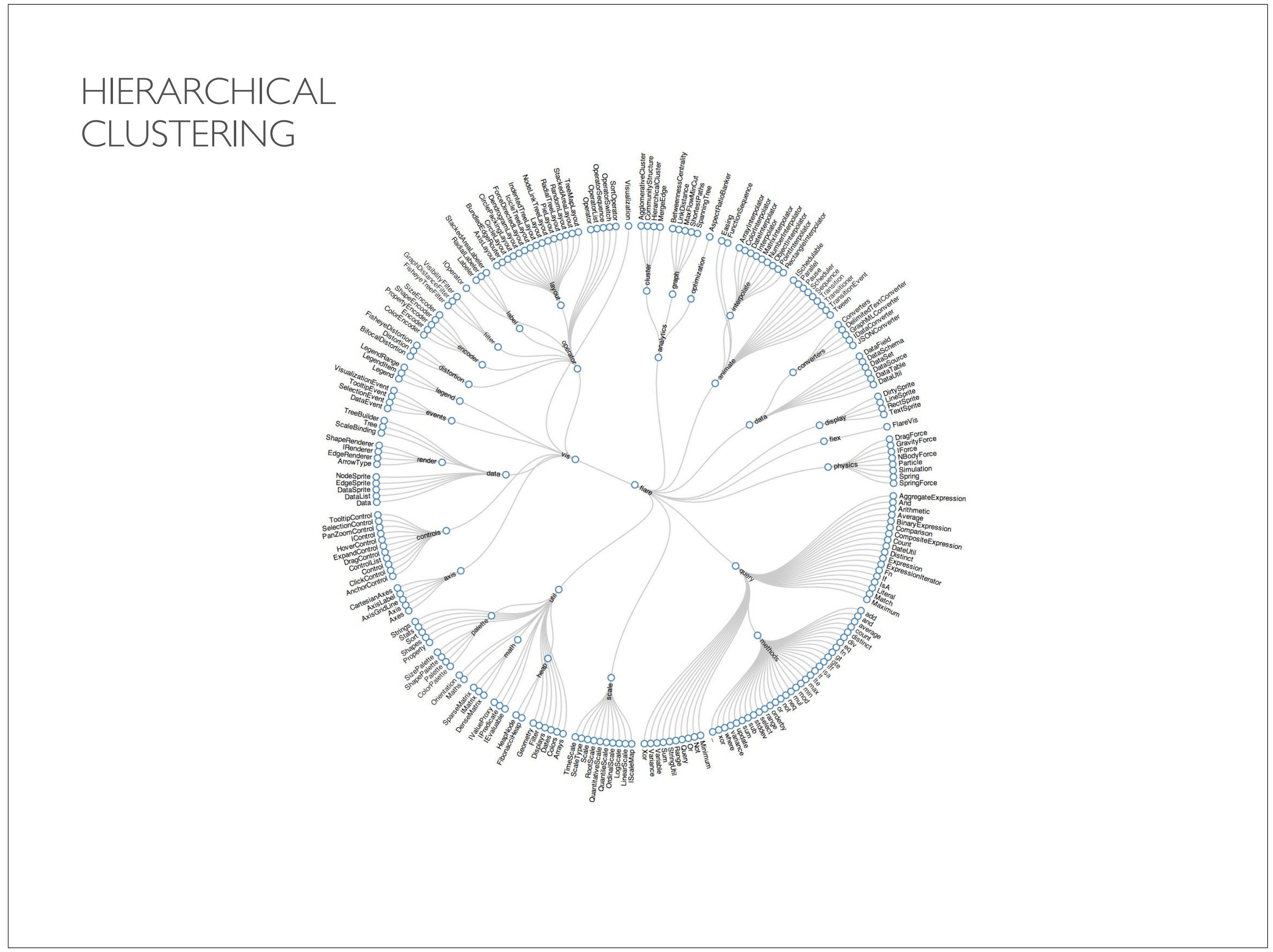
Zela

- Same as k-means on a curved manifold, but we need a different way of computing the distances and means
- Our new distance metric is cosine similarity: it’s very easy to take the distance between two vectors which lie on the hypersphere; it’s just the angle between the two vectors
- Second, there exists a very nice approximation for the Karcher mean on a hypersphere which we used instead
- This second method finds a centroid using the normalized gravity center. It finds resultant of two vectors and then normalizes to map back onto the unit hypersphere



Zela

- Same as k-means on a curved manifold, but we need a different way of computing the distances and means
- Our new distance metric is cosine similarity: it’s very easy to take the distance between two vectors which lie on the hypersphere; it’s just the angle between the two vectors
- Second, there exists a very nice approximation for the Karcher mean on a hypersphere which we used instead
- This second method finds a centroid using the normalized gravity center. It finds resultant of two vectors and then normalizes to map back onto the unit hypersphere



Yixin

- A method in cluster analysis to create hierarchies of clusters
    - Take similar data points and group them
    - This lets us capture the clustering structure of the data
    - Then group those groups, recursively
- Multilevel hierarchy encodes different layers of abstraction
- Shows organization and relationship among clusters
- Visualize with a dendrogram
- Linkage criterion: Determines how to link or split clusters at each level
- There are two algorithms to construct hierarchical clusters

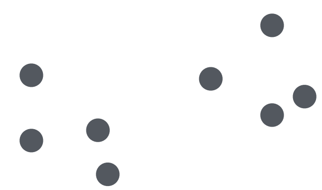
Add graphic that looks like a dendrogram. Epic intro

Yixin

- Multilevel hierarchy representing a group of data
- Shows organization and relationship among clusters with a dendrogram
- Two approaches: Divisive and Agglomerative
- Linkage criterion: Determines how to link or split clusters. Using centroid linkage involves taking the average of all points in a cluster and assigning that value as the mean centroid. Then the centroids with the shortest distances are merged together.

HIERARCHICAL
CLUSTERING

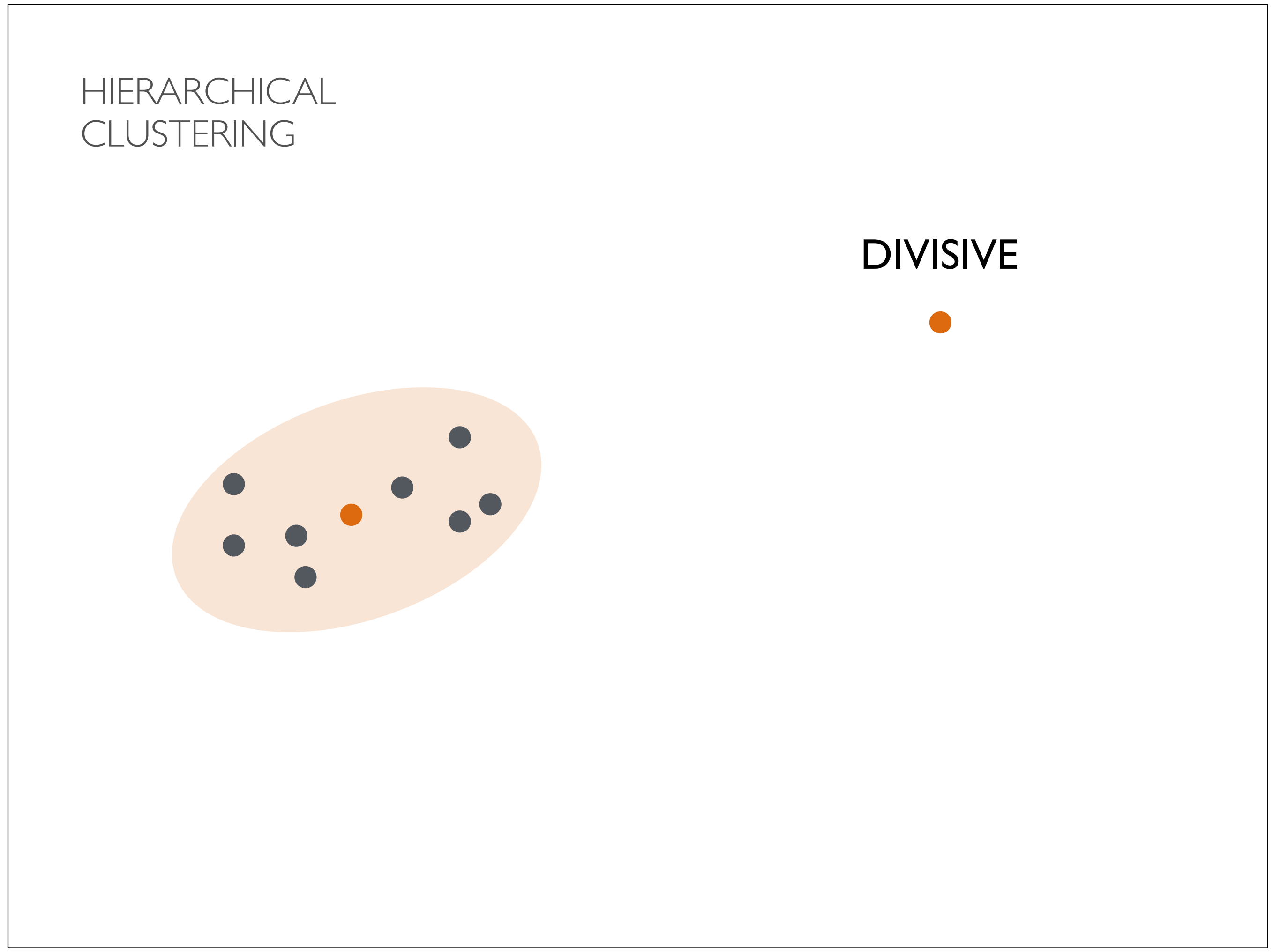
DIVISIVE



Zela

Divisive hierarchical clustering is thought of as a top down approach. This takes in a data set as one large cluster with a single centroid (animation). These nodes are important because they are the average representation of the points beneath it. We’ll keep track of nodes on the tree to the right. This, then branches out and splits the cluster into two more clusters with its own average representation until it reaches the individual observations.

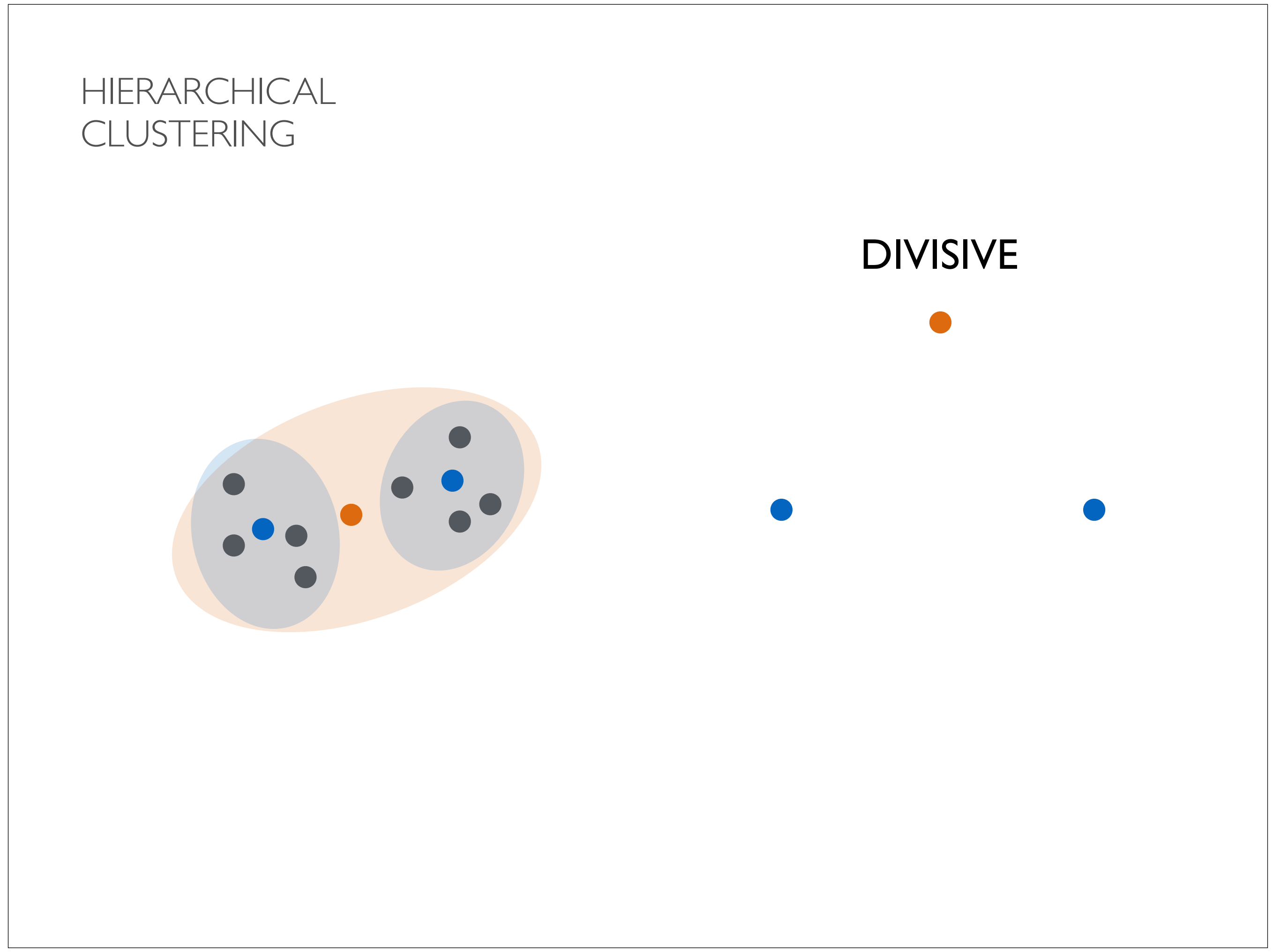
- Top Down Approach
- This takes in a data set as one large cluster with a single centroid. Then it recursively splits into two until it reaches the single points.
- Each centroid acts as a node



Zela

Divisive hierarchical clustering is thought of as a top down approach. This takes in a data set as one large cluster with a single centroid (animation). These nodes are important because they are the average representation of the points beneath it. We'll keep track of nodes on the tree to the right. This, then branches out and splits the cluster into two more clusters with its own average representation until it reaches the individual observations.

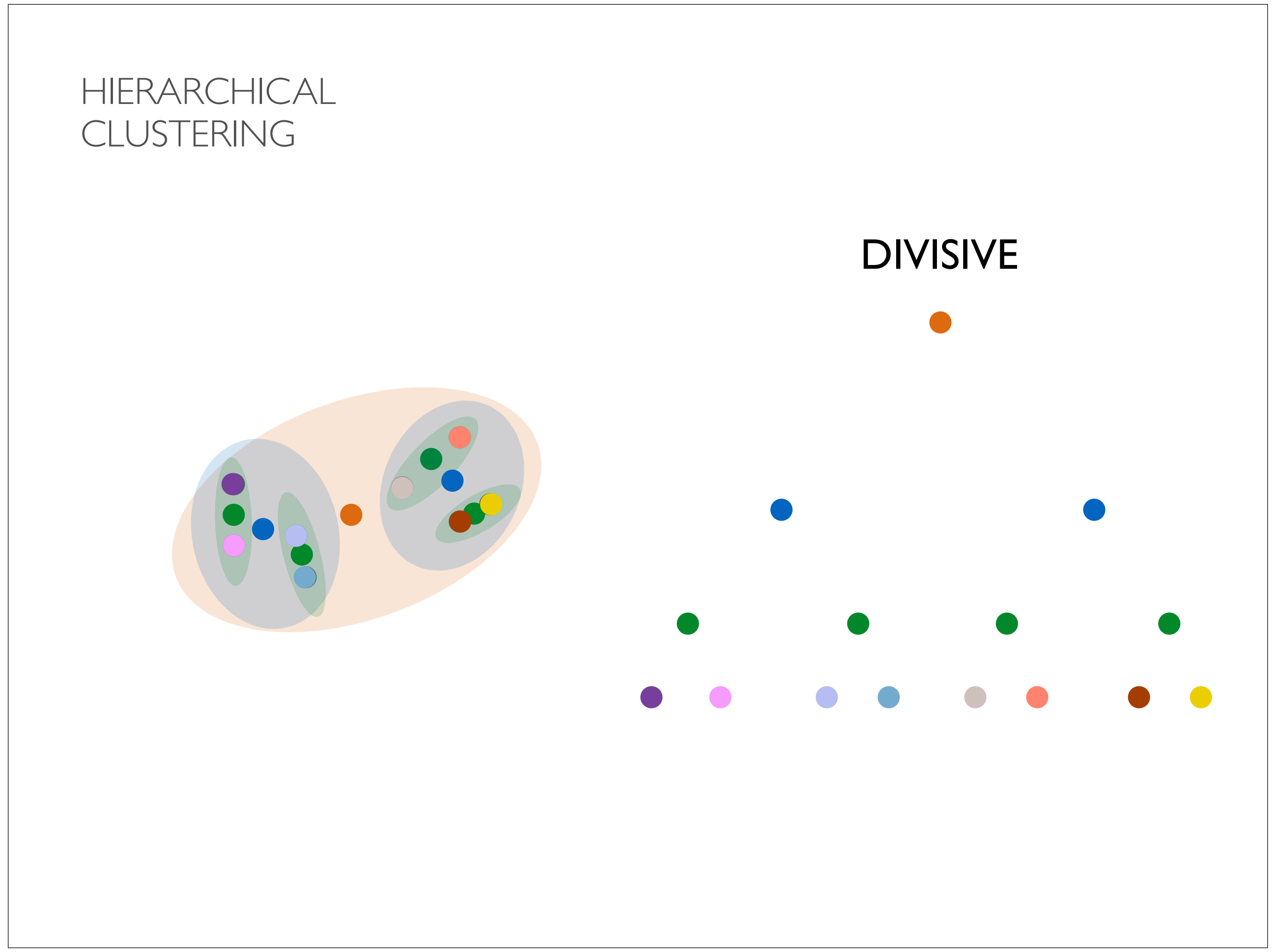
- Top Down Approach
- This takes in a data set as one large cluster with a single centroid. Then it recursively splits into two until it reaches the single points.
- Each centroid acts as a node



Zela

Divisive hierarchical clustering is thought of as a top down approach. This takes in a data set as one large cluster with a single centroid (animation). These nodes are important because they are the average representation of the points beneath it. We'll keep track of nodes on the tree to the right. This, then branches out and splits the cluster into two more clusters with its own average representation until it reaches the individual observations.

- Top Down Approach
- This takes in a data set as one large cluster with a single centroid. Then it recursively splits into two until it reaches the single points.
- Each centroid acts as a node



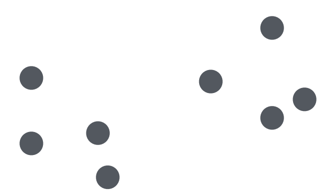
Zela

Divisive hierarchical clustering is thought of as a top down approach. This takes in a data set as one large cluster with a single centroid (animation). These nodes are important because they are the average representation of the points beneath it. We'll keep track of nodes on the tree to the right. This, then branches out and splits the cluster into two more clusters with its own average representation until it reaches the individual observations.

- Top Down Approach
- This takes in a data set as one large cluster with a single centroid. Then it recursively splits into two until it reaches the single points.
- Each centroid acts as a node

HIERARCHICAL
CLUSTERING

AGGLOMERATIVE

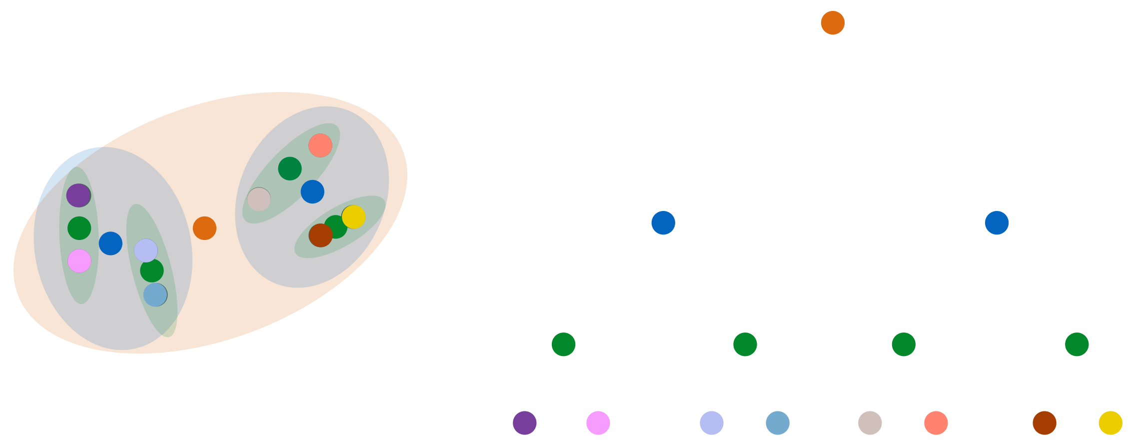


Zela

Agglomerative hierarchical clustering is similar but approach the problem bottom up. It initially looks at each data point as its own cluster. Then it clusters all the data into groups of two, which form the next layer. The means of those clusters become the data points of the next layer. This continues until the entire dataset is represented by a single average centroid. Now we’re going to compare the two methods.

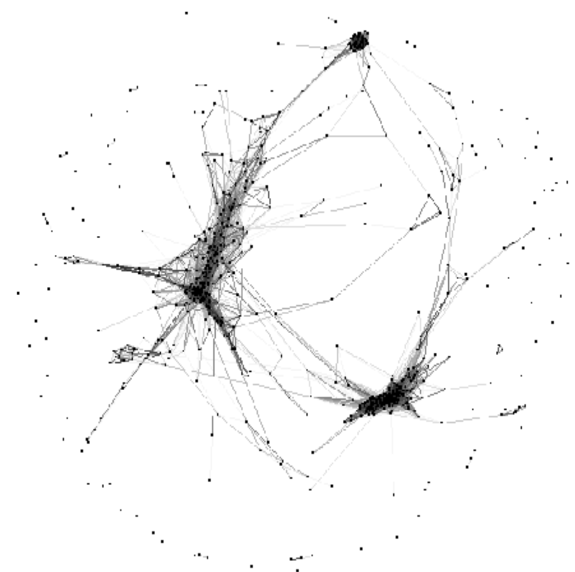
HIERARCHICAL
CLUSTERING

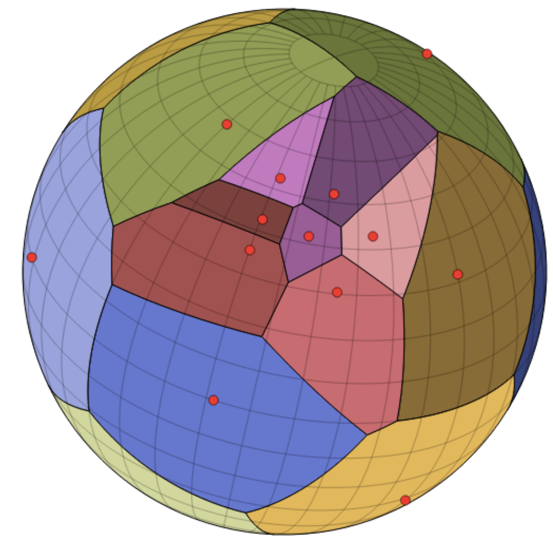
AGGLOMERATIVE



Zela
Agglomerative hierarchical clustering is similar but approach the problem bottom up. It initially looks at each data point as its own cluster. Then it clusters all the data into groups of two, which form the next layer. The means of those clusters become the data points of the next layer. This continues until the entire dataset is represented by a single average centroid. Now we're going to compare the two methods.

HIERARCHICAL
CLUSTERING

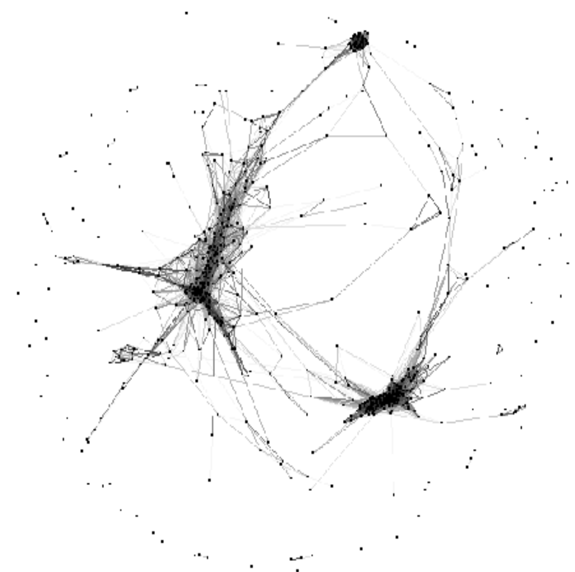


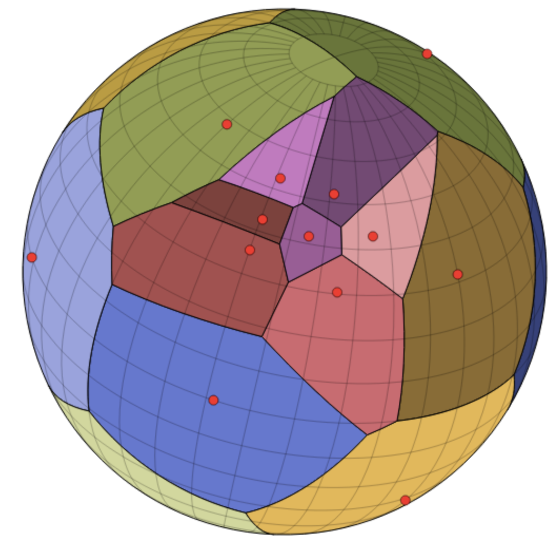


Yixin

- We wrote both hierarchical clustering algorithms
- We tested them, as well as a built-in MATLAB one, on standard datasets
- Thousands of data points which live on 13,000 dimensional
- Illustration of tree structure with means
- Example Voronoi diagram: which points are clustered with which means

HIERARCHICAL
CLUSTERING





Yixin

- We wrote both hierarchical clustering algorithms
- We tested them, as well as a built-in MATLAB one, on standard datasets
- Thousands of data points which live on 13,000 dimensional
- Illustration of tree structure with means
- Example Voronoi diagram: which points are clustered with which means

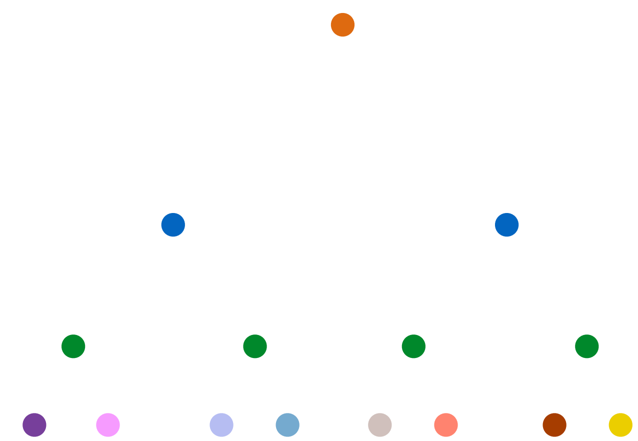
ALGORITHM ANALYSIS

$$T_{div}(n) = \Theta(n \log(n) di)$$

Yixin
k - means complexity
agglomerative
divisive

ALGORITHM ANALYSIS

$$T_{div}(n) = \Theta(n \log(n) di)$$



Yixin
k - means complexity
agglomerative
divisive

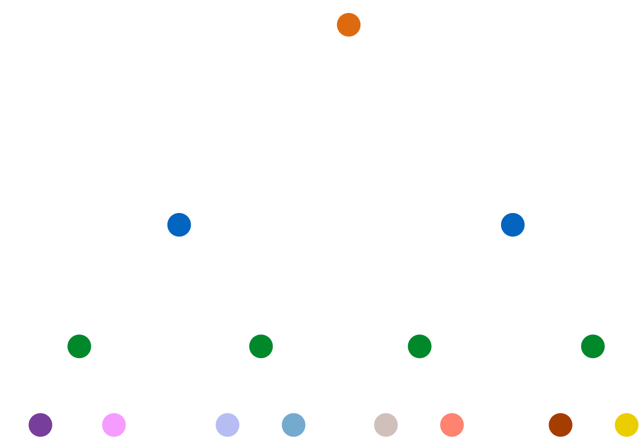
ALGORITHM ANALYSIS

$$T_{aggl}(n) = \Theta(n^2 di)$$

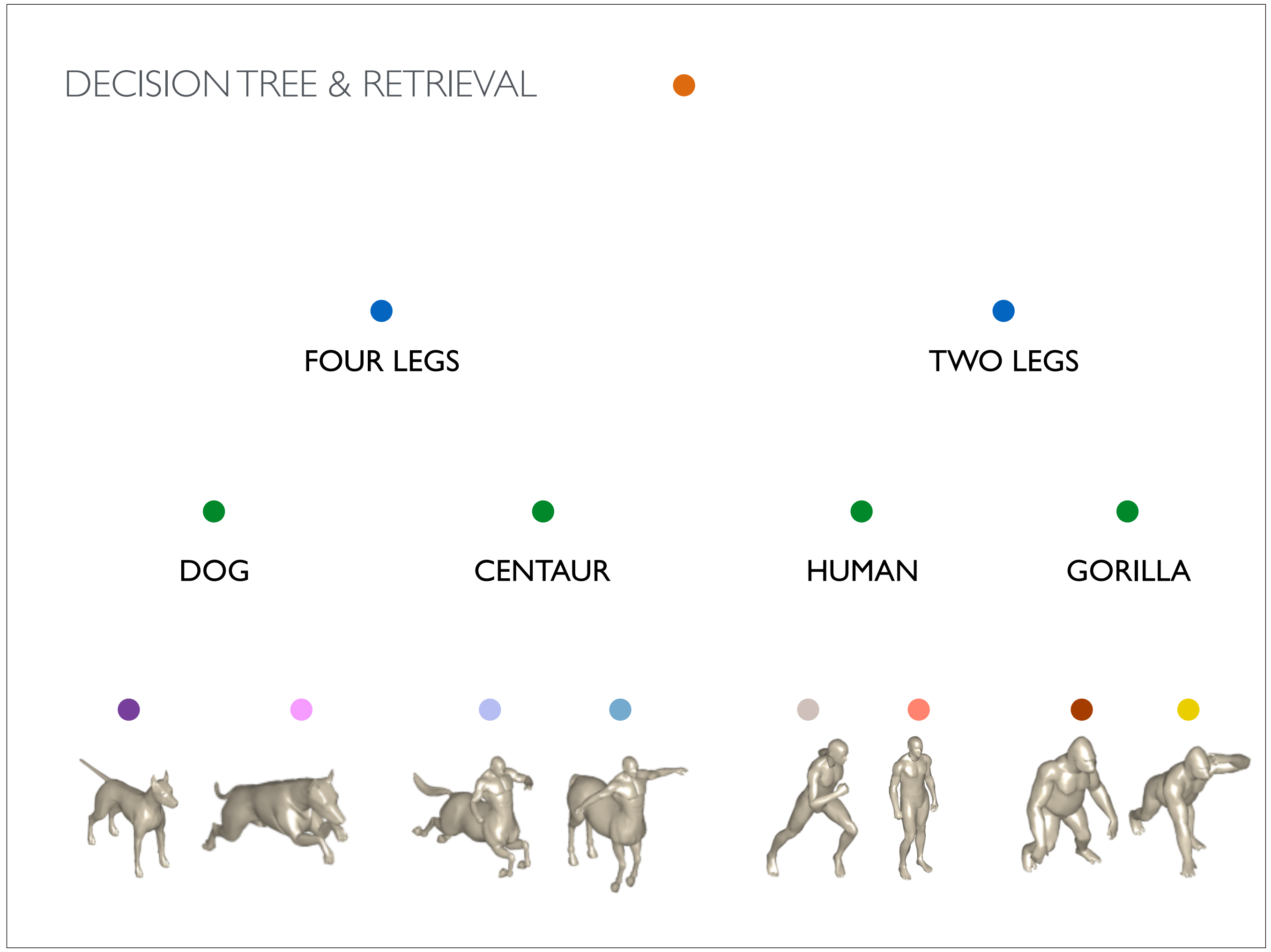
Yixin
k - means complexity
agglomerative
divisive

ALGORITHM ANALYSIS

$$T_{aggl}(n) = \Theta(n^2 di)$$



Yixin
k - means complexity
agglomerative
divisive



Zela

So how does all of this help us in our over all goal? Let's do a quick over view of whats happening. We we're given a data base, we have now clustered each shape in the data base. And when you give us a query, we can run it through the decision tree to give you back the most similar object as efficiently as possible. (Animation) a query shape comes in and begins to traverse the tree. Is the object closer to this point or this point? It will traverse again and ask is the object closer to this point or this point? And once it hits the leaves it will compare to the objects in the subtree and retrieve the closest one. And because this is a balanced tree, it takes log(n) time.

- The goal is to figure out what the closest points are, efficiently
- For each internal node in the tree, find the closest subtree mean (using spherical distance, aka Karcher mean)
- Once you get to a leaf, you have the "closest" object, but in log(n) time

log(amount of data points) 2^n = h

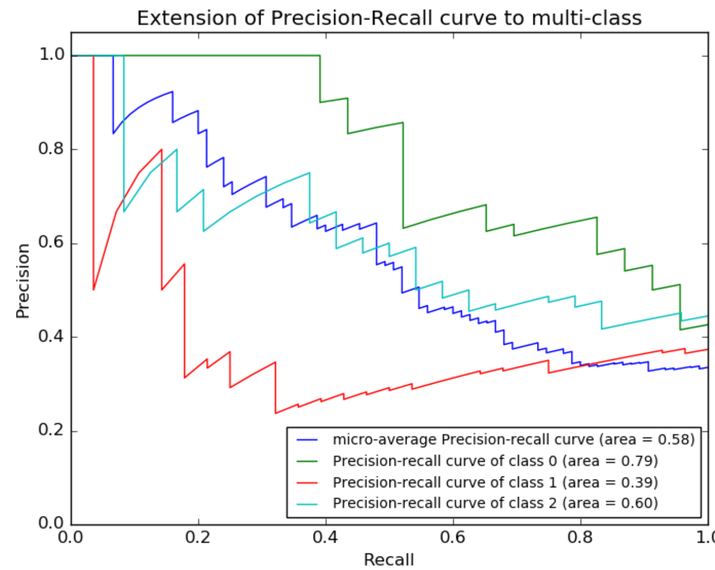
h = log2(n)

Yixin

Retrieval implementation

- How do we grab multiple results? Retrieval, not classification
- Use a recursive mergesort-like algorithm (except not really)
    - At each level, return the concatenation of the closer and the farther subproblems

METRICS



Zela

- Precision recall
- Precision recall curve
  - AP
  - MAP
- DCG: The intuition behind Discounted Cumulative Gain is that in information retrieval a user only investigates about the first few pages of retrieved objects and users are less likely to go beyond the first page. So if you retrieve a relevant object that is beyond the first page, it is less likely it will be viewed; therefore it will be counted less to the overall score.
  - NN: Only looks at the first retrieved object

RESULTS

Yixin

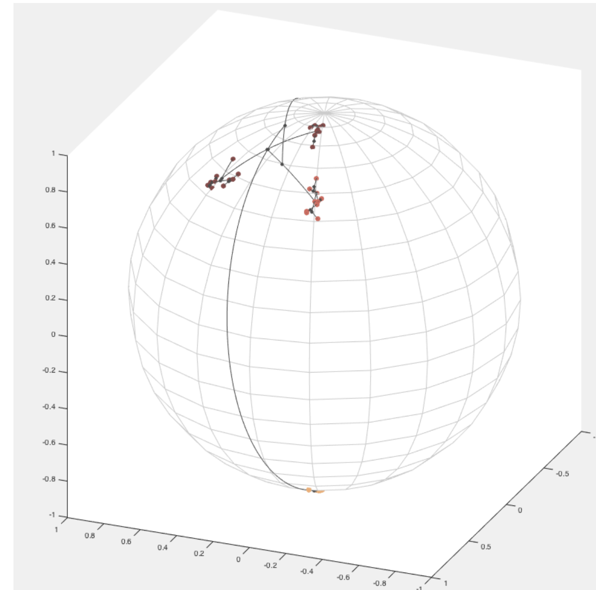
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don’t overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS



Yixin

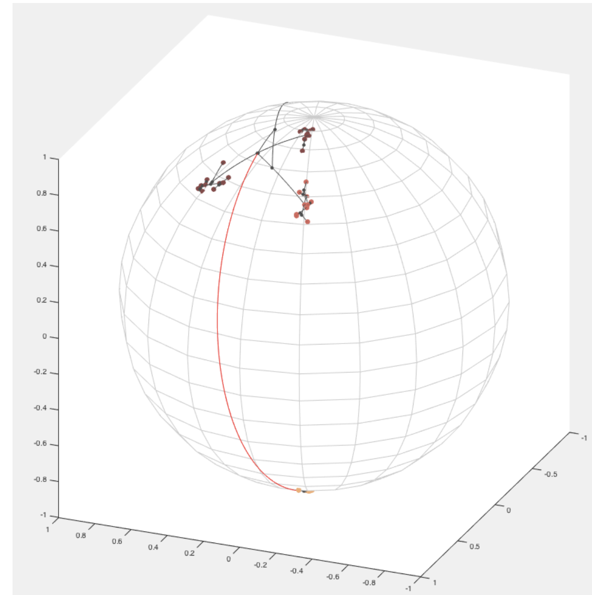
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don’t overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS

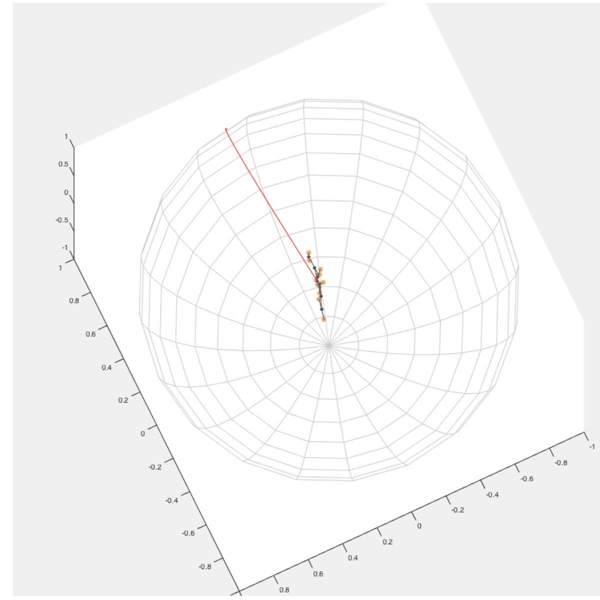


Yixin

- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don’t overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin
Clustering Graphics
Explain how well our approach worked

RESULTS



Yixin

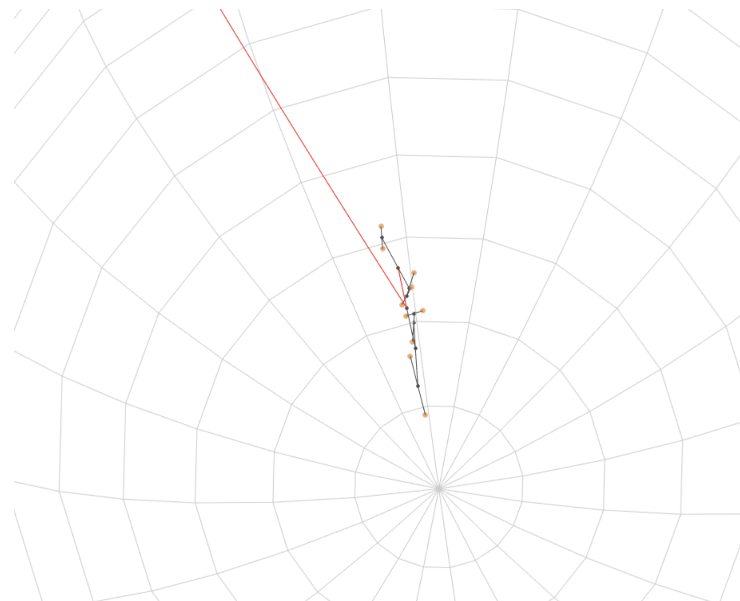
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don’t overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS



Yixin

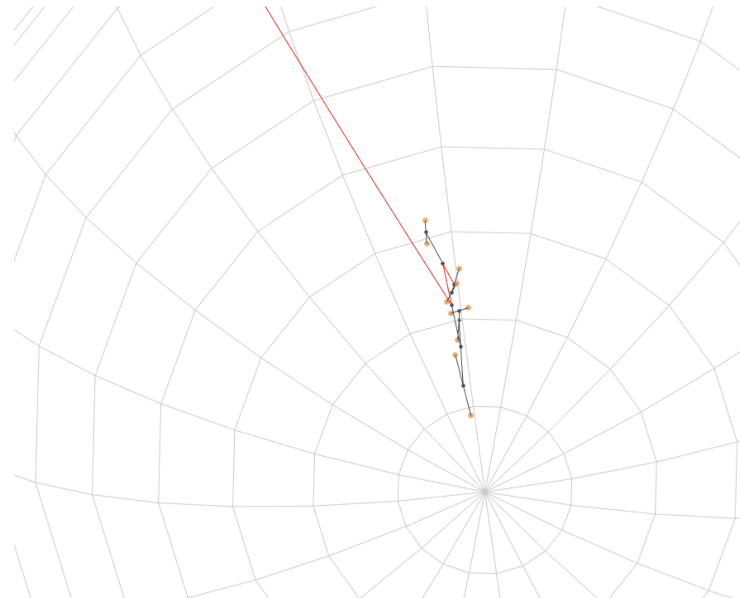
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don't overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS



Yixin

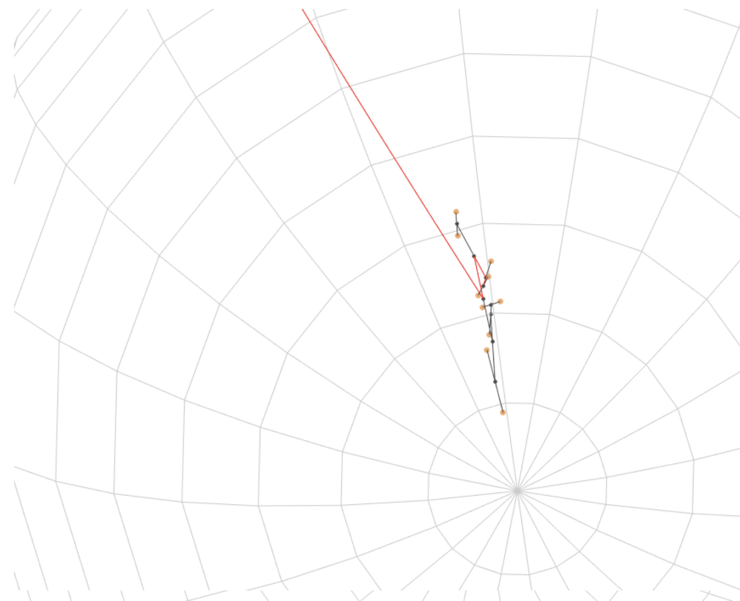
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don't overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS



Yixin

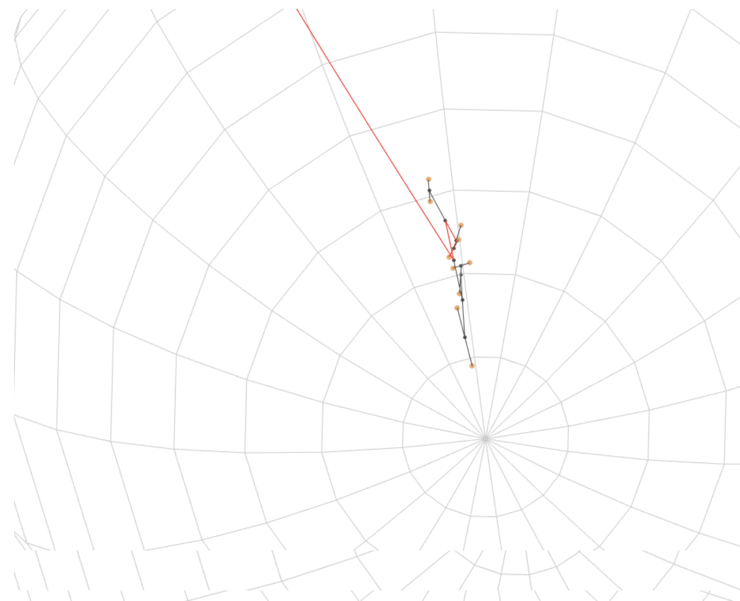
- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don't overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

RESULTS



Yixin

- First we randomly constructed spherical clusters using the von Mises distribution (generalization of the normal distribution on a circle)
- Depending on the kappa value (concentration parameter), it performed quite well
- When clusters don’t overlap, it gets nearly perfect scores
- Then we ran it on common datasets
- Very good at nearest neighbors (close to 1)
- Not so good on some of the other metrics (outperformed by state of the art)
- Clusters overlap in higher dimensions, explore different ways of fitting the structure of the data

Yixin

Clustering Graphics

Explain how well our approach worked

CONCLUSION

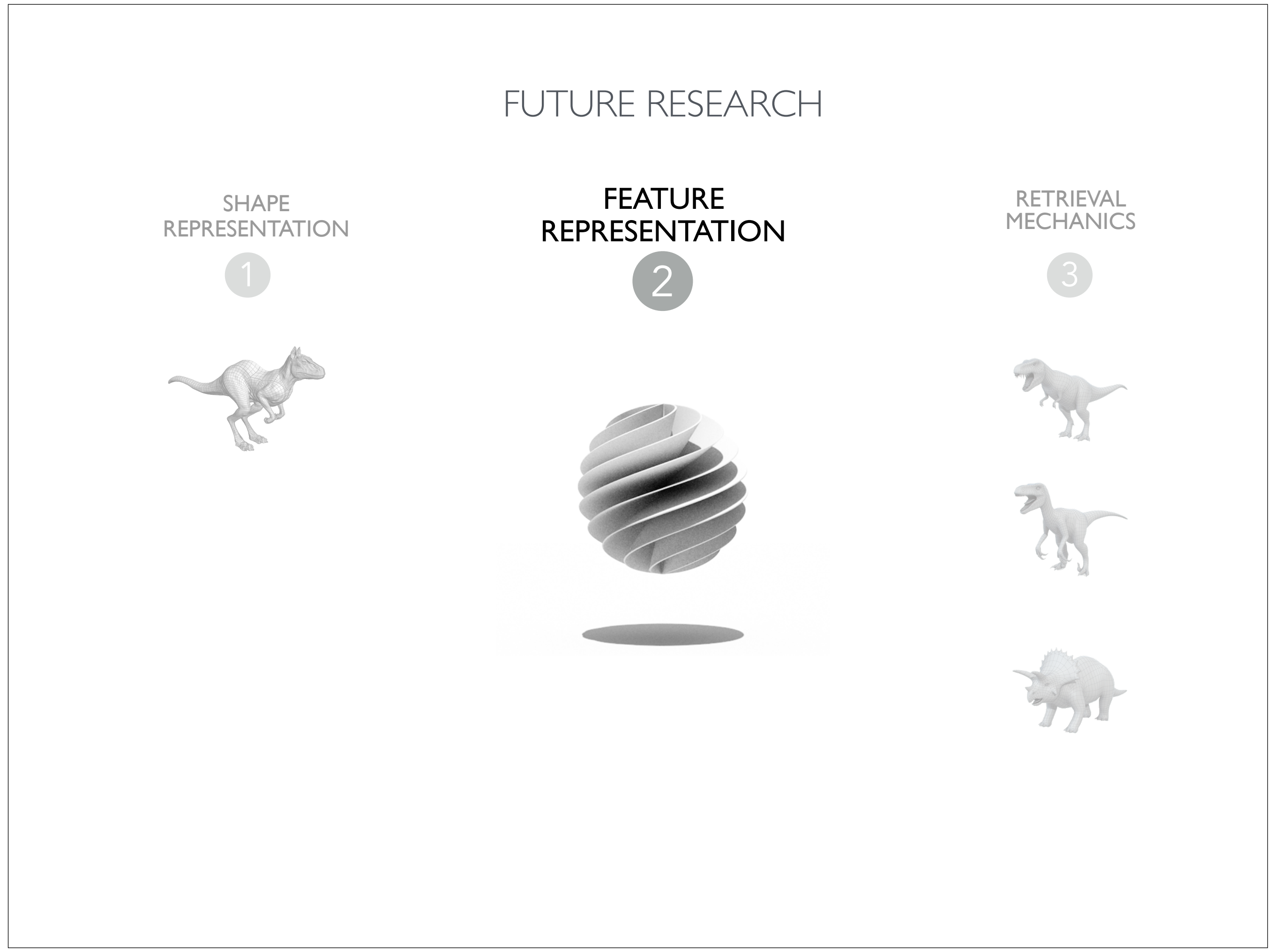
Zixin

- We’ve explored the significant problem of shape analysis and retrieval, important for many applications including 3d animation and design and robot and computer vision
- We’ve implemented a faster retrieval data structure and retrieval mechanism that runs in O(log(n)) instead of O(n)
- We proved the running time of hierarchical clustering using spherical K-means

CONCLUSION

Zixin

- We've explored the significant problem of shape analysis and retrieval, important for many applications including 3d animation and design and robot and computer vision
- We've implemented a faster retrieval data structure and retrieval mechanism that runs in O(log(n)) instead of O(n)
- We proved the running time of hierarchical clustering using spherical K-means



Zixin

- Delve deeper into the math behind feature representation (Laplace-Beltrami operator, wavelet density estimation)
- Optimize the wavelet density estimation code
- Linear assignment of wavelet coefficients: how can we “distort” or manipulate the feature representation so that close shapes are closer together?
- Looking forward to results

THANK YOU

HIERARCHICAL SHAPE RETRIEVAL on the UNIT HYPERSPHERE

Yixin Lin
Glizela Taino
Mark Moyou
Adrian M. Peter

HIERARCHICAL SHAPE RETRIEVAL on a UNIT HYPERSPHERE

Yixin Lin
Glizela Taino
Mark Moyou
Adrian M. Peter