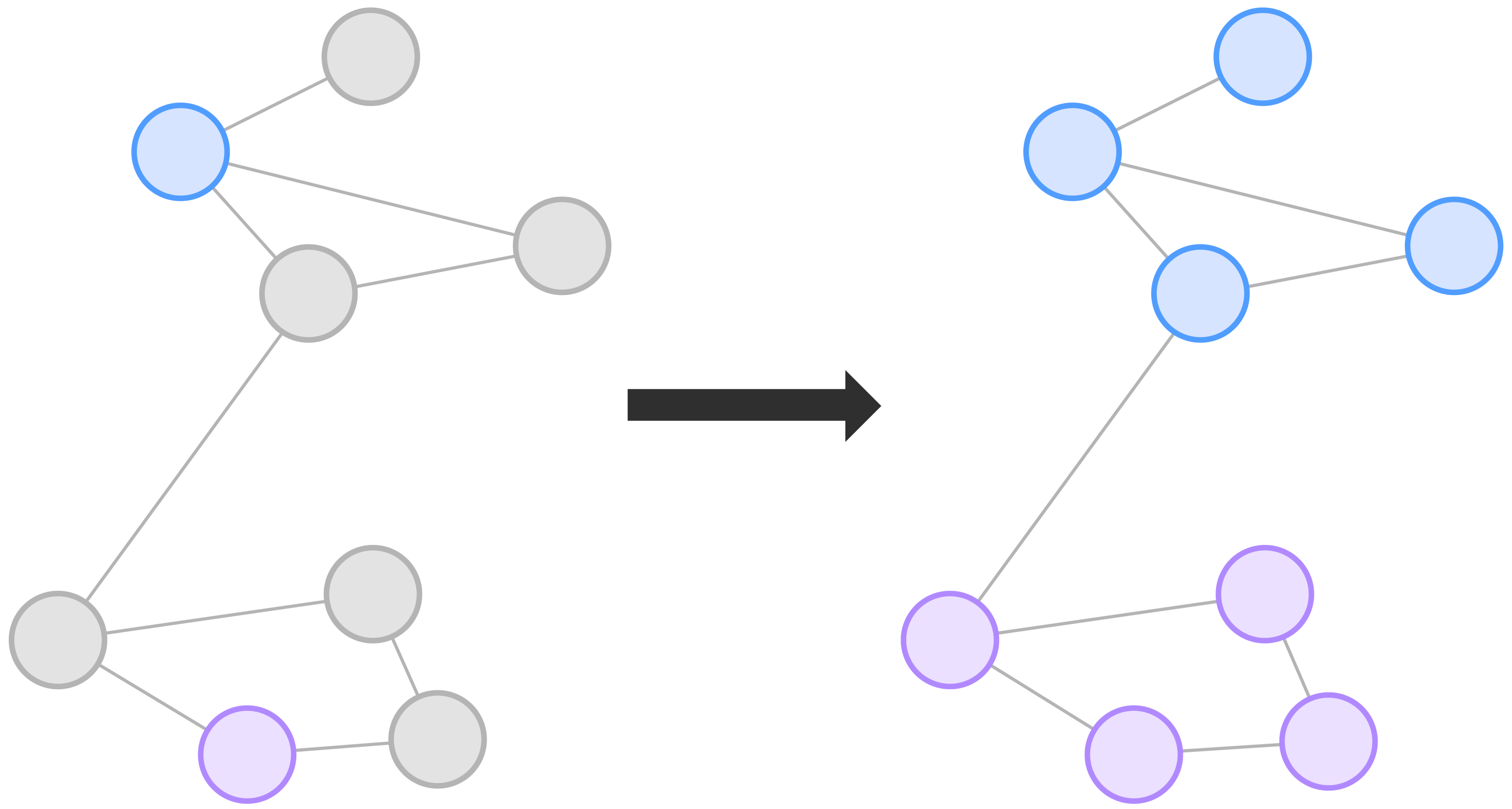
Semi-supervised Learning

DSCC 251/451: Machine Learning with Limited Data

C.M. Downey

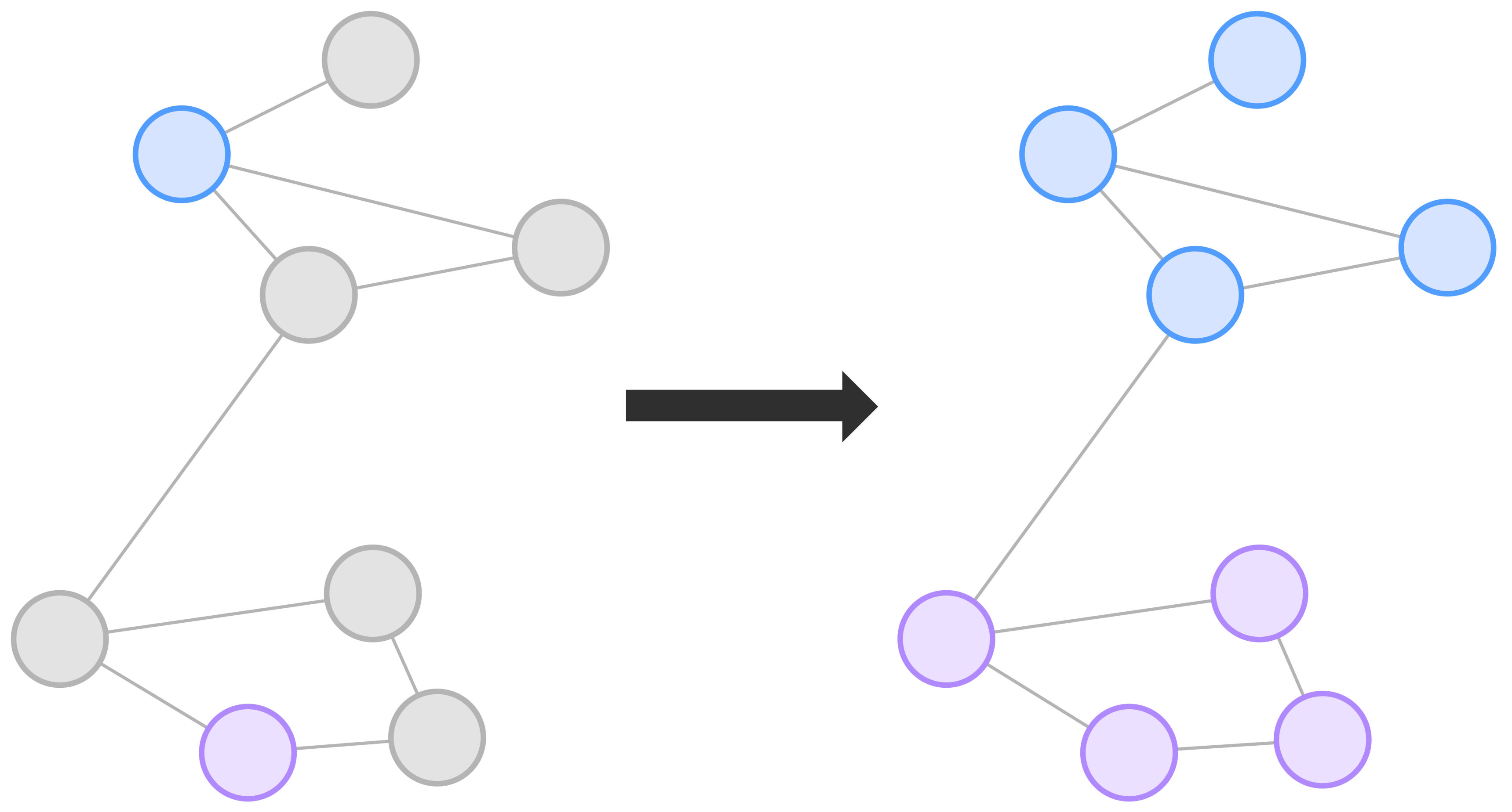
Spring 2026

Roadmap



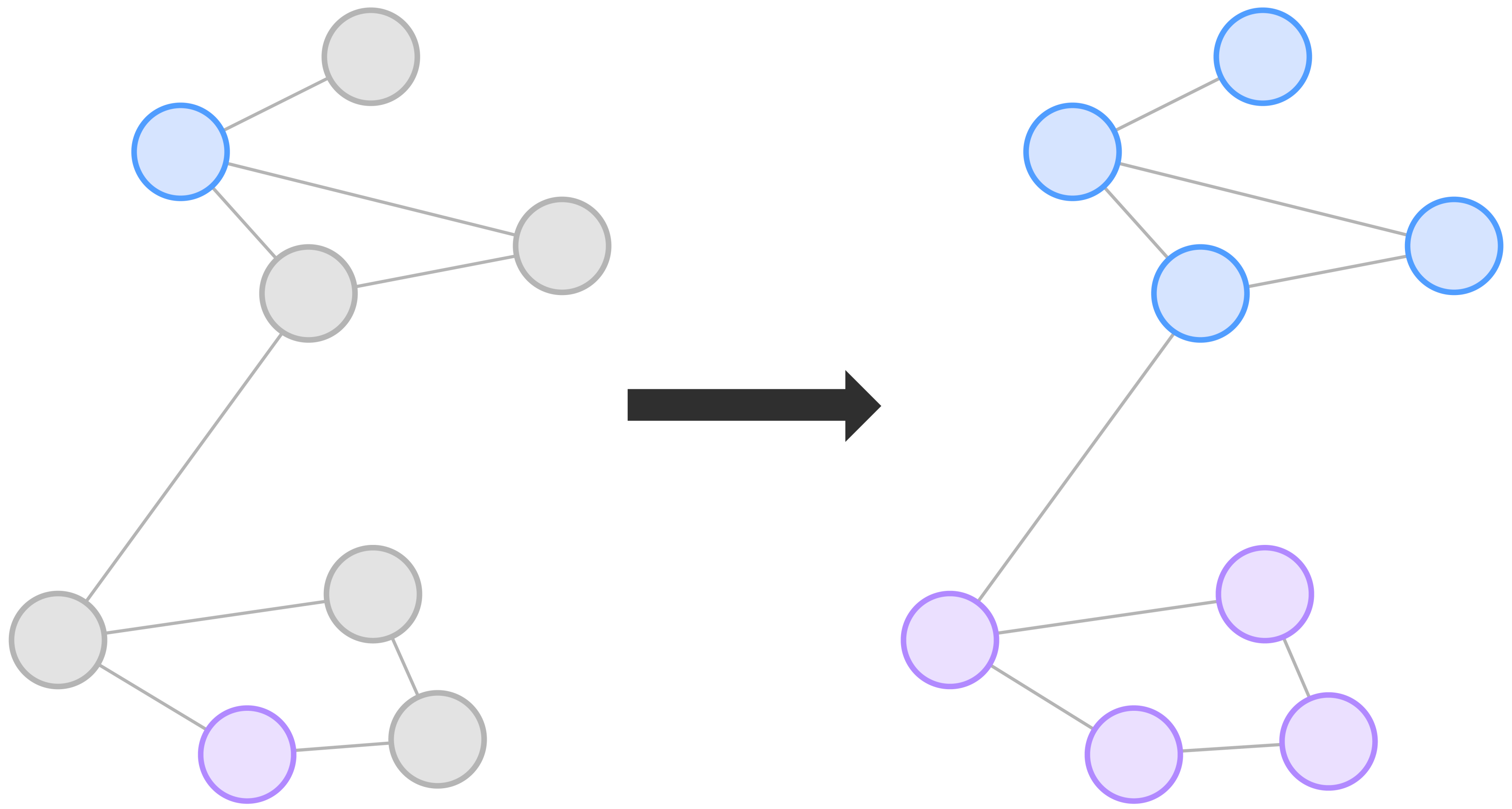
Roadmap

- Last few lectures: what can you do **without data labels?**



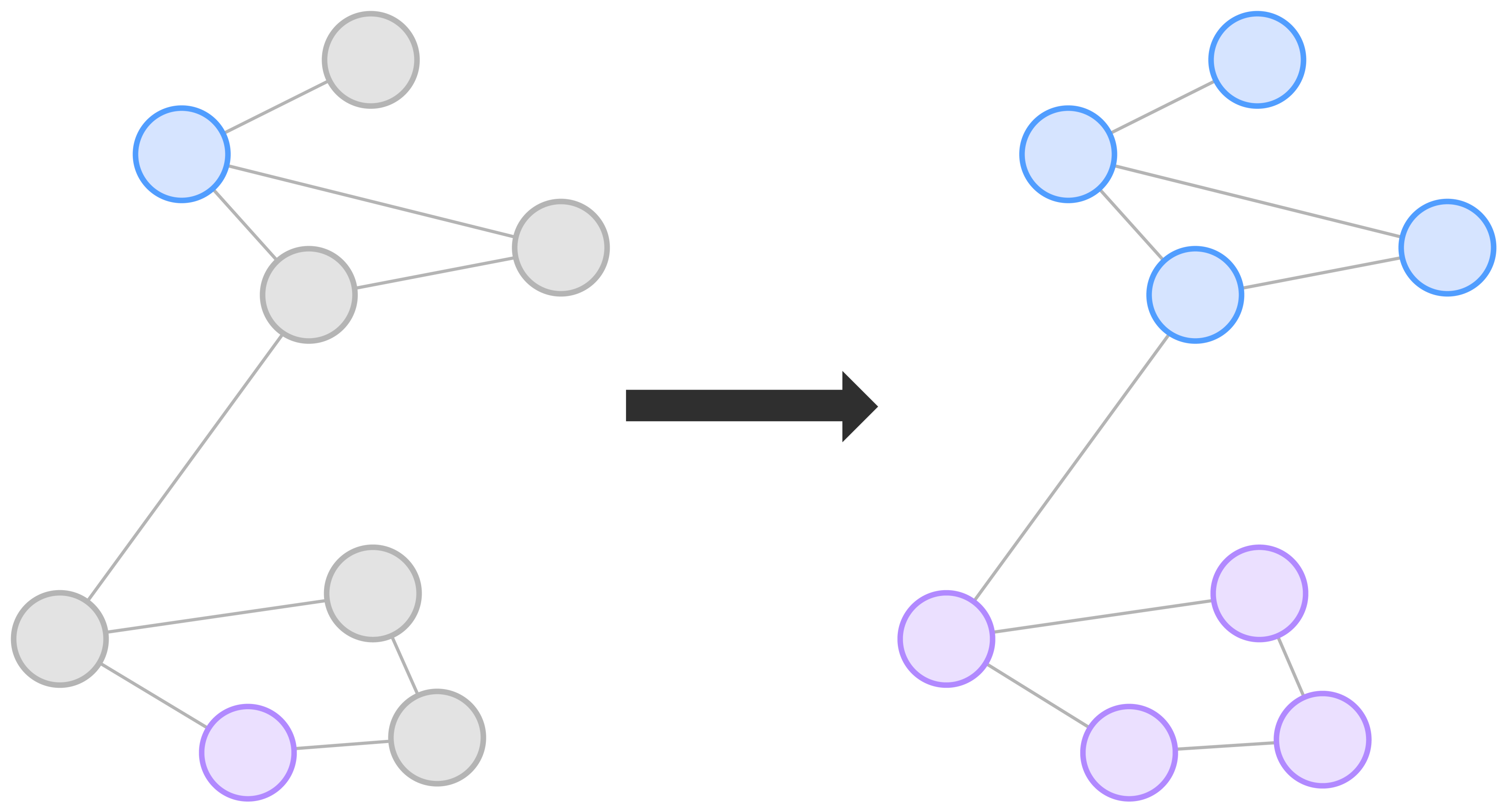
Roadmap

- Last few lectures: what can you do **without data labels?**
- Semi-supervised Learning: what can you do with a **combination** of labeled and unlabeled data?



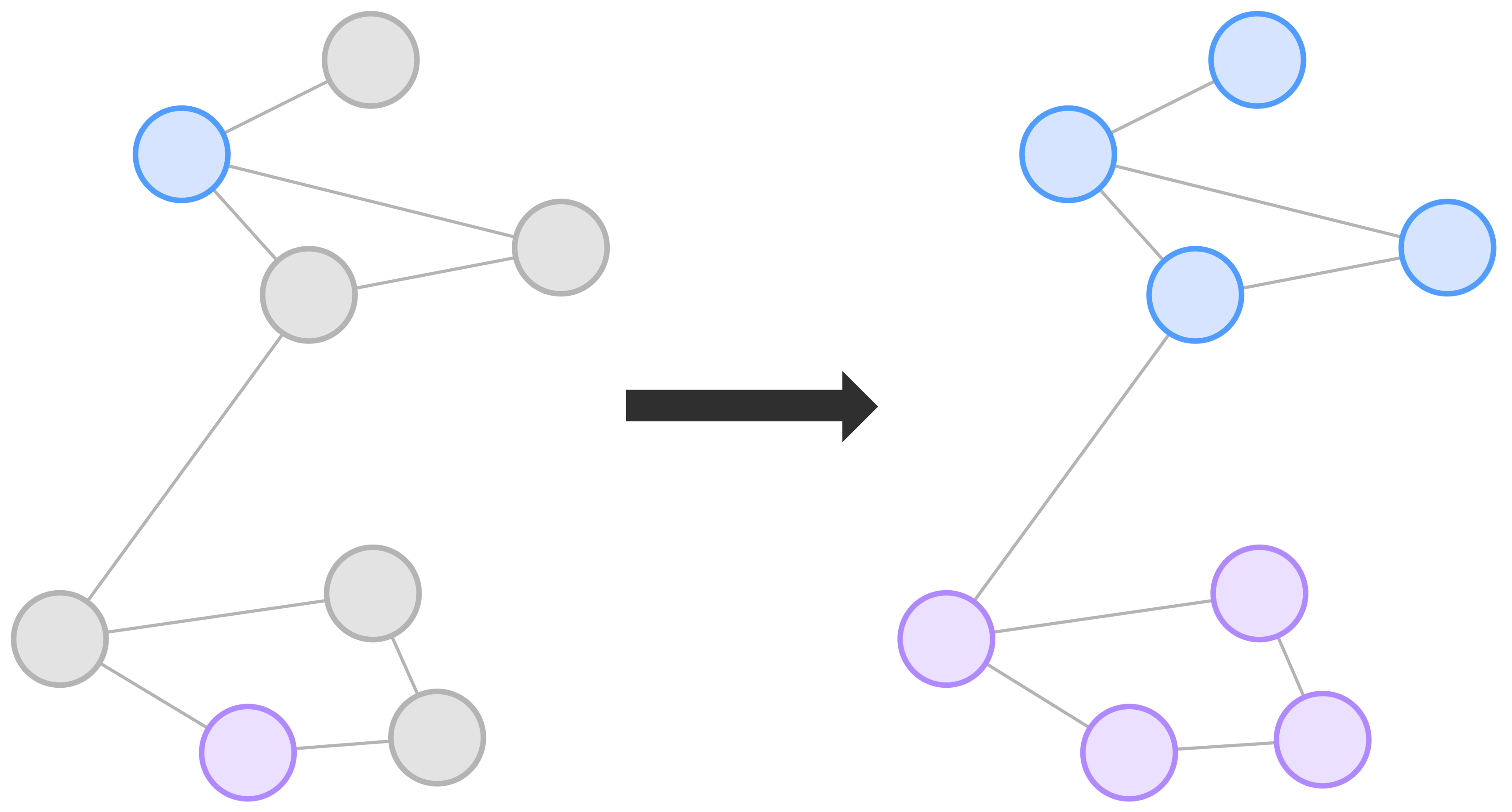
Roadmap

- Last few lectures: what can you do **without data labels?**
- Semi-supervised Learning: what can you do with a **combination** of labeled and unlabeled data?
    - Usually assumed we have **much more unlabeled data** than labeled



Roadmap

- Last few lectures: what can you do **without data labels?**
- Semi-supervised Learning: what can you do with a **combination** of labeled and unlabeled data?
  - Usually assumed we have **much more unlabeled data** than labeled
  - Relies on **several key assumptions** about the **input space** (X)



The Scenario

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m >> n)$: **numerous and cheap**

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m >> n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m > > n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**
- Example: **classify clinical notes** into those indicating cancer vs. not

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m >> n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**
- Example: **classify clinical notes** into those indicating cancer vs. not
  - Labeled: 200 notes manually reviewed by physician

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m > > n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**
- Example: **classify clinical notes** into those indicating cancer vs. not
  - Labeled: 200 notes manually reviewed by physician
  - Unlabeled: 50,000 notes in the database

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m >> n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**
- Example: **classify clinical notes** into those indicating cancer vs. not
  - Labeled: 200 notes manually reviewed by physician
  - Unlabeled: 50,000 notes in the database
  - **Purely supervised**: train on 200 labeled examples only

The Scenario

- You have n labeled examples (x_i, y_i): **few and expensive**
- You have m unlabeled examples x_j $(m >> n)$: **numerous and cheap**
- We want to **do better than ignoring the unlabeled examples**
- Example: **classify clinical notes** into those indicating cancer vs. not
  - Labeled: 200 notes manually reviewed by physician
  - Unlabeled: 50,000 notes in the database
  - **Purely supervised**: train on 200 labeled examples only
  - **Semi-supervised**: make use of the 50k other notes

Difference from what we've seen

Method	Labeled?	Unlabeled?	Goal
Supervised	Yes (many)	No	Learn classifier
Unsupervised	No	Yes	Structure discovery
Self-supervised	No	Yes	Representation learning
Semi-supervised	Yes (few)	Yes (many)	Learn classifier, using both

Difference from what we've seen

- Unsupervised: learn structure **without knowing downstream task**

Method	Labeled?	Unlabeled?	Goal
Supervised	Yes (many)	No	Learn classifier
Unsupervised	No	Yes	Structure discovery
Self-supervised	No	Yes	Representation learning
Semi-supervised	Yes (few)	Yes (many)	Learn classifier, using both

Difference from what we've seen

- Unsupervised: learn structure **without knowing downstream task**
- Self-supervised: essentially **ignore labels** during pre-training
  - (only use for **task fine-tuning**)

Method	Labeled?	Unlabeled?	Goal
Supervised	Yes (many)	No	Learn classifier
Unsupervised	No	Yes	Structure discovery
Self-supervised	No	Yes	Representation learning
Semi-supervised	Yes (few)	Yes (many)	Learn classifier, using both

Difference from what we've seen

- Unsupervised: learn structure **without knowing downstream task**
- Self-supervised: essentially **ignore labels** during pre-training
  - (only use for **task fine-tuning**)
- Semi-supervised: labels and unlabeled data **used jointly**
  - Labels inform **what sort of latent structure matters**

Method	Labeled?	Unlabeled?	Goal
Supervised	Yes (many)	No	Learn classifier
Unsupervised	No	Yes	Structure discovery
Self-supervised	No	Yes	Representation learning
Semi-supervised	Yes (few)	Yes (many)	Learn classifier, using both

Key Assumptions of Semi-supervised Learning

Structure Assumptions

Structure Assumptions

- Semi-supervised Learning relies on **key assumptions** about the **data structure**
  - Without these, **no good reason** to think it will work!

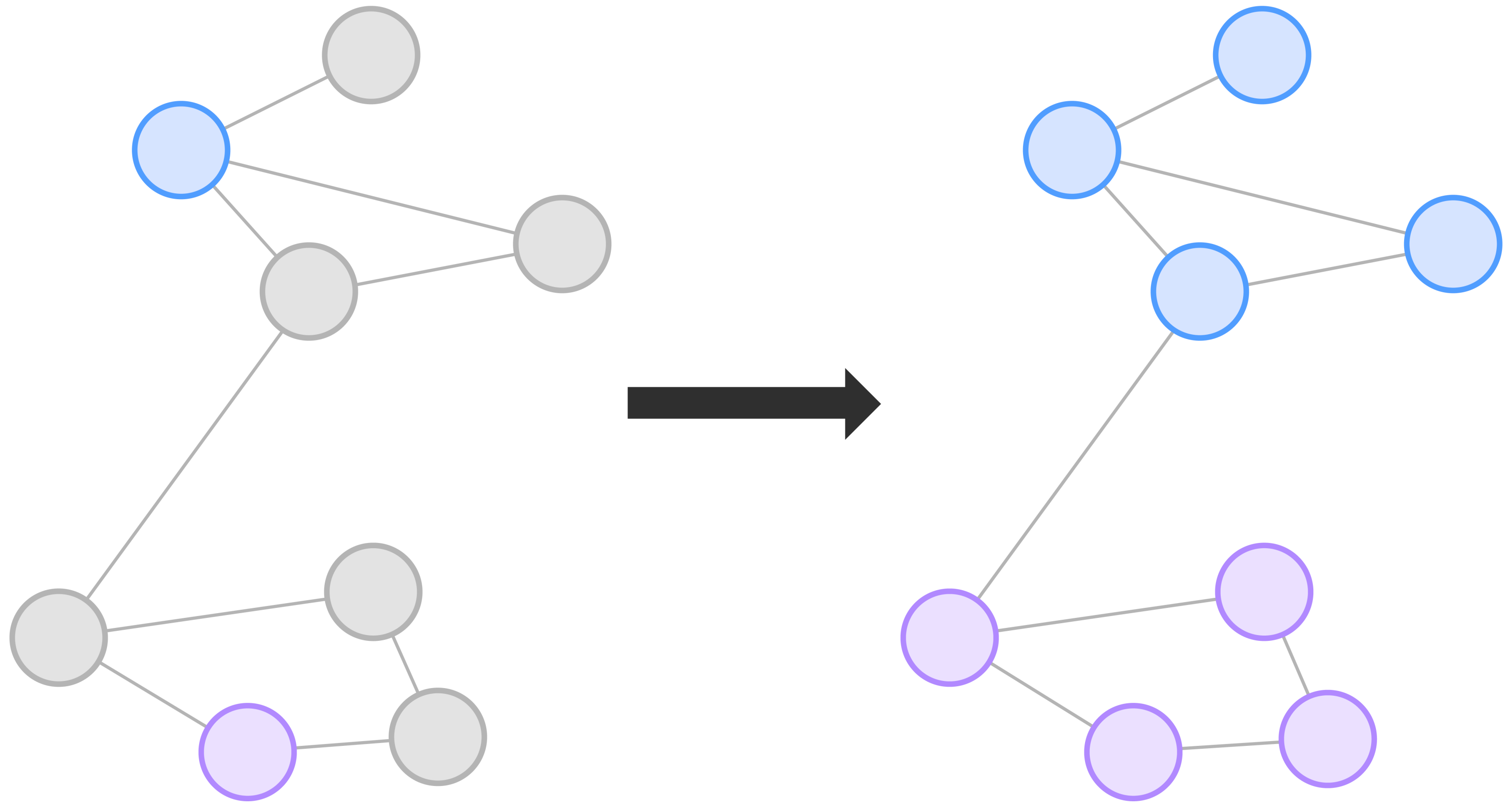
Structure Assumptions

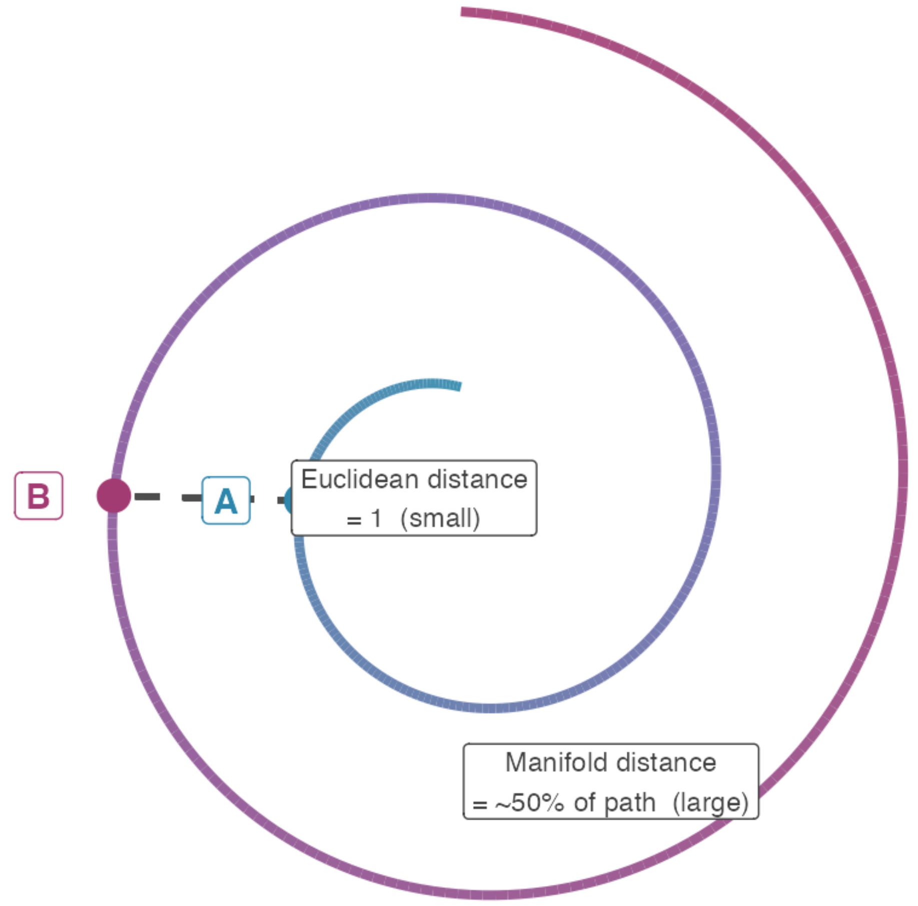
- Semi-supervised Learning relies on **key assumptions** about the **data structure**
  - Without these, **no good reason** to think it will work!
- Formally: unlabeled data tells you about $P(X)$, supervised learning is about $P(Y|X)$
  - Does knowing $P(X)$ **actually help** with $P(Y|X)$?

Structure Assumptions

- Semi-supervised Learning relies on **key assumptions** about the **data structure**
  - Without these, **no good reason** to think it will work!
- Formally: unlabeled data tells you about $P(X)$, supervised learning is about $P(Y|X)$
  - Does knowing $P(X)$ **actually help** with $P(Y|X)$?
- **Three classic assumptions** (Chapelle et al., 2006)
  - **Smoothness:** nearby points have the same label
  - **Cluster:** points in the same cluster should have the same label
  - **Manifold:** data lies on a low-dimensional manifold that can be learned

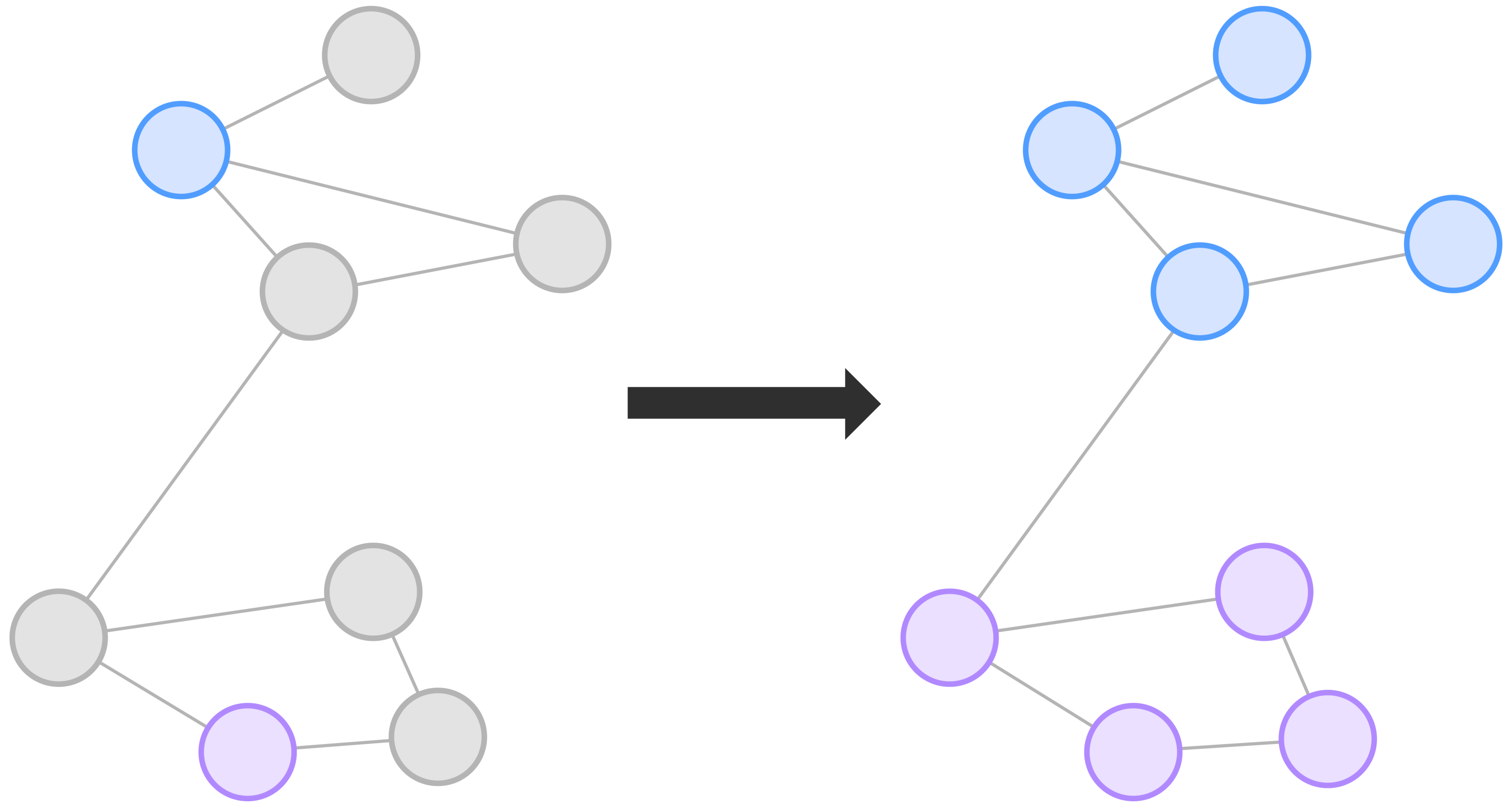
Smoothness Assumption

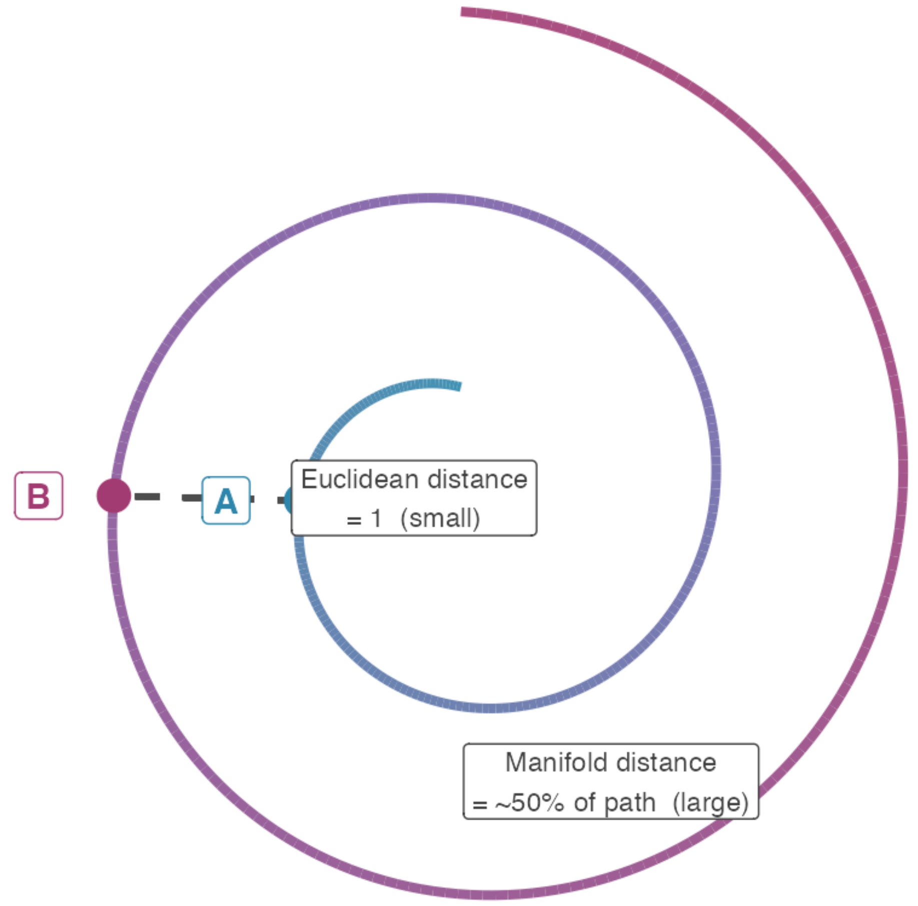




Smoothness Assumption

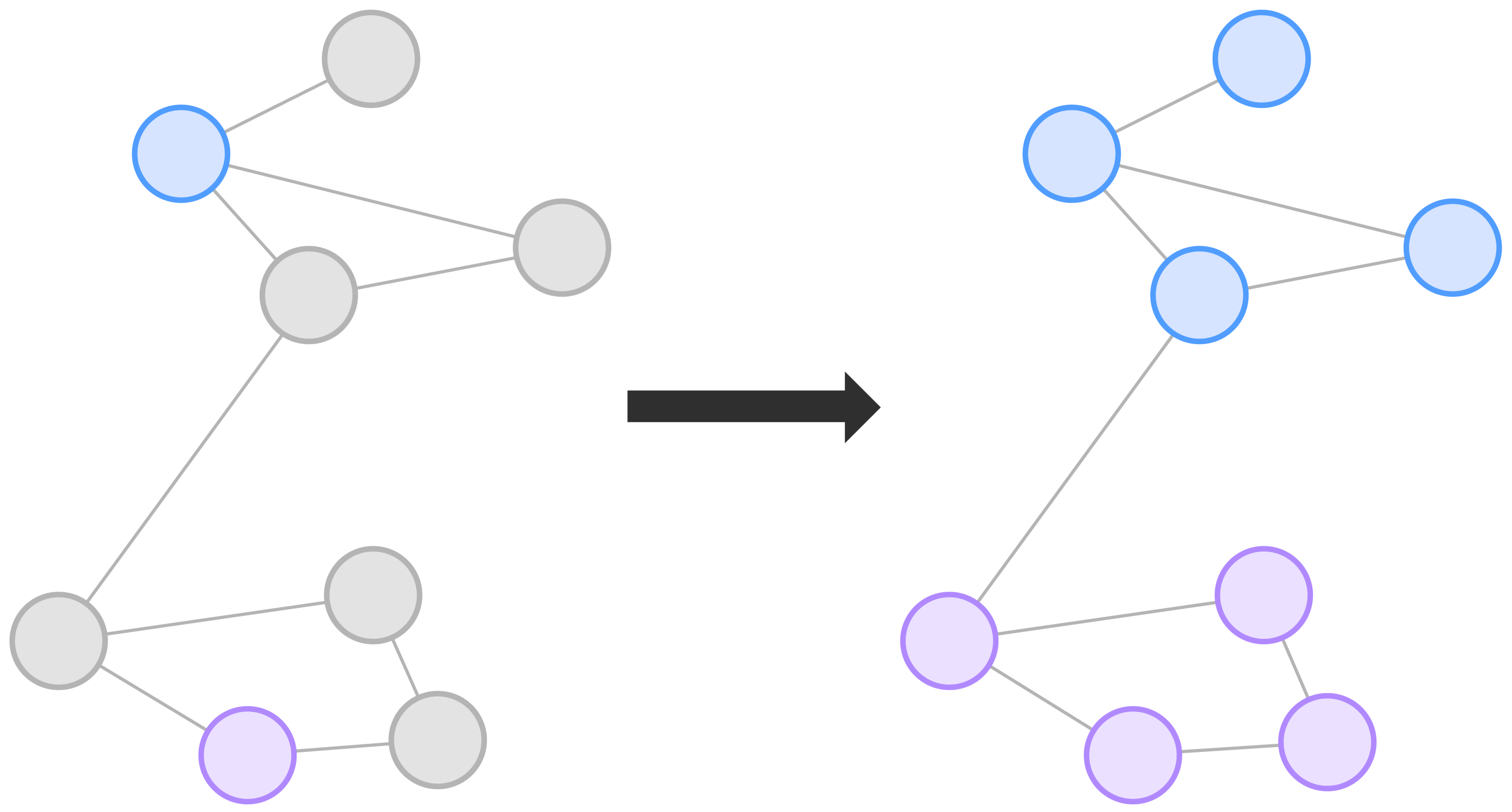
- Idea: if x_i and x_j are **similar**, their labels should be the **same**
  - Labeled data gives you **anchors**
  - Unlabeled data lets you **fill in the space** between anchors

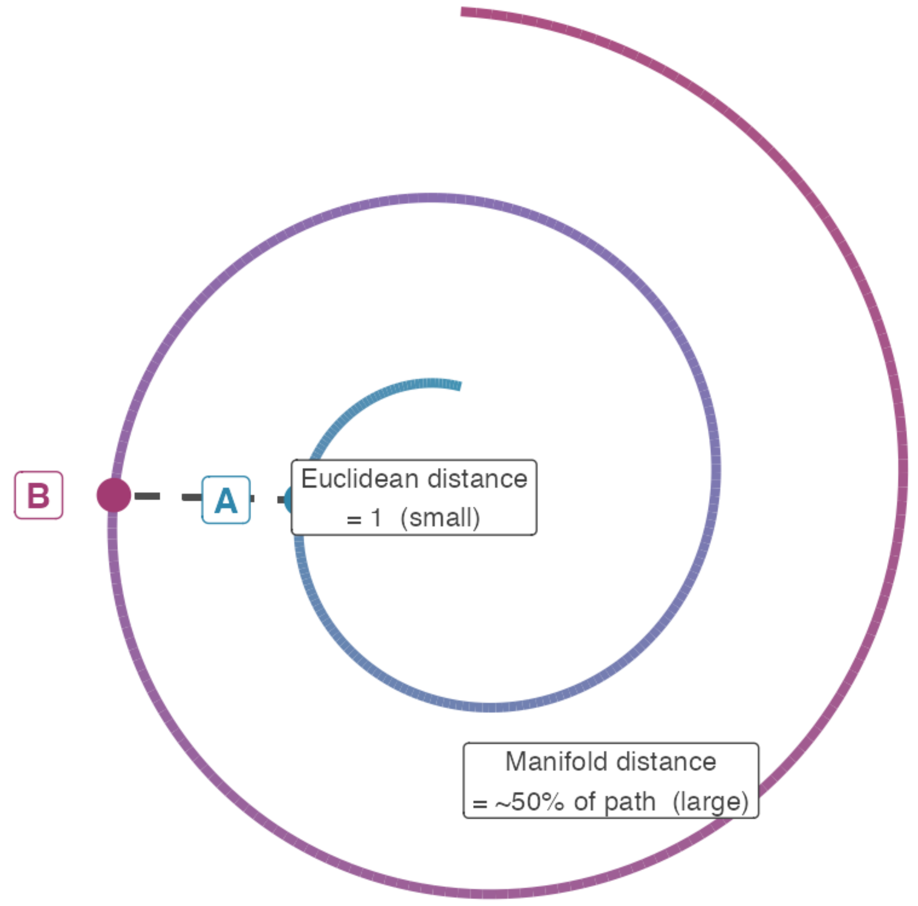




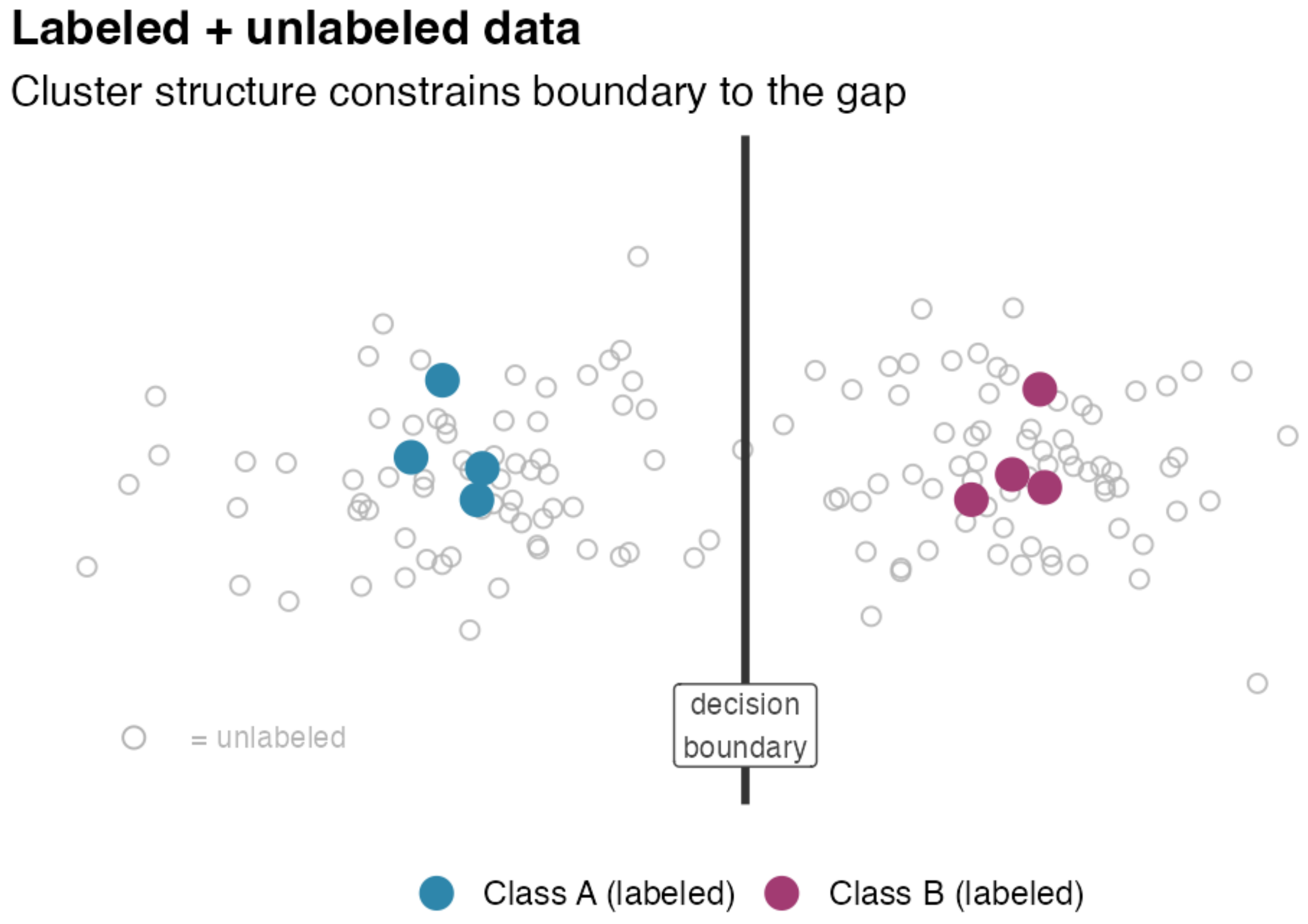
Smoothness Assumption

- Idea: if x_i and x_j are **similar**, their labels should be the **same**
  - Labeled data gives you **anchors**
  - Unlabeled data lets you **fill in the space** between anchors
- The risk: this **might not be true**
  - Similar-looking inputs can have **different labels**
  - Might be on a **deceptive manifold**





Cluster Assumption



Cluster Assumption

- Data forms **natural clusters** and a decision boundary should **not split within a cluster**
  - Unlabeled data helps **find low-density space**
  - Adds **information not available from the labels**

Labeled data only

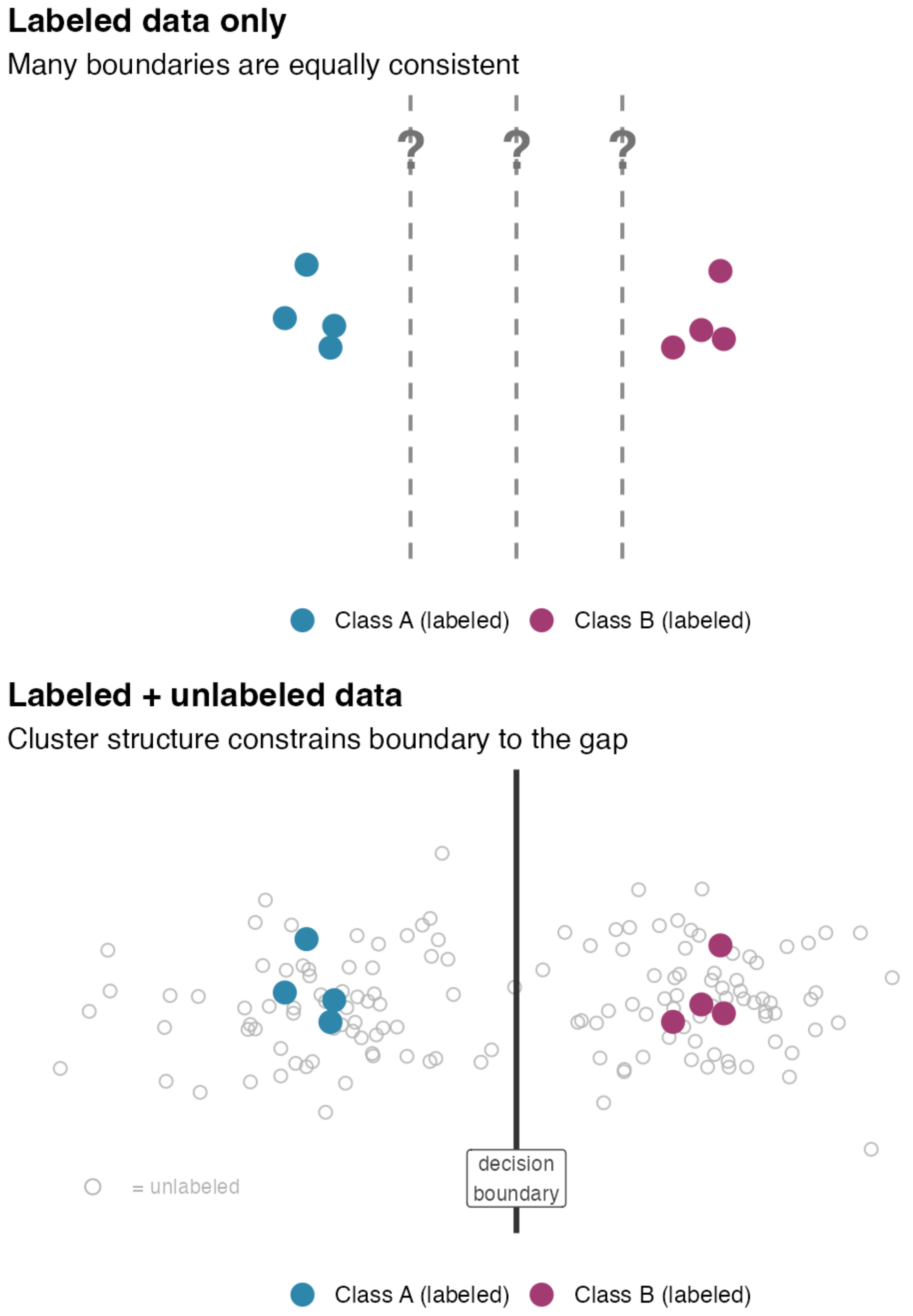
Many boundaries are equally consistent

Labeled + unlabeled data

Cluster structure constrains boundary to the gap

Cluster Assumption

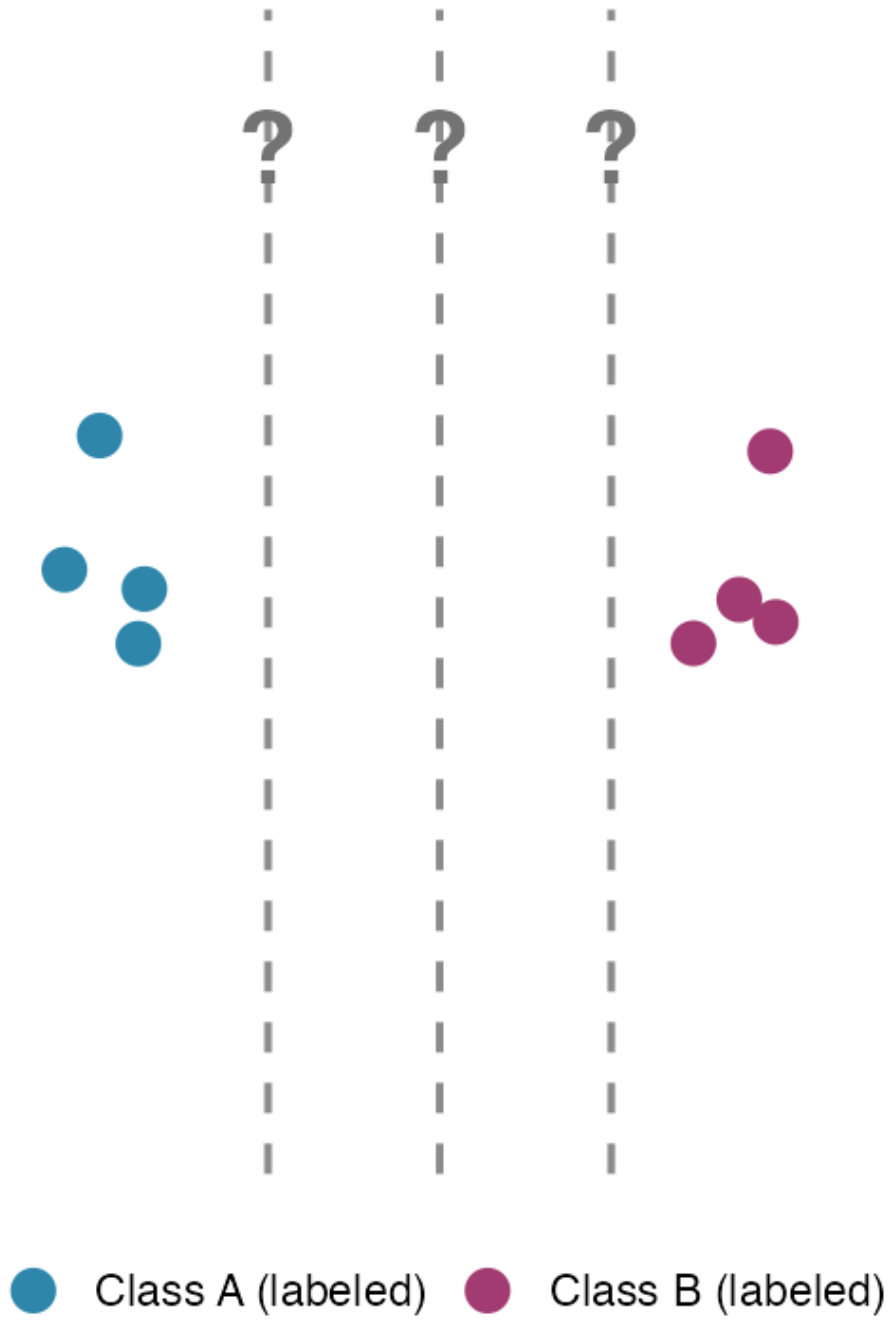
- Data forms **natural clusters** and a decision boundary should **not split within a cluster**
  - Unlabeled data helps **find low-density space**
  - Adds **information not available from the labels**
- Risk: natural clusters **might not exist**



Cluster Assumption

Labeled data only

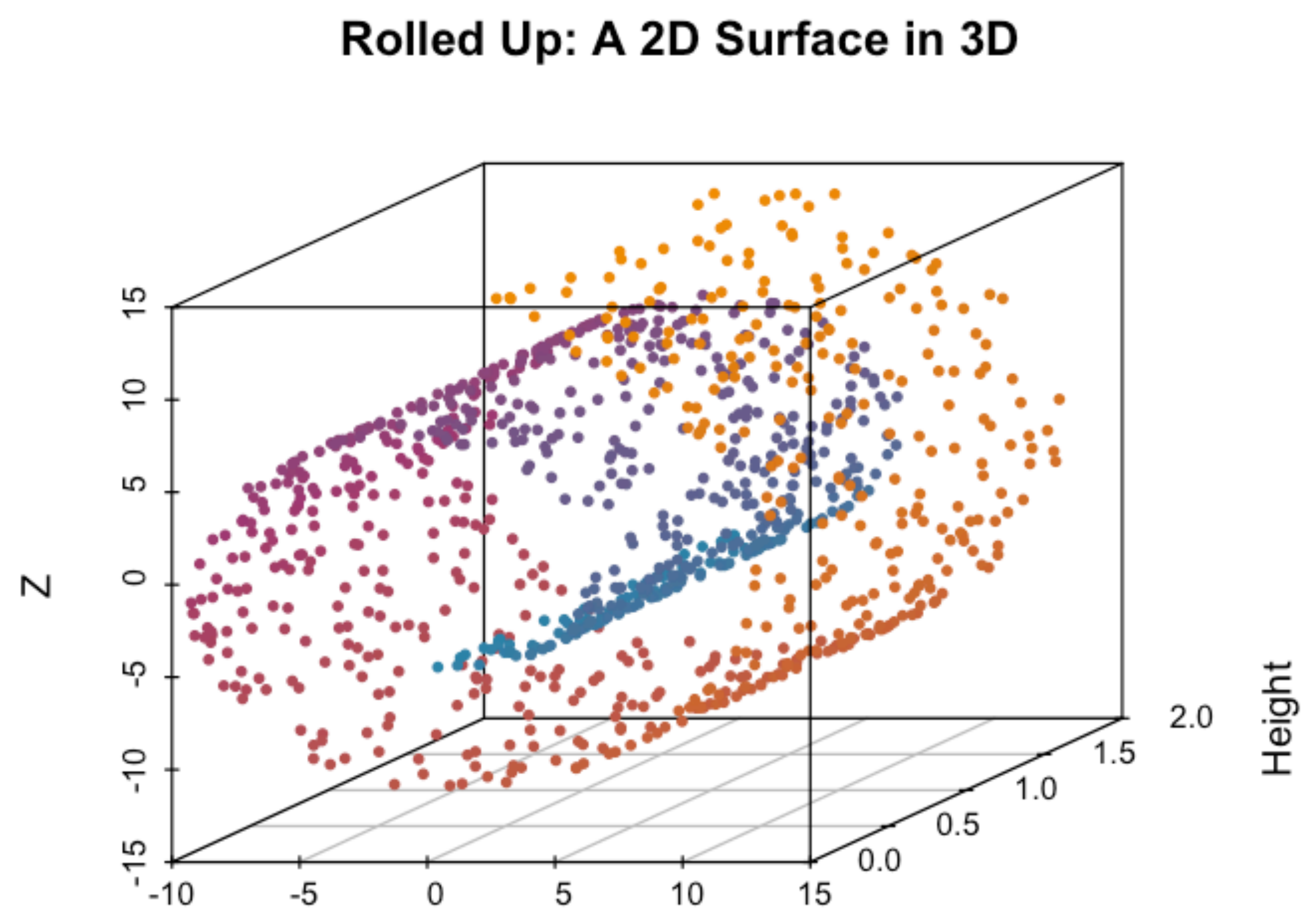
Many boundaries are equally consistent

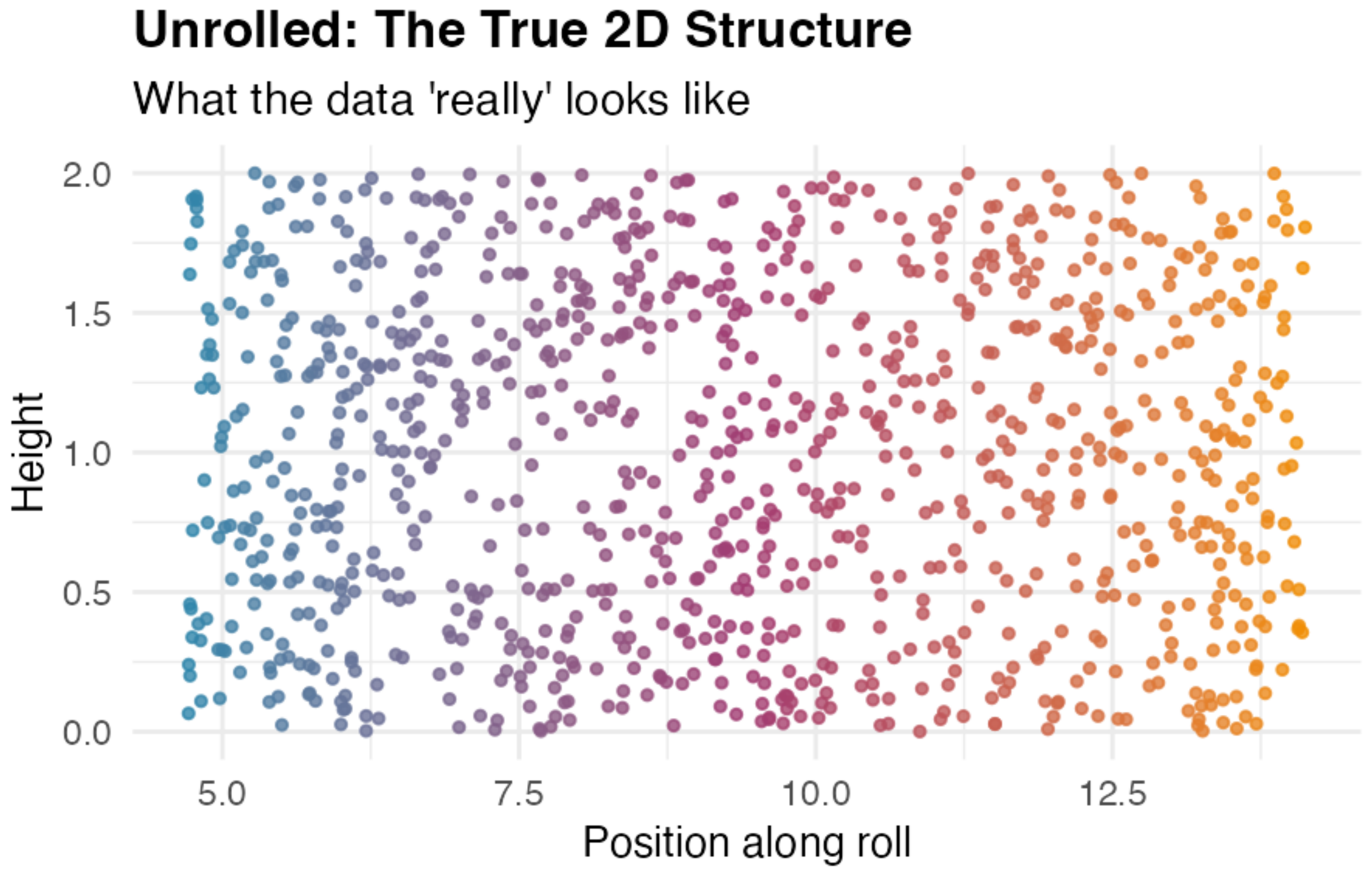


Labeled + unlabeled data

Cluster structure constrains boundary to the gap

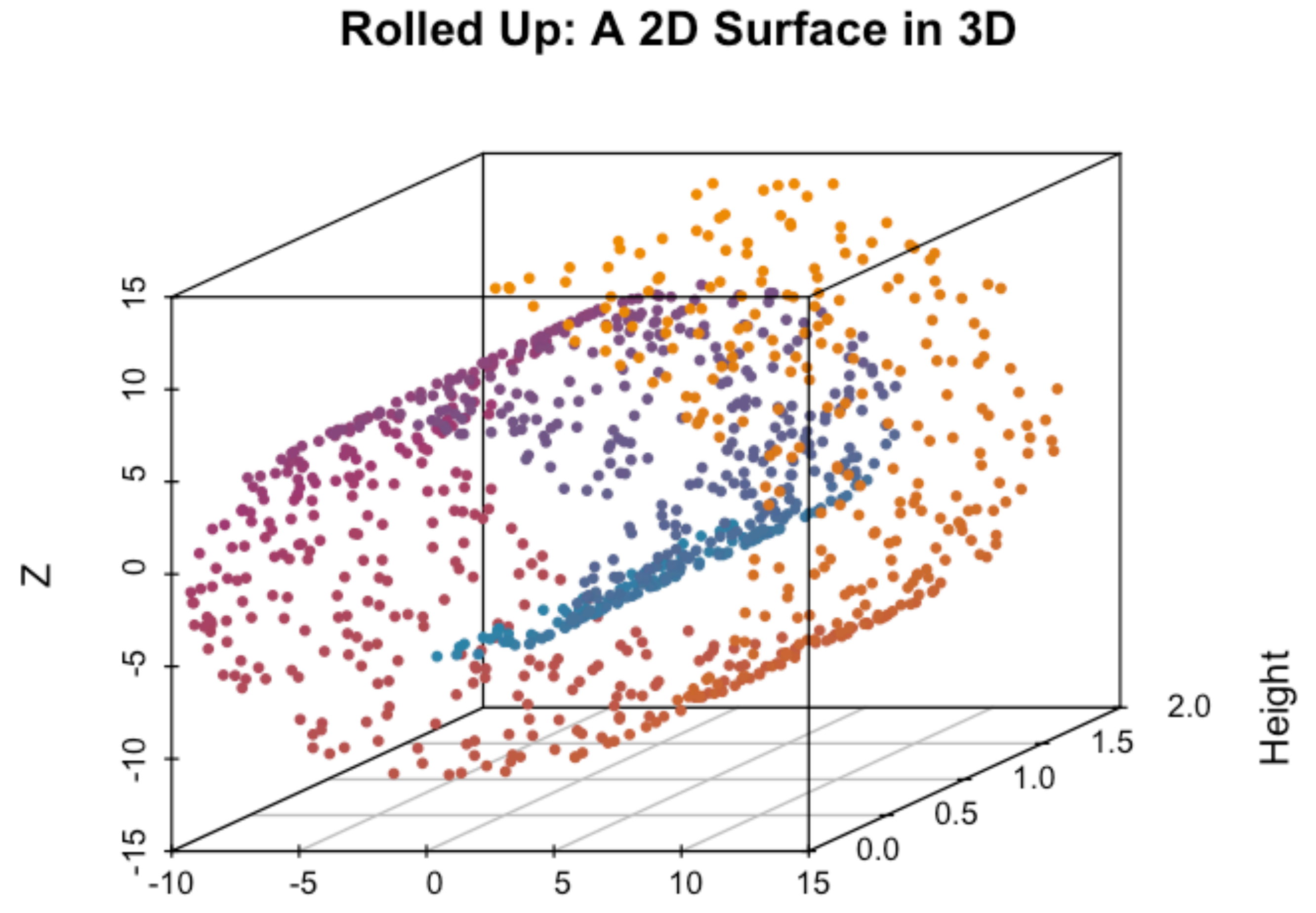
Manifold Assumption

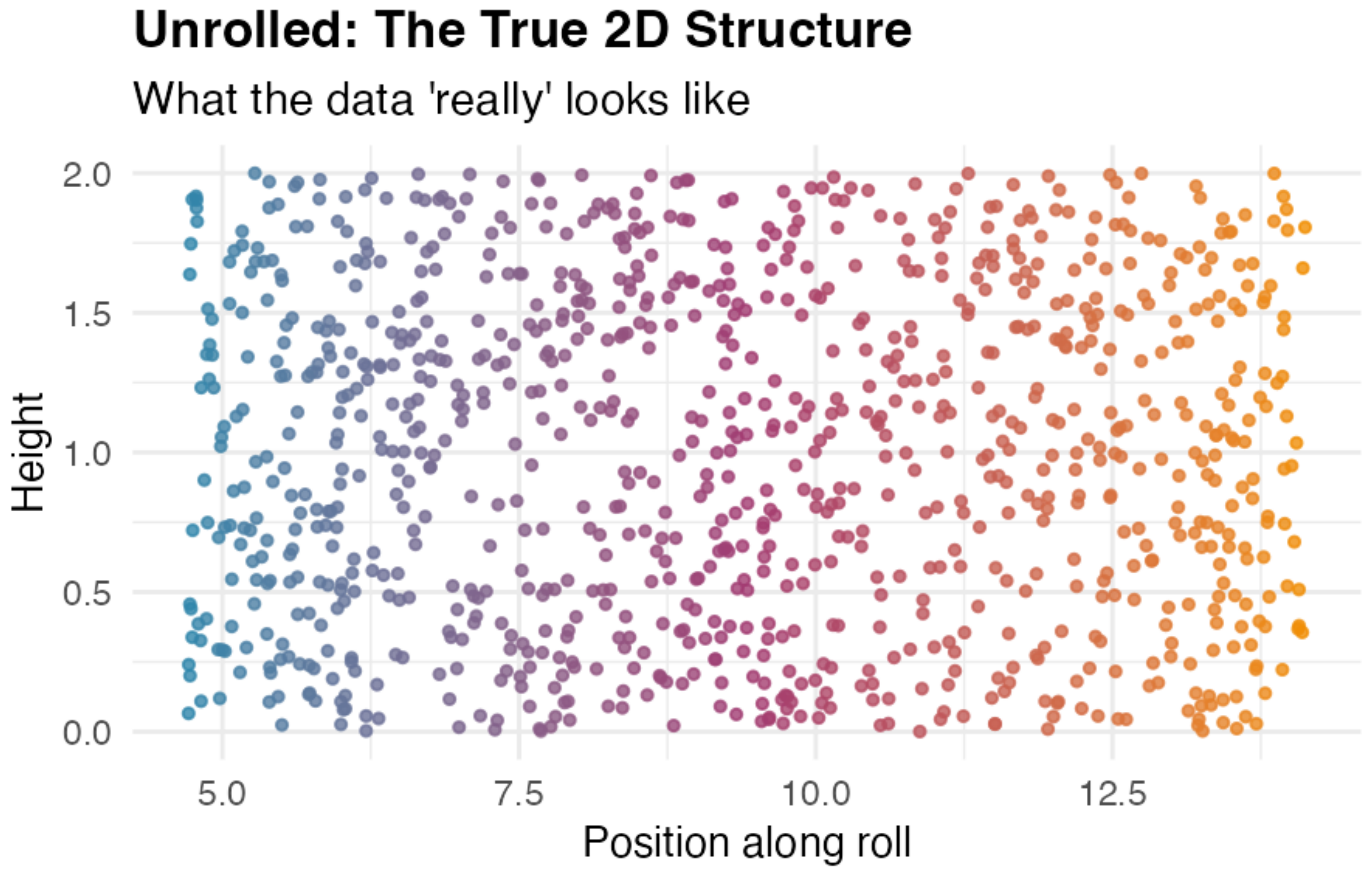




Manifold Assumption

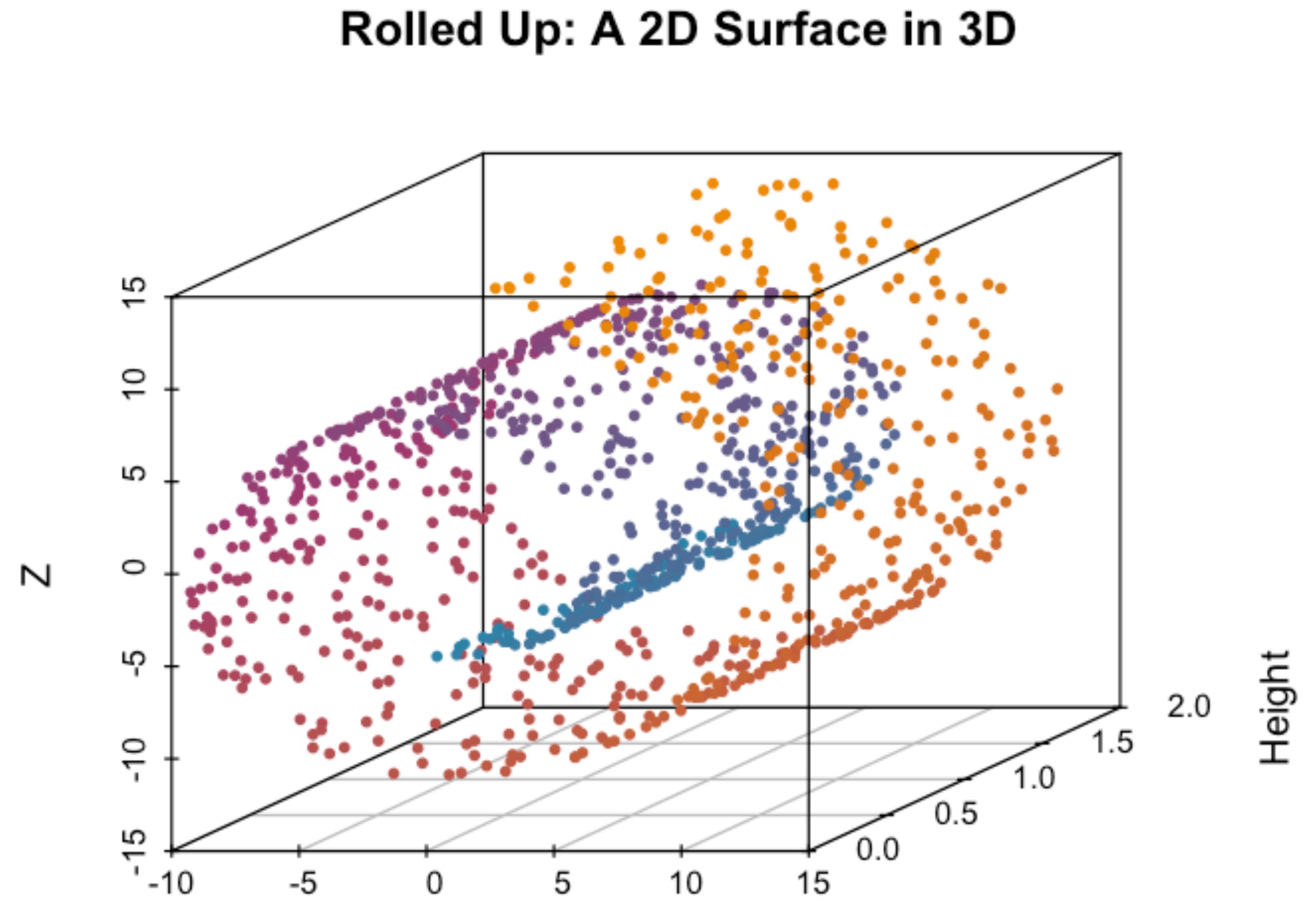
- High-dimensional data lies on a **learnable, low-dimensional manifold**
  - I.e. the **unlabeled structure is discoverable**

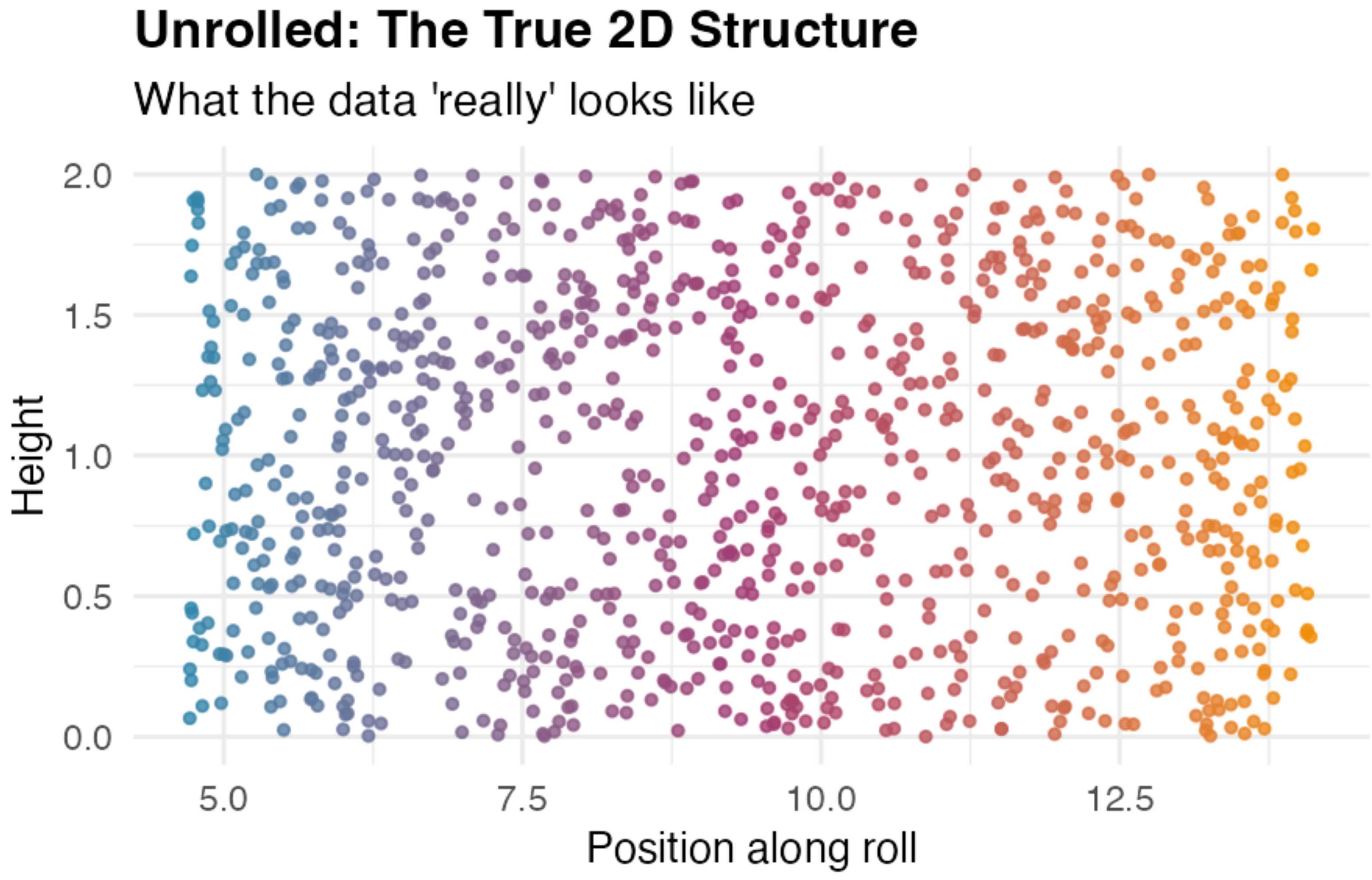




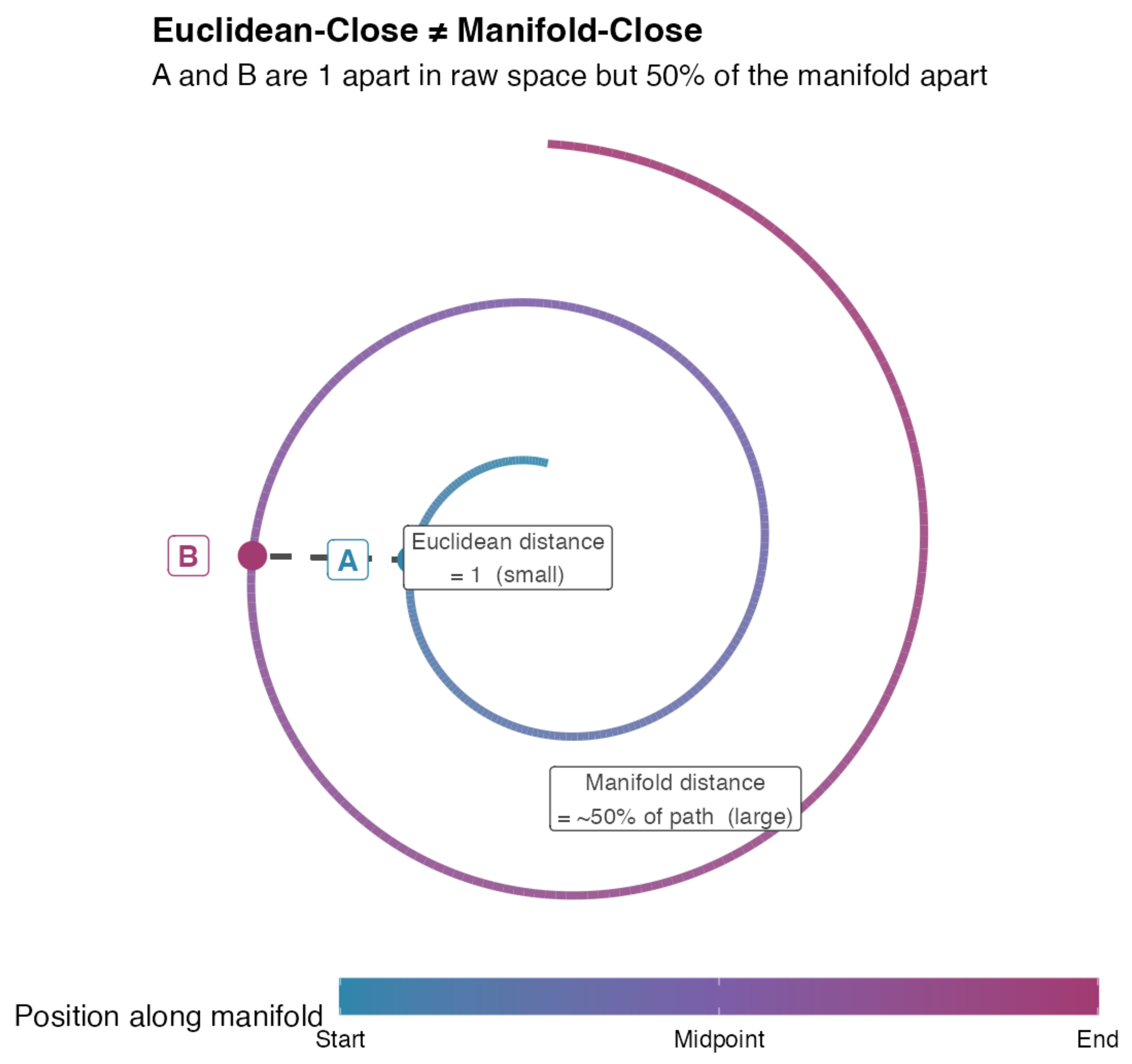
Manifold Assumption

- High-dimensional data lies on a **learnable, low-dimensional manifold**
  - I.e. the **unlabeled structure is discoverable**
- Important: other two assumptions probably **only hold along the manifold**
  - Manifold needs to be "unraveled" **before labels can be extrapolated**



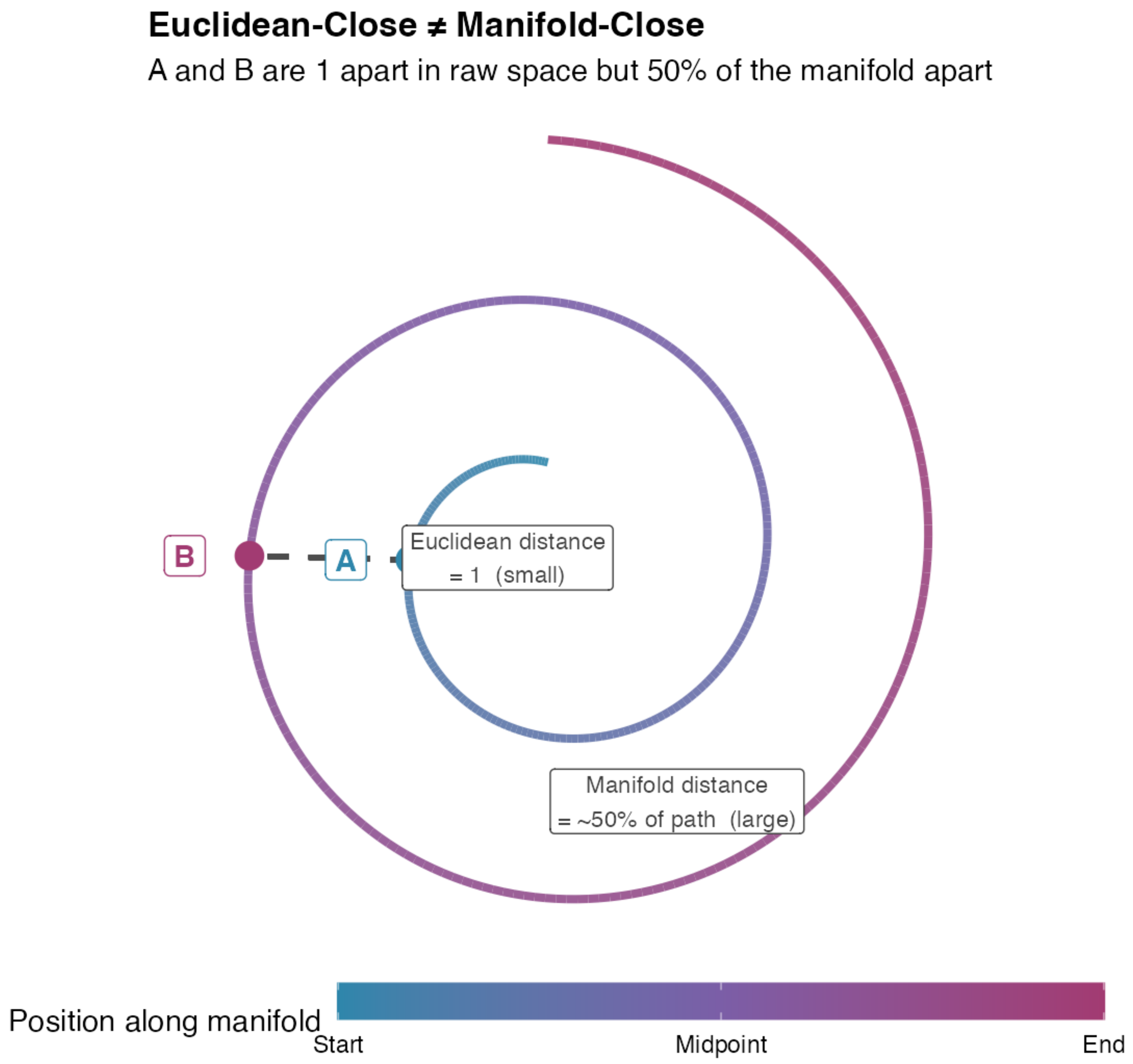


Manifold Importance



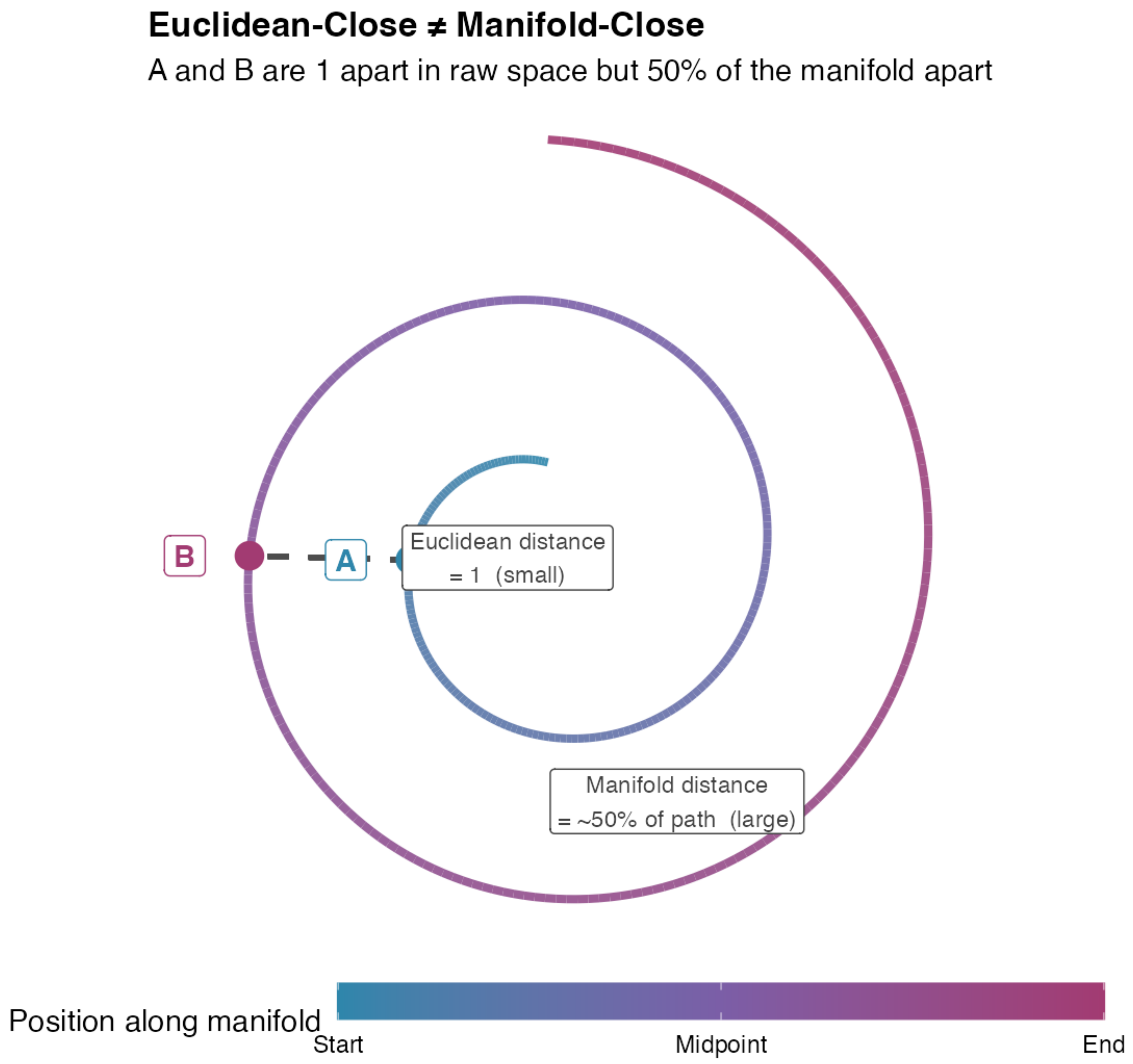
Manifold Importance

- Key point: smoothness and clustering assumptions need to hold in the **representation space**, **NOT** in the **raw input space**



Manifold Importance

- Key point: smoothness and clustering assumptions need to hold in the **representation space**, **NOT** in the **raw input space**
- In a deep neural model, the final layer tries to **linearly separate classes**
  - It is **these representations** that need to have the assumed properties
  - The prior layers do the work of **unraveling the manifold**

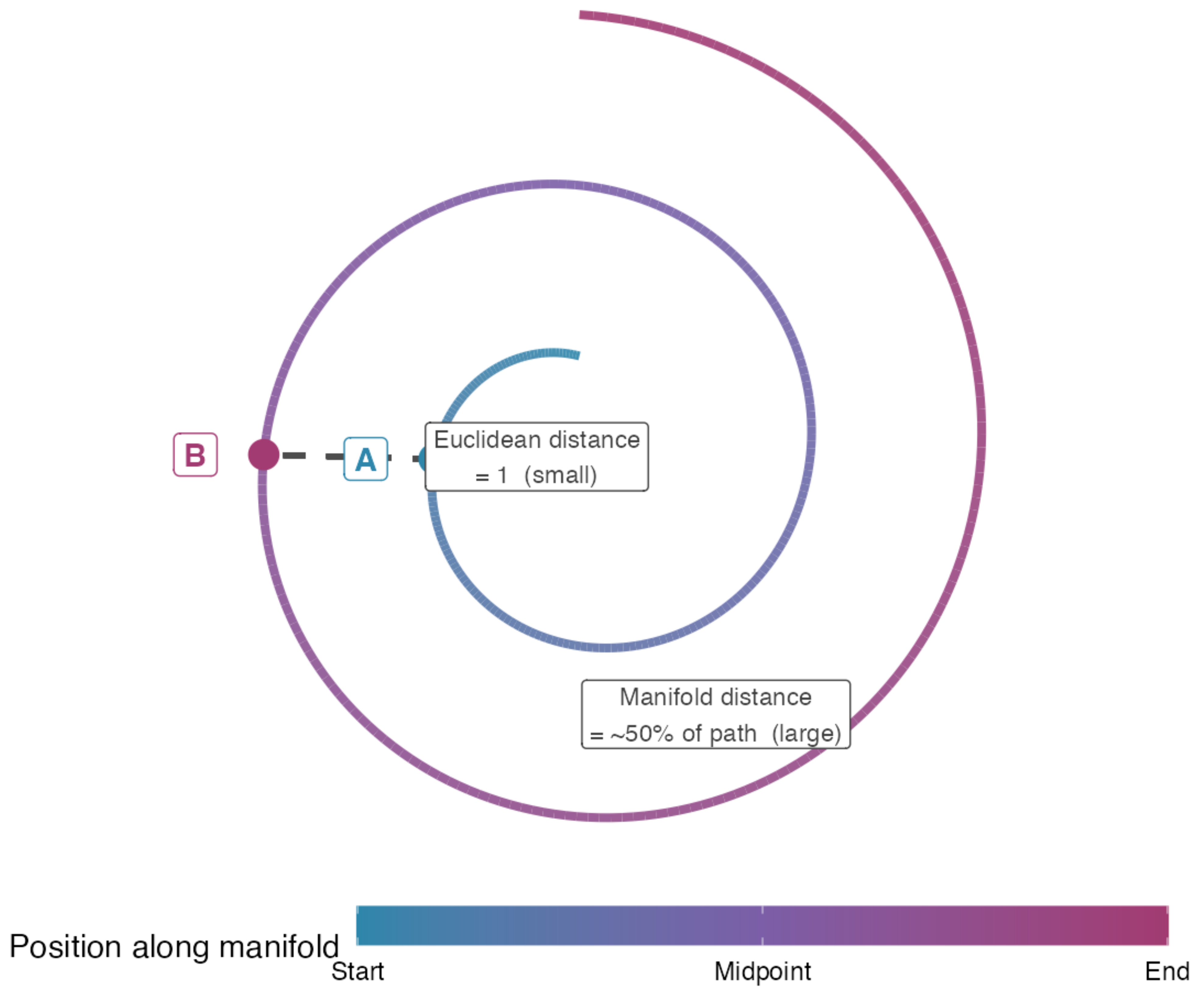


Manifold Importance

- Key point: smoothness and clustering assumptions need to hold in the **representation space**, **NOT** in the **raw input space**
- In a deep neural model, the final layer tries to **linearly separate classes**
  - It is **these representations** that need to have the assumed properties
  - The prior layers do the work of **unraveling the manifold**
- There is **no principled reason** to believe the assumptions will hold **before representation learning**

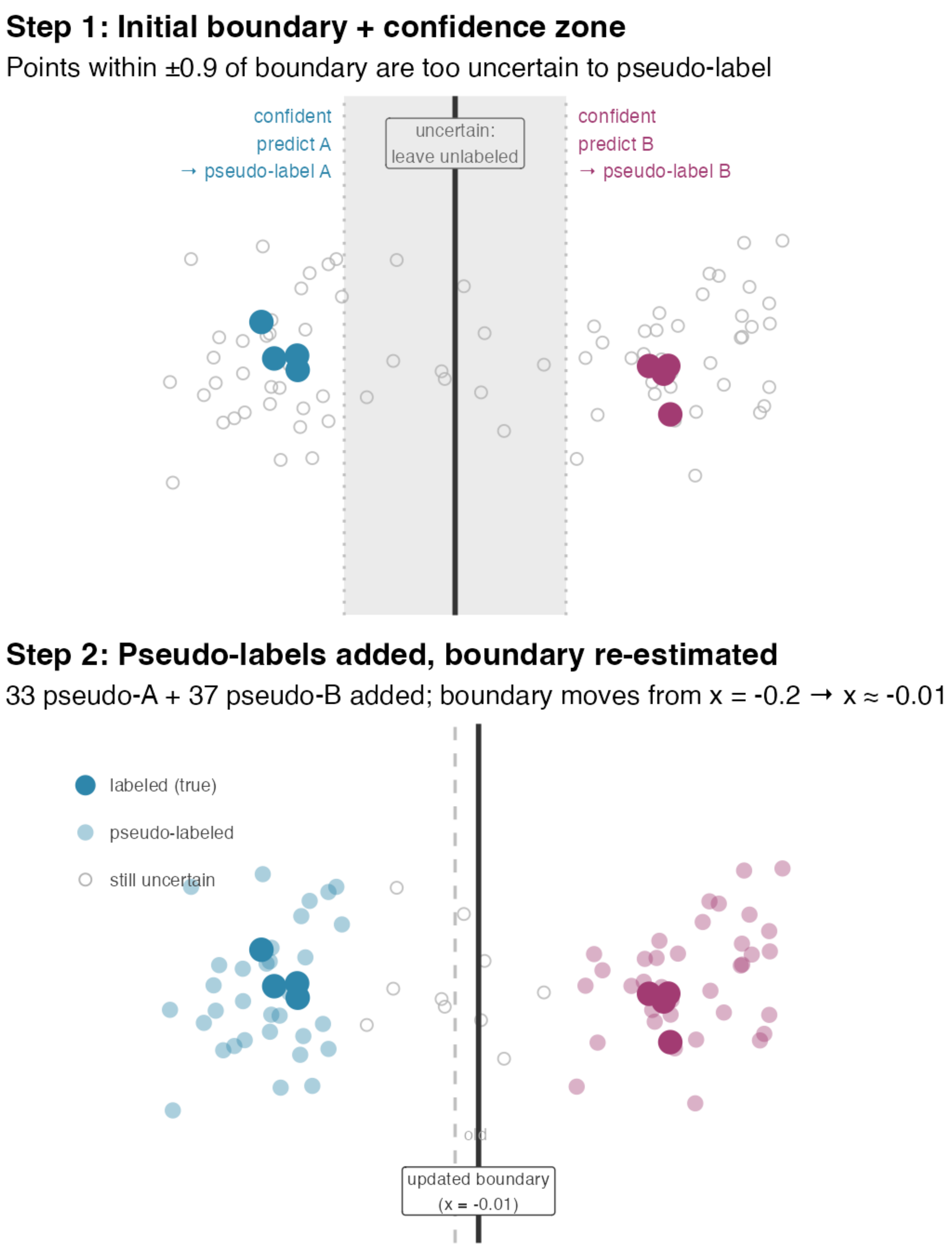
Euclidean-Close ≠ Manifold-Close

A and B are 1 apart in raw space but 50% of the manifold apart



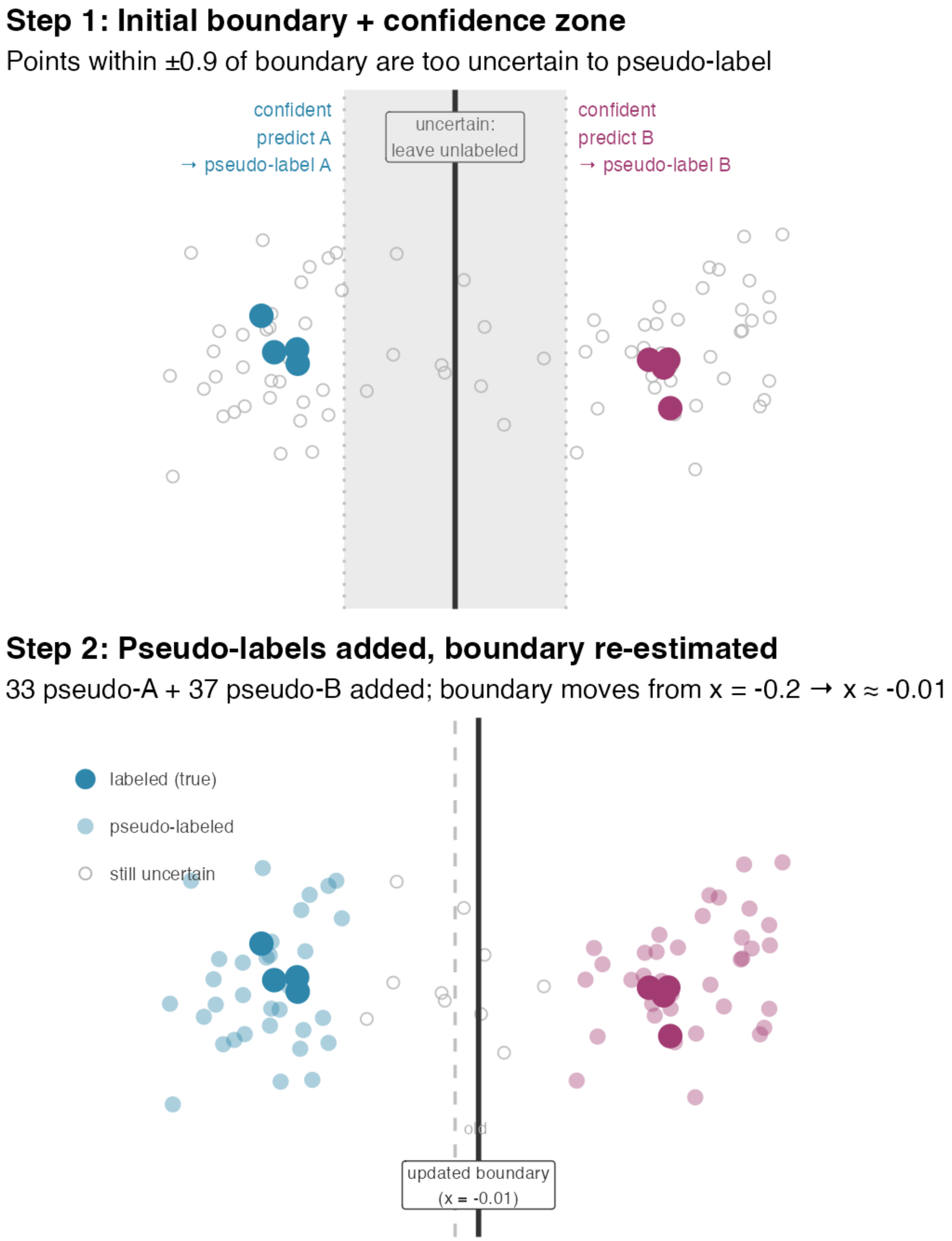
Pseudo-Labeling

Pseudo-Labeling Algorithm



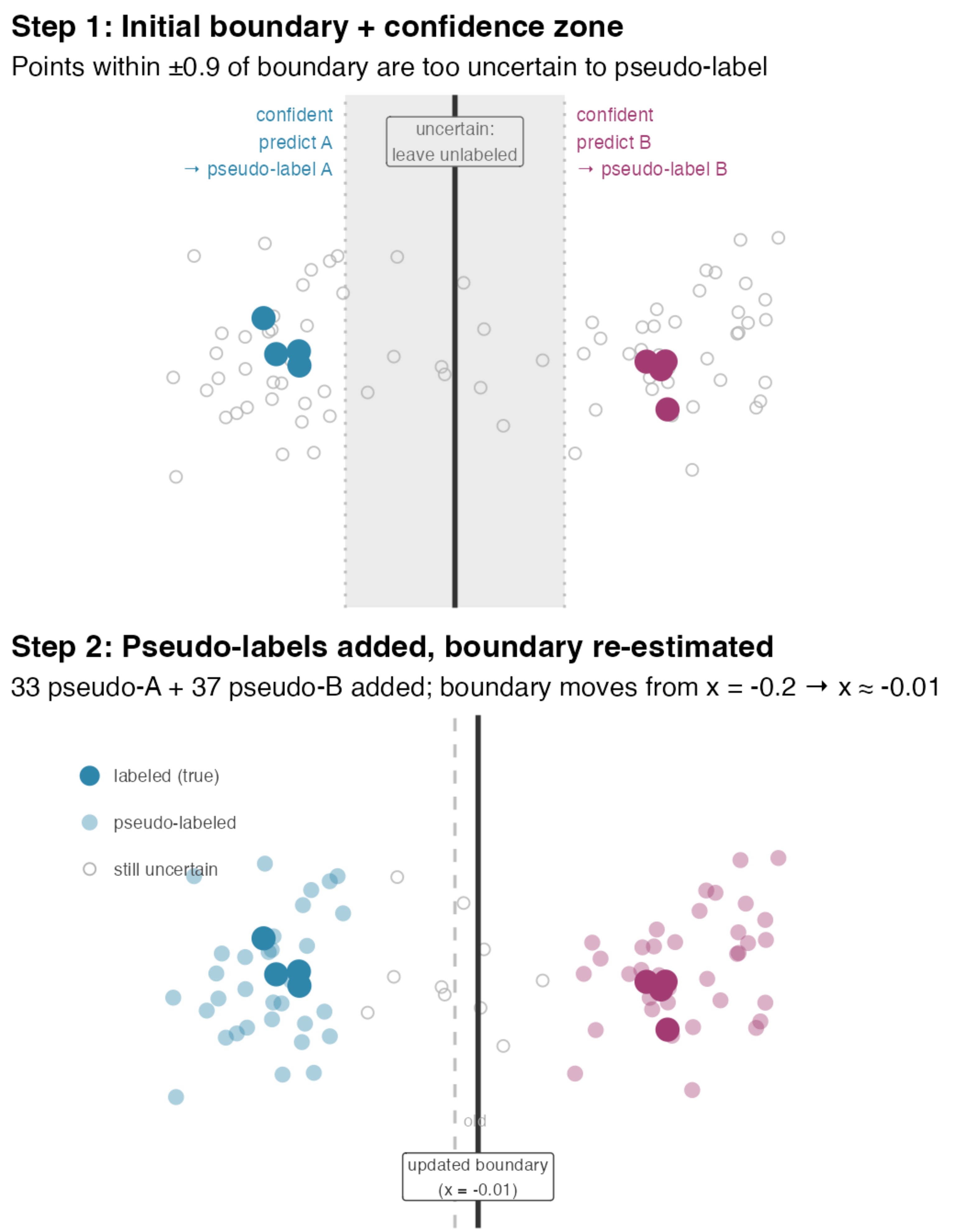
Pseudo-Labeling Algorithm

- Train classifier on **labeled set** L



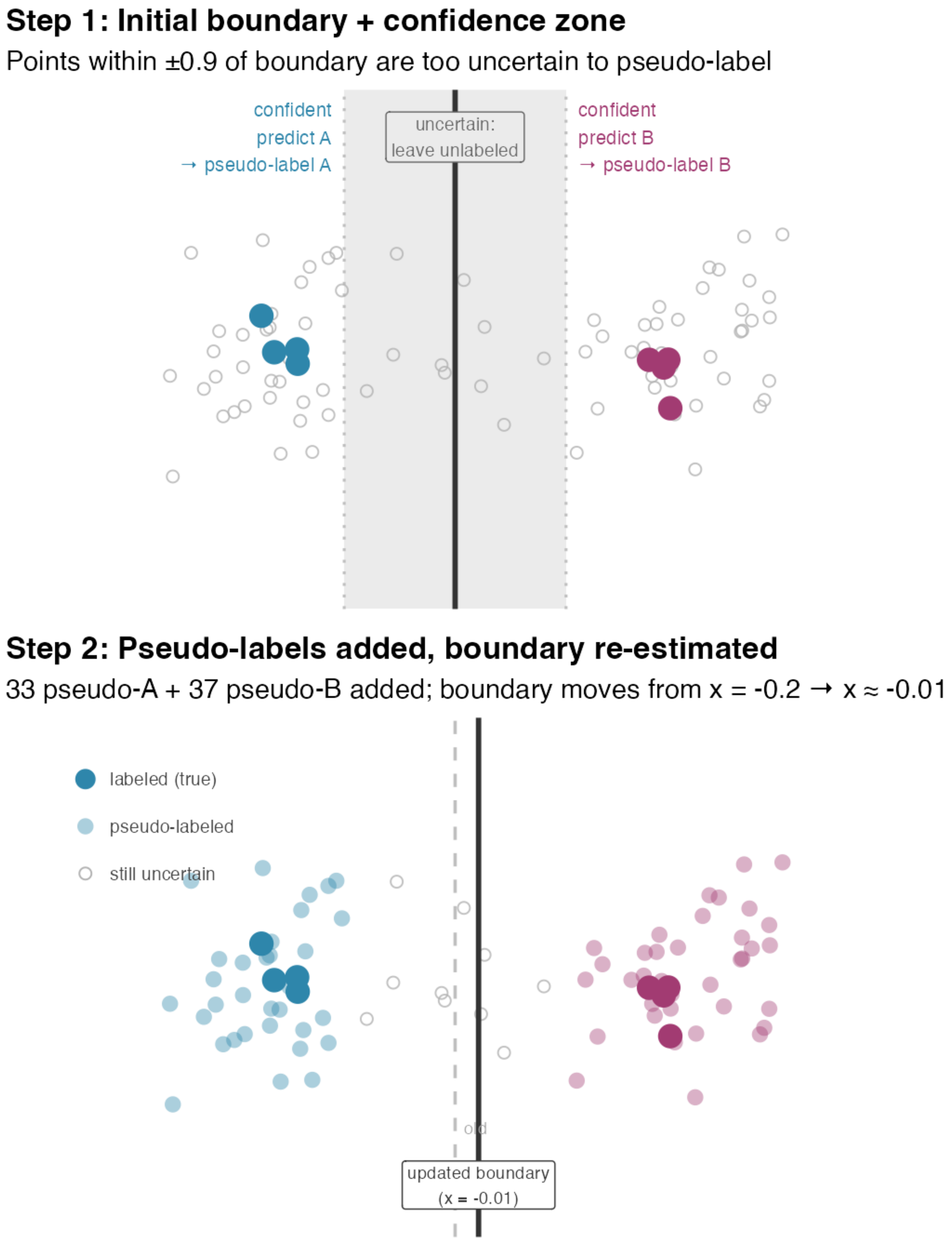
Pseudo-Labeling Algorithm

- Train classifier on **labeled set** L
- **Apply** classifier to **unlabeled set** U $\rightarrow$ get **label predictions**



Pseudo-Labeling Algorithm

- Train classifier on **labeled set** L
- **Apply** classifier to **unlabeled set** U
  $\rightarrow$ get **label predictions**
- Select **high-confidence predictions**, keep them as "pseudo-labels"

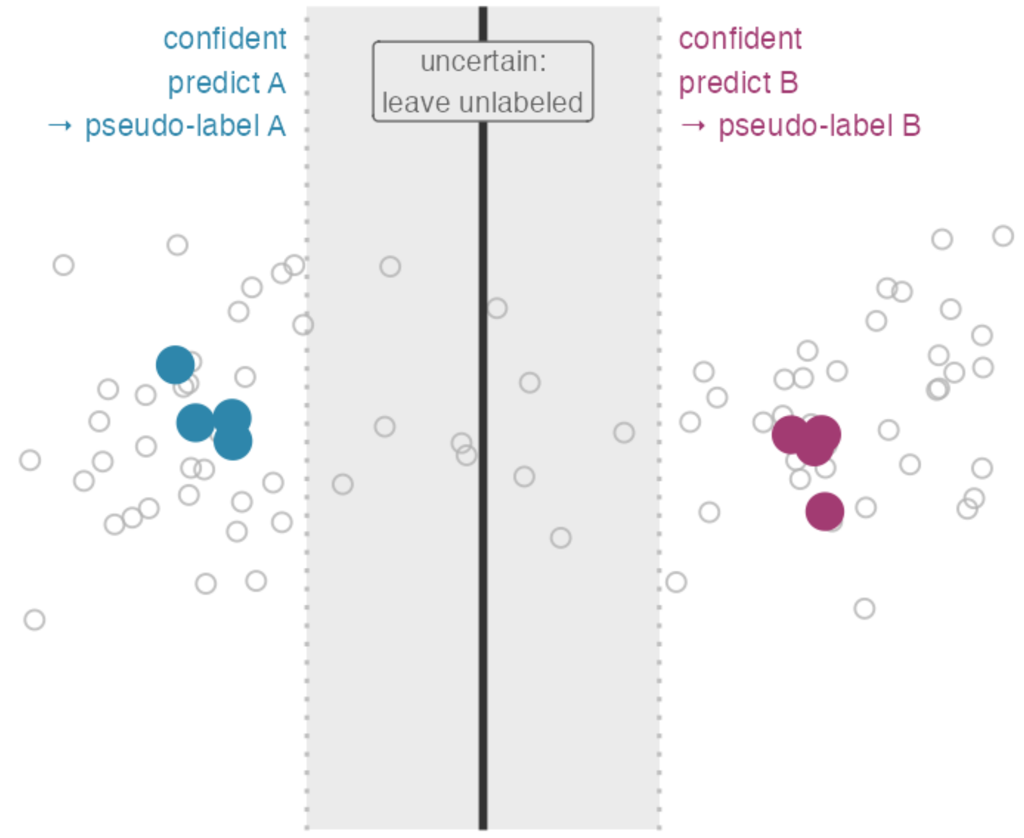


Pseudo-Labeling Algorithm

- Train classifier on **labeled set** L
- **Apply** classifier to **unlabeled set** U
  $\rightarrow$ get **label predictions**
- Select **high-confidence predictions**, keep them as "pseudo-labels"
- **Add pseudo-pairs** $(x_i, \hat{y}_i)$ to L

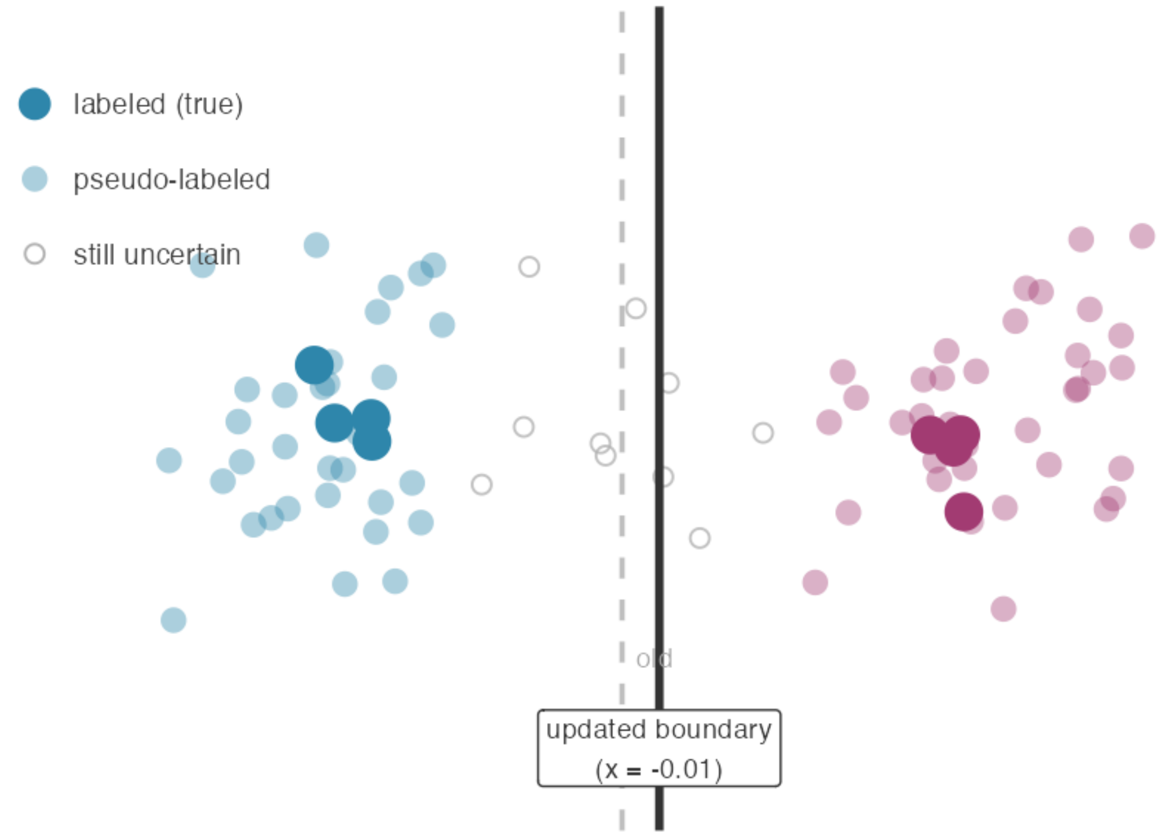
Step 1: Initial boundary + confidence zone

Points within ±0.9 of boundary are too uncertain to pseudo-label



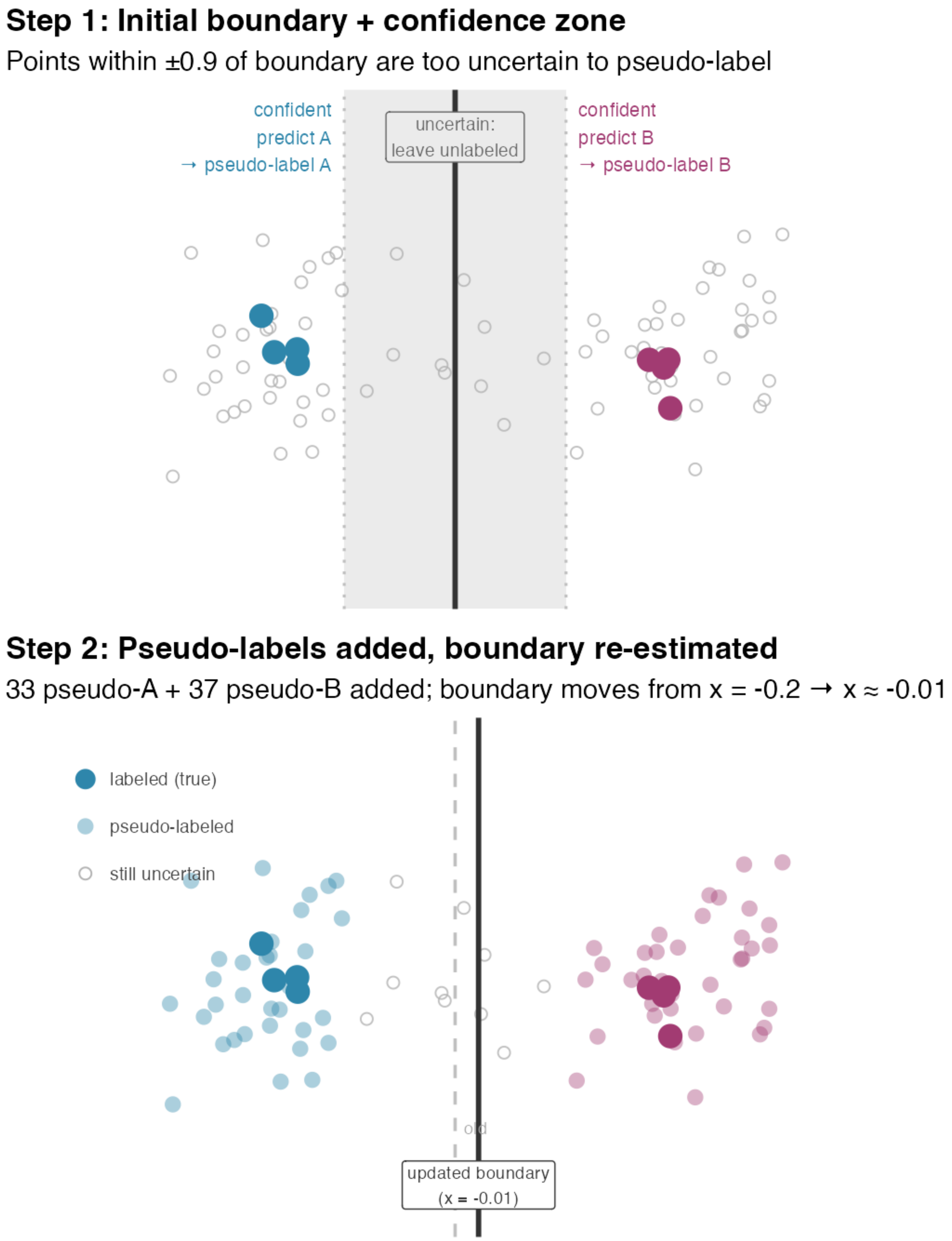
Step 2: Pseudo-labels added, boundary re-estimated

33 pseudo-A + 37 pseudo-B added; boundary moves from x = -0.2 → x ≈ -0.01



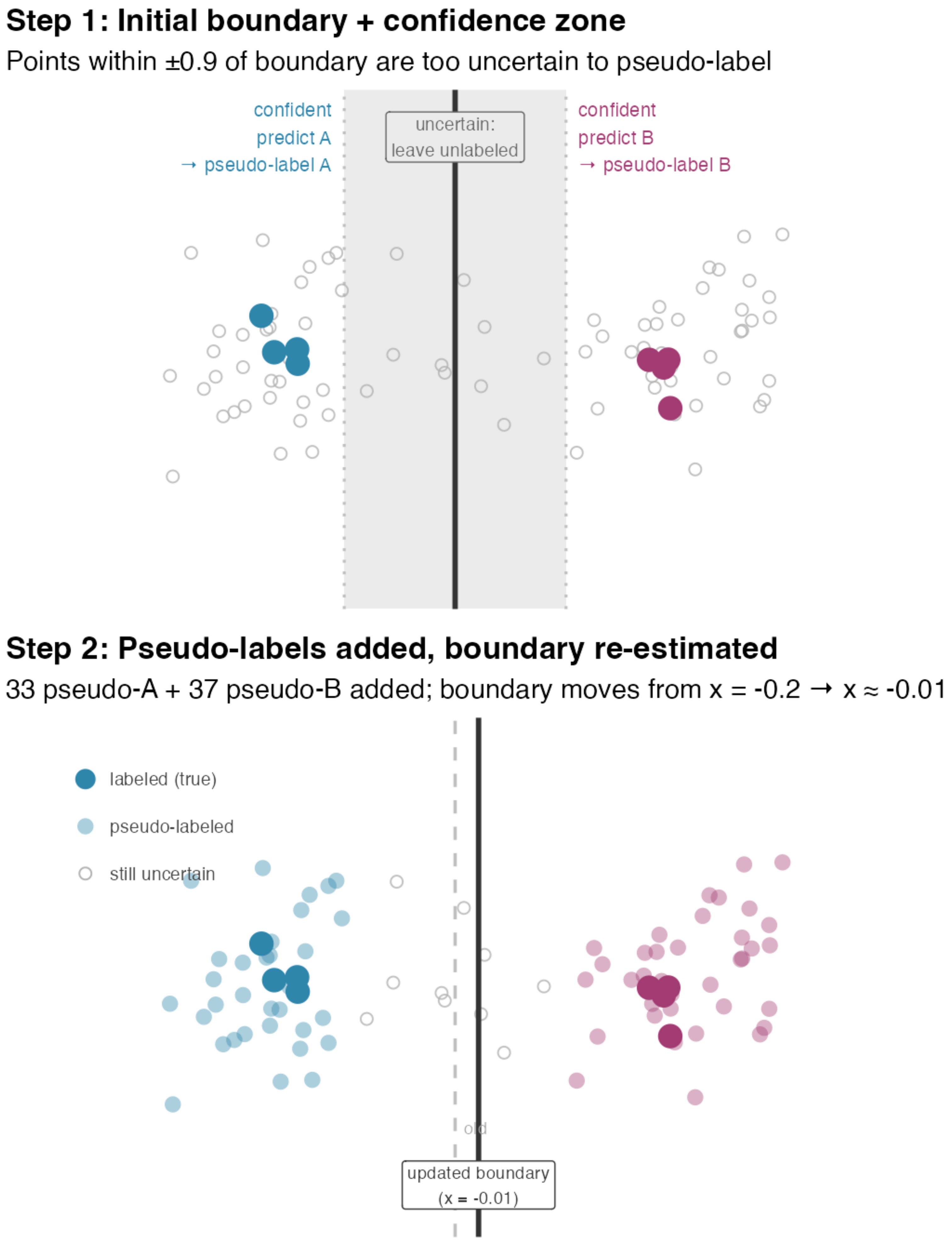
Pseudo-Labeling Algorithm

- Train classifier on **labeled set** L
- **Apply** classifier to **unlabeled set** U $\rightarrow$ get **label predictions**
- Select **high-confidence predictions**, keep them as "pseudo-labels"
- **Add pseudo-pairs** $(x_i, \hat{y}_i)$ to L
- **Re-train** on expanded dataset



Pseudo-Labeling Algorithm

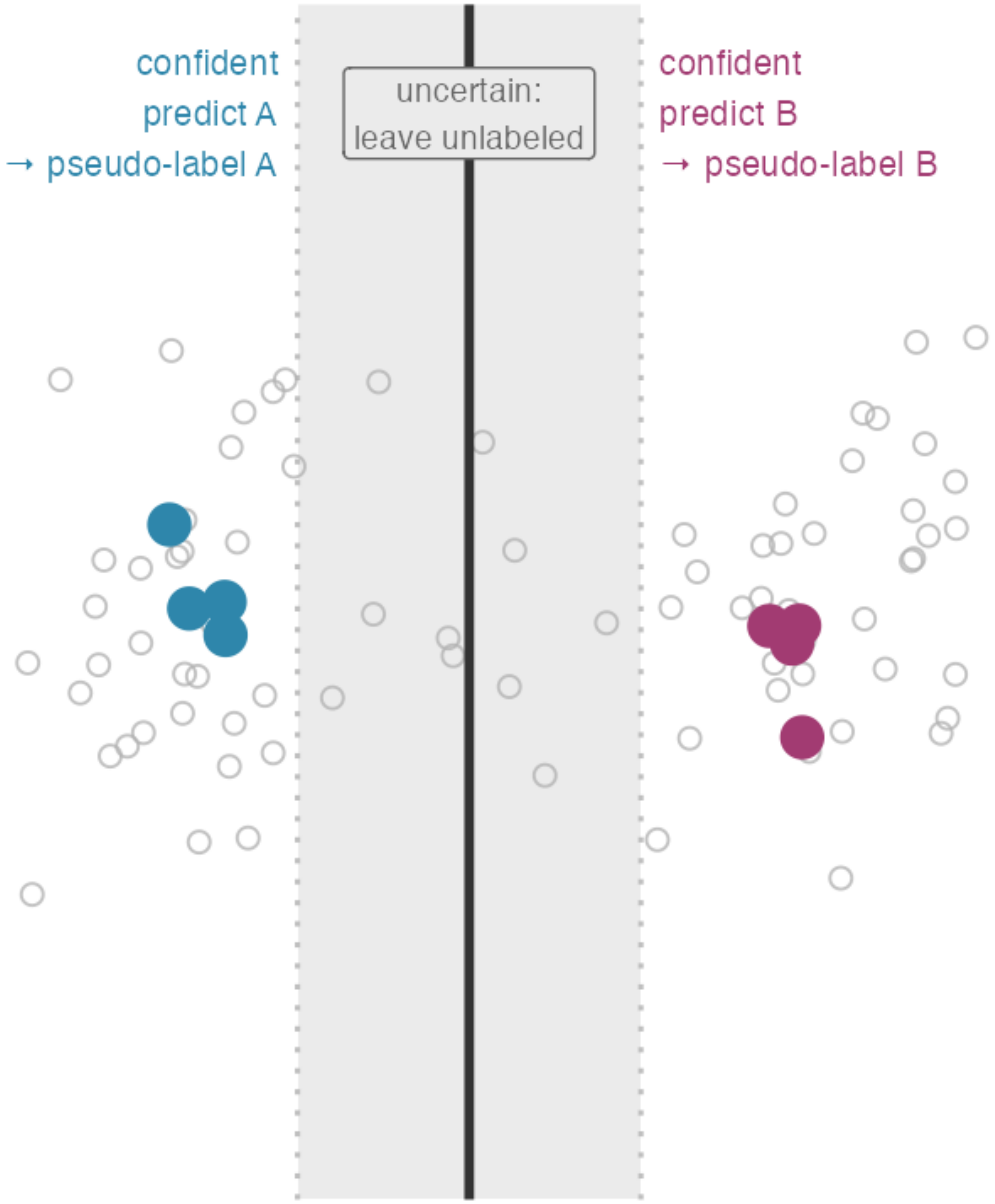
- Train classifier on **labeled set** L
- **Apply** classifier to **unlabeled set** U $\rightarrow$ get **label predictions**
- Select **high-confidence predictions**, keep them as "pseudo-labels"
- **Add pseudo-pairs** $(x_i, \hat{y}_i)$ to L
- **Re-train** on expanded dataset
- **Repeat** until improvement stops



Pseudo-Labeling

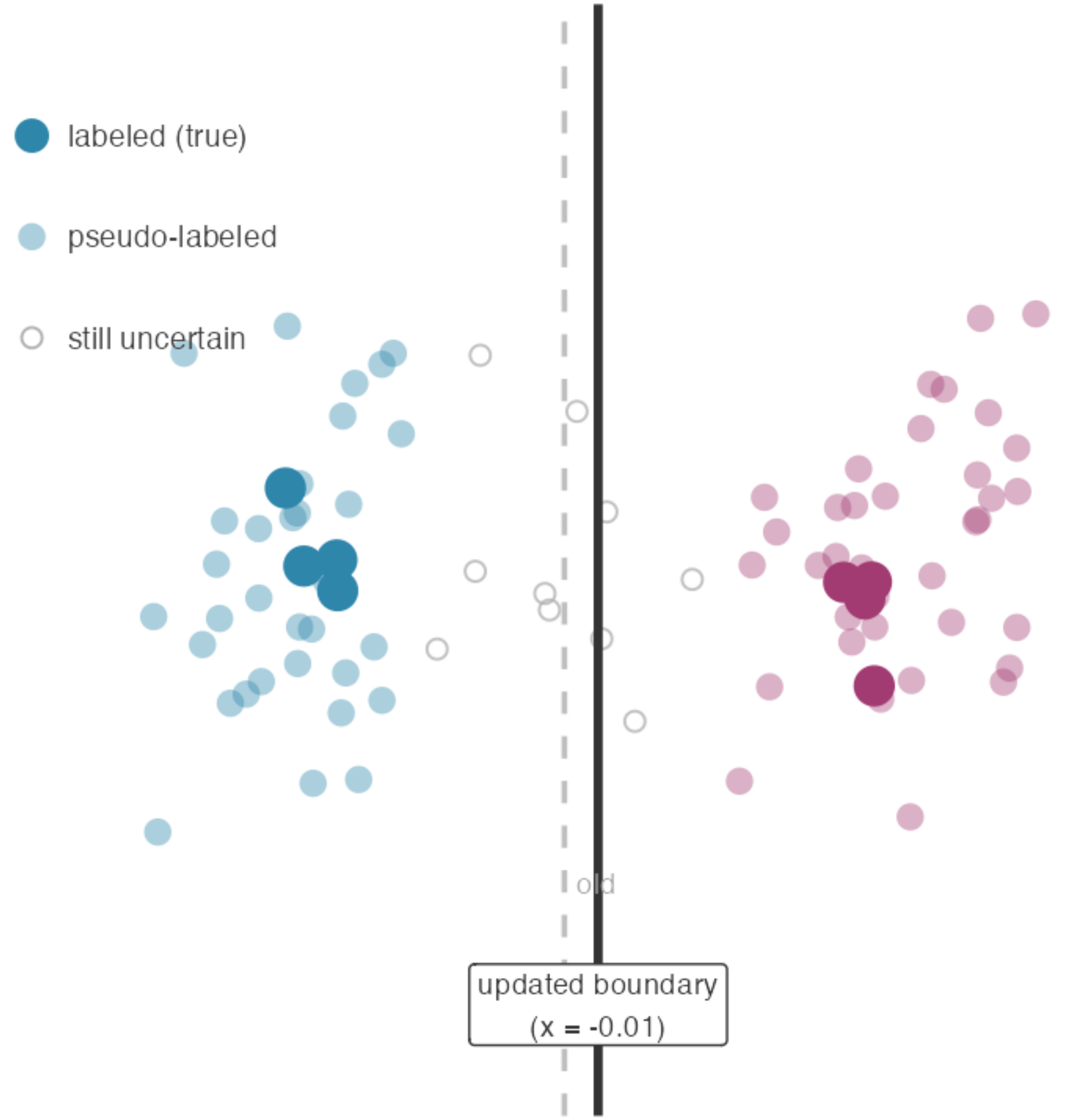
Step 1: Initial boundary + confidence zone

Points within ±0.9 of boundary are too uncertain to pseudo-label



Step 2: Pseudo-labels added, boundary re-estimated

33 pseudo-A + 37 pseudo-B added; boundary moves from x = -0.2 →



Why Pseudo-Labeling might work

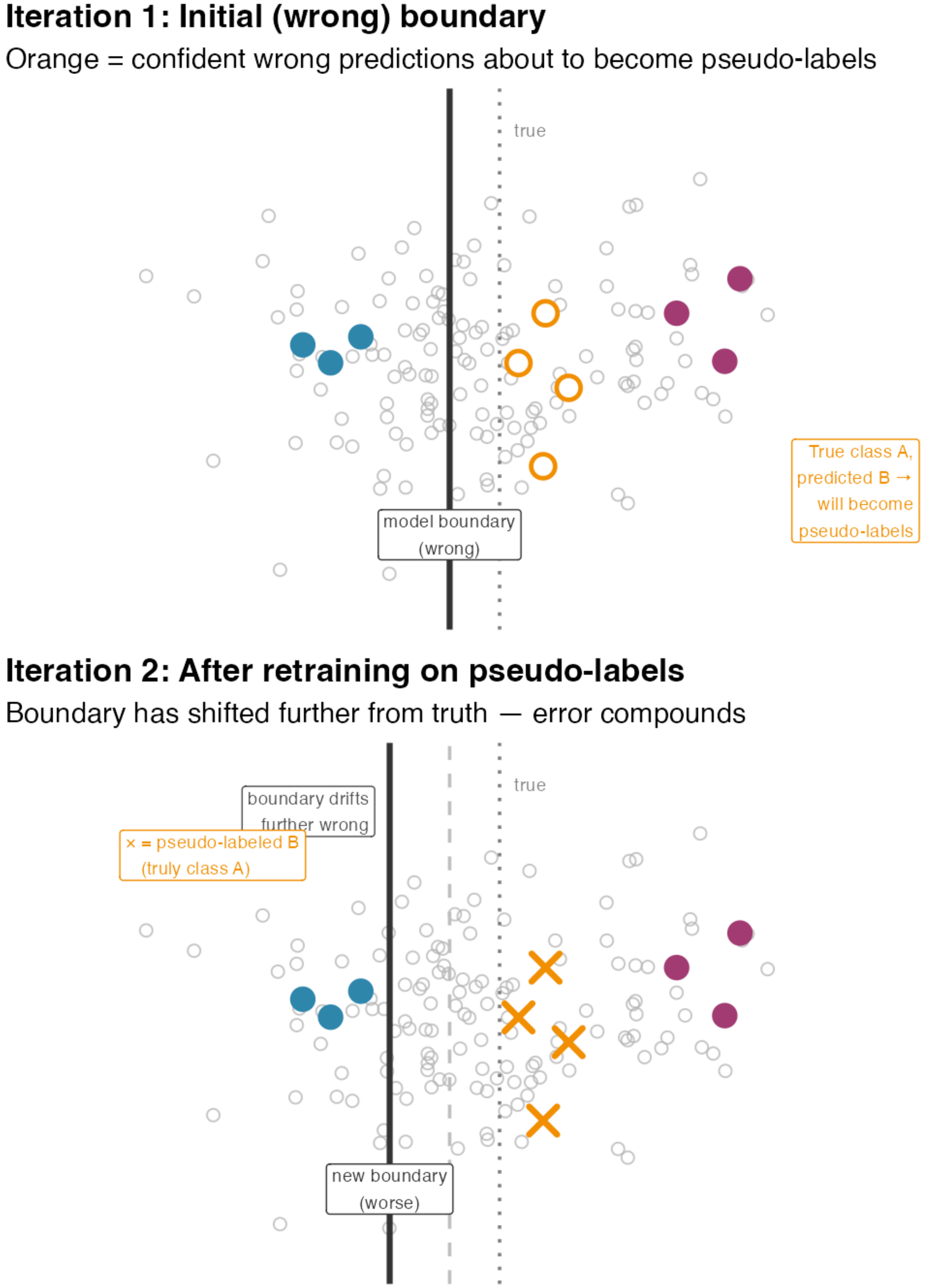
Why Pseudo-Labeling might work

- If your model is **already decent**, confident predictions are **probably right**
  - Adding these as training data should **reinforce what's been learned**
  - Over time, might be able to **label all data**

Why Pseudo-Labeling might work

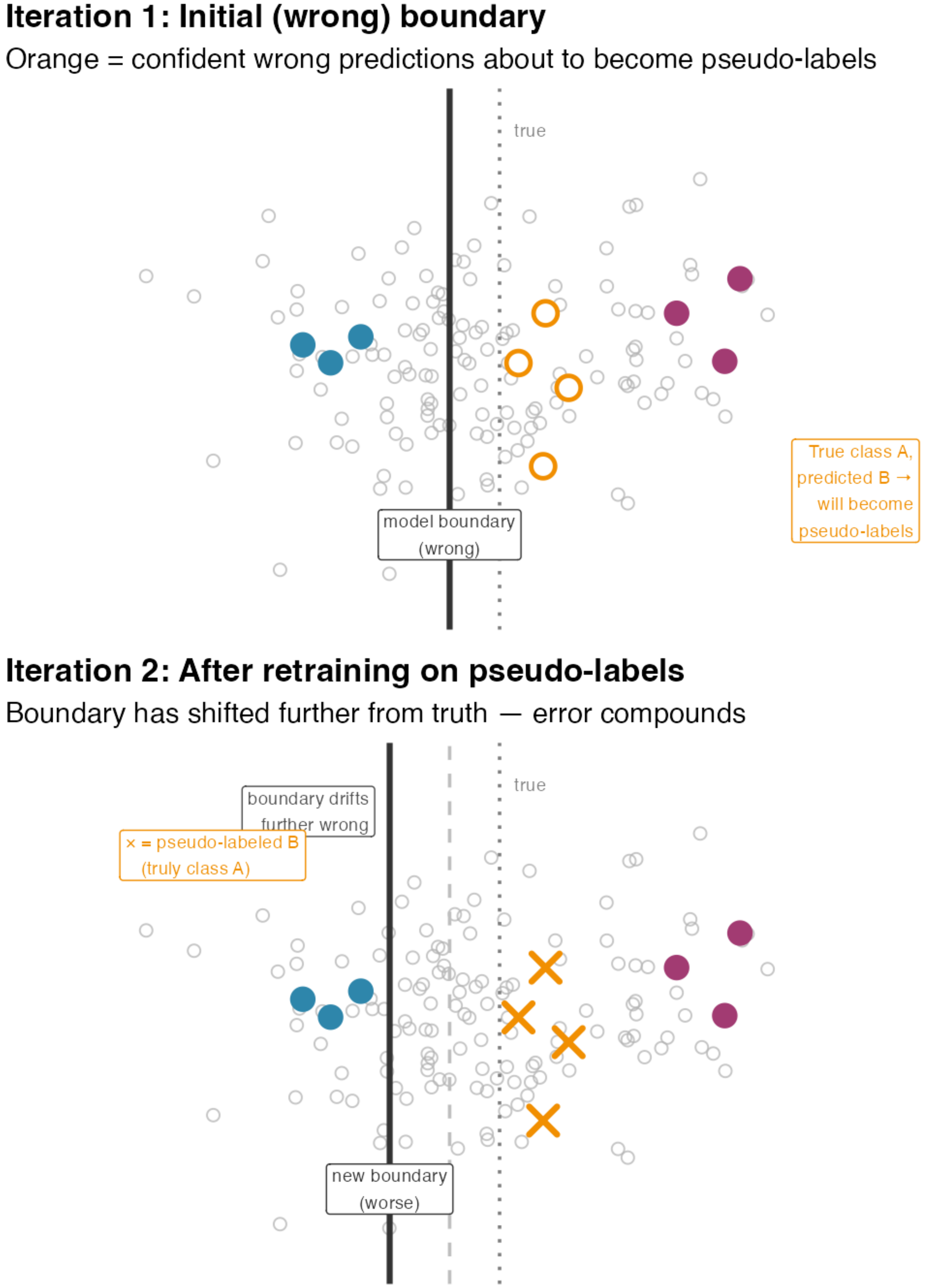
- If your model is **already decent**, confident predictions are **probably right**
  - Adding these as training data should **reinforce what's been learned**
  - Over time, might be able to **label all data**
- The **confidence threshold** encourages **high-quality data**
  - Common choice: $P(\hat{y}_i = y_i) > 0.95$
  - Lower threshold: **more pseudo-labels** but **more noise too**
  - Higher threshold: **fewer pseudo-labels** but **cleaner signal**

Failure Mode: Confirmation Bias



Failure Mode: Confirmation Bias

- Label level: classifier is **wrong** → generates **wrong pseudo-labels** → wrong labels **used for training** → classifier becomes **more confidently wrong** → **errors compound**

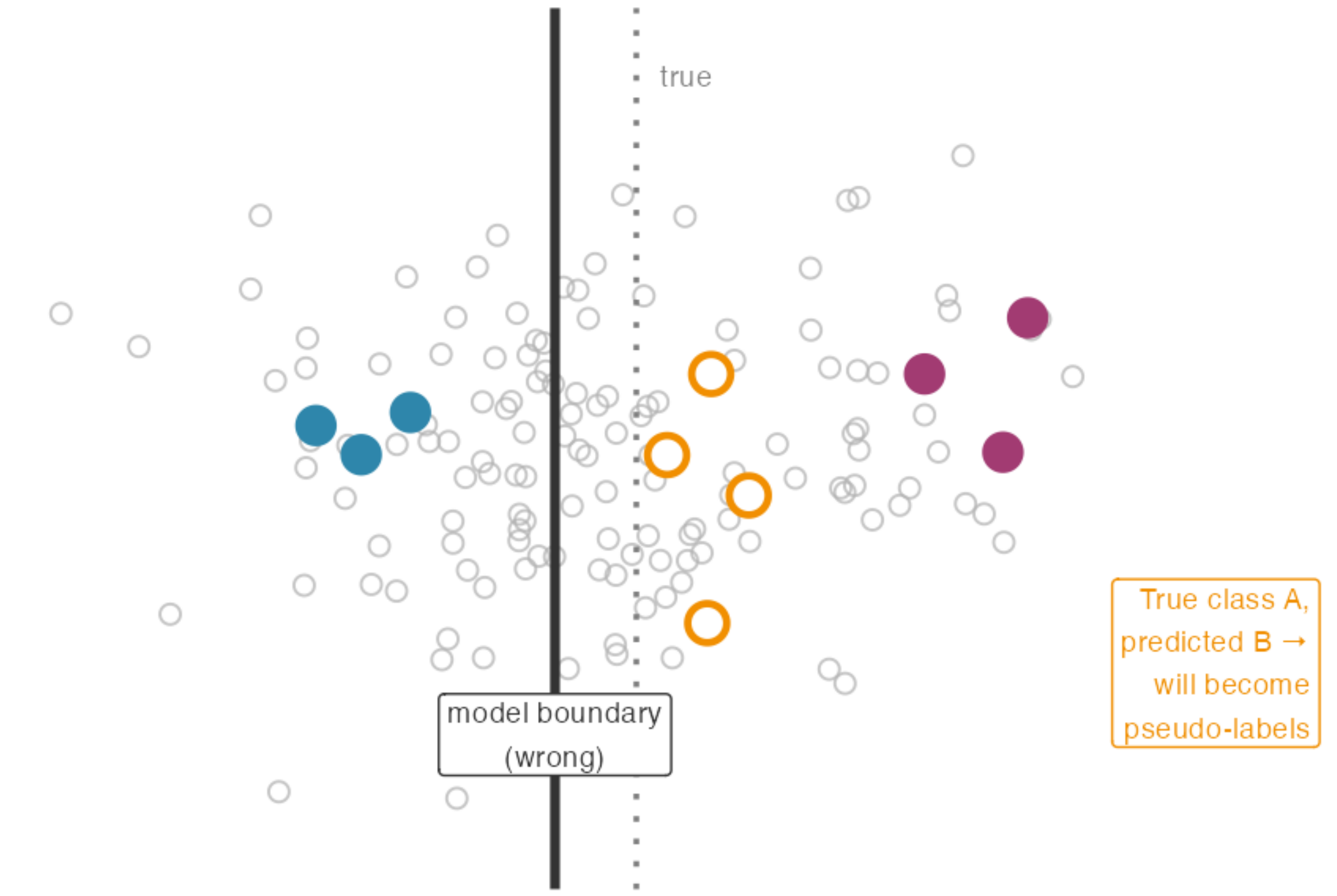


Failure Mode: Confirmation Bias

- Label level: classifier is **wrong** → generates **wrong pseudo-labels** → wrong labels **used for training** → classifier becomes **more confidently wrong** → **errors compound**
- Another view: an **overfit classifier** on top of a representation learning model will **distort the learned representations** (unfold the manifold incorrectly)

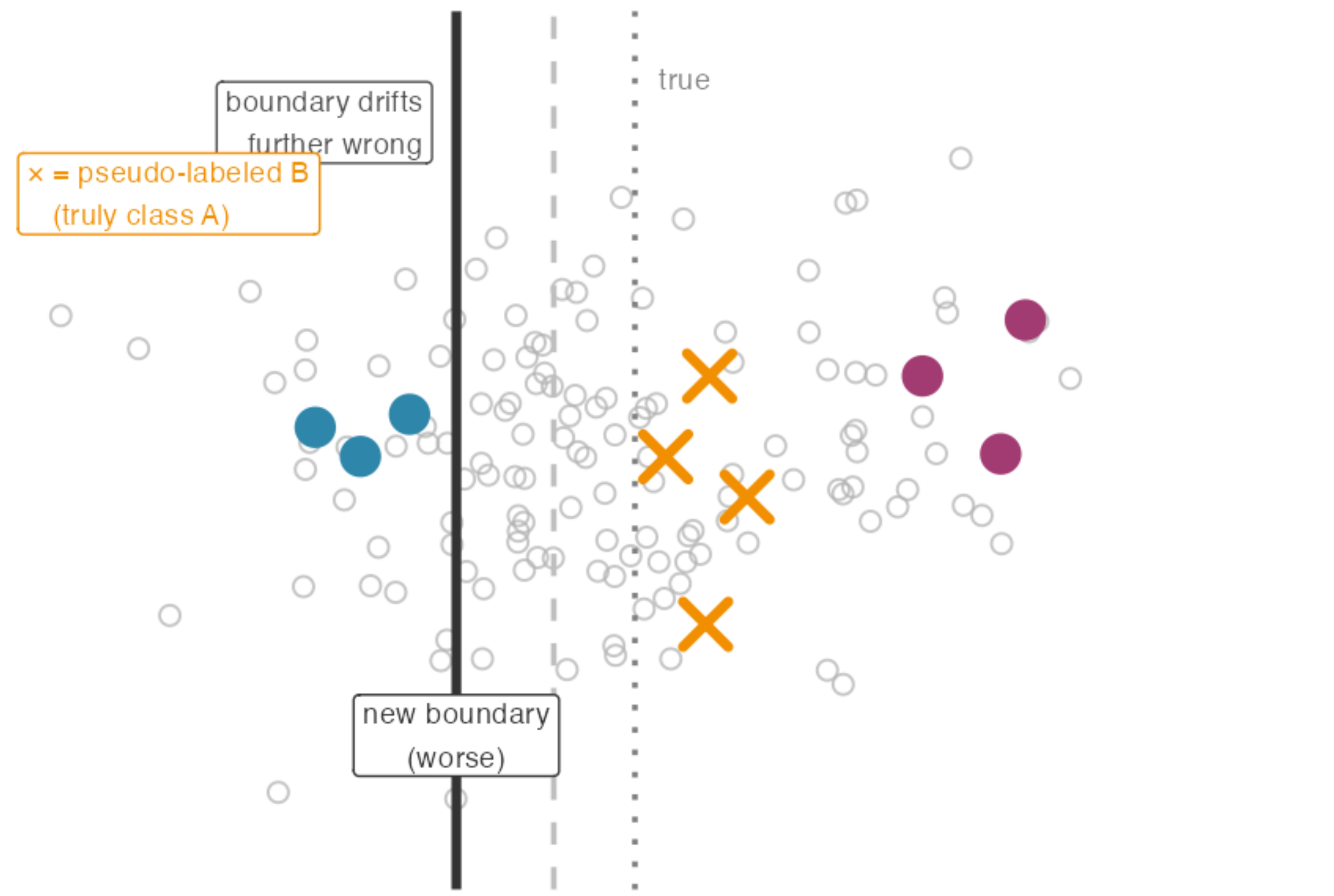
Iteration 1: Initial (wrong) boundary

Orange = confident wrong predictions about to become pseudo-labels



Iteration 2: After retraining on pseudo-labels

Boundary has shifted further from truth — error compounds

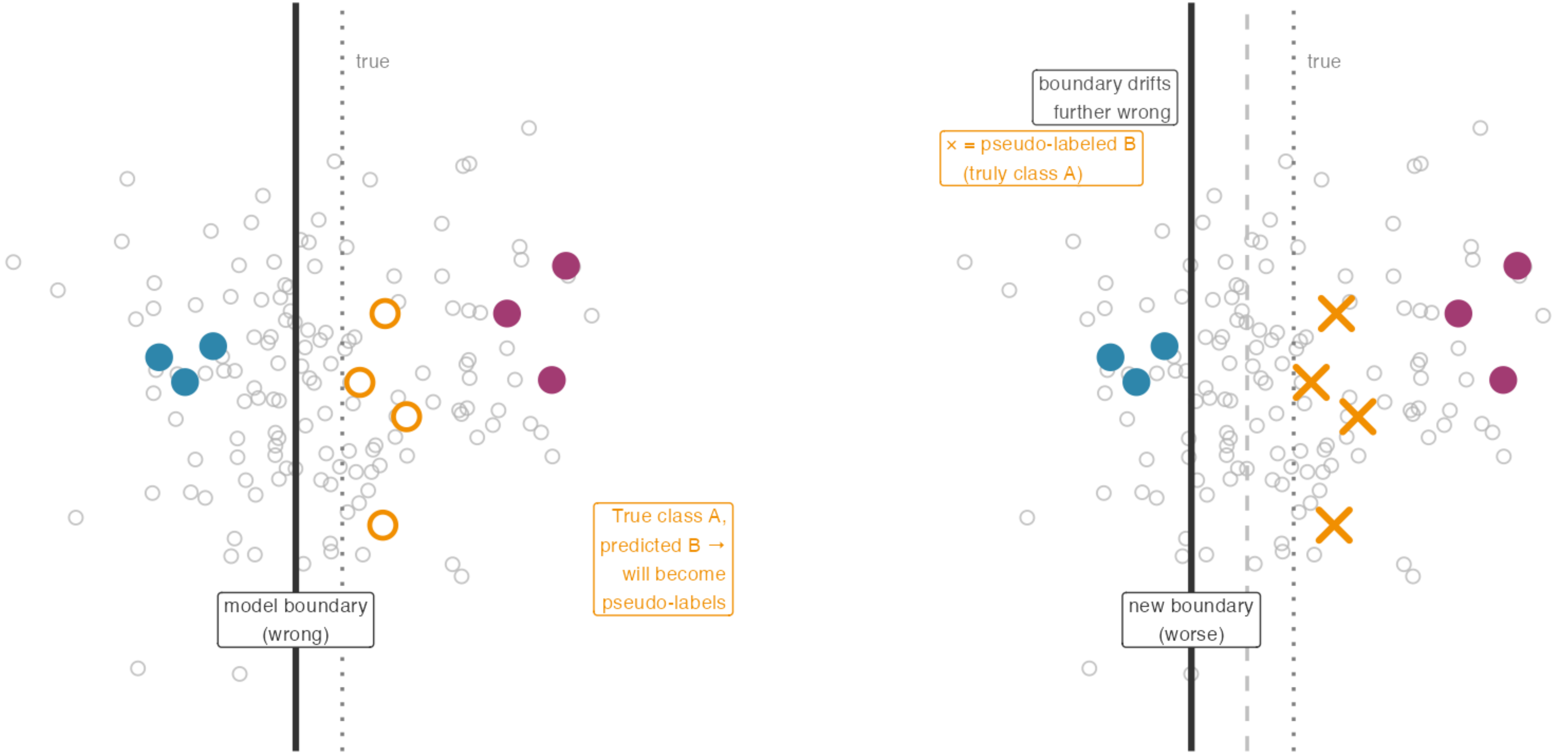


Iteration 1: Initial (wrong) boundary

Orange = confident wrong predictions about to become pseudo-label

Iteration 2: After retraining on pseudo-labels

Boundary has shifted further from truth — error compounds



Mitigating Confirmation Bias

Mitigating Confirmation Bias

- Use a **higher confidence threshold** (only very confident predictions)

Mitigating Confirmation Bias

- Use a **higher confidence threshold** (only very confident predictions)
- Limit the **number of iterations** (don't let error compound too much)

Mitigating Confirmation Bias

- Use a **higher confidence threshold** (only very confident predictions)
- Limit the **number of iterations** (don't let error compound too much)
- Use an **ensemble** of models for pseudo-labeling

Mitigating Confirmation Bias

- Use a **higher confidence threshold** (only very confident predictions)
- Limit the **number of iterations** (don't let error compound too much)
- Use an **ensemble** of models for pseudo-labeling
- Ensure **classifier consistency** before pseudo-labeling
  - Next section: **Consistency Regularization**

Consistency Regularization

Consistency Regularization Idea

Consistency Regularization Idea

- **Smoothness Assumption:** nearby inputs should have the same label

Consistency Regularization Idea

- **Smoothness Assumption:** nearby inputs should have the same label
- Related idea: if you **slightly perturb** an input, the output **shouldn't change**
  - Take an **unlabeled input** x and **perturb** to get $\tilde{x}$
  - Require that $f(x) = f(\tilde{x}) = \hat{y}$ (original and augment give the **same output**)
  - **NOT necessary** that $\hat{y} = y$ (the ground truth)

Consistency Regularization Idea

- **Smoothness Assumption:** nearby inputs should have the same label
- Related idea: if you **slightly perturb** an input, the output **shouldn't change**
  - Take an **unlabeled input** x and **perturb** to get $\tilde{x}$
  - Require that $f(x) = f(\tilde{x}) = \hat{y}$ (original and augment give the **same output**)
  - **NOT necessary** that $\hat{y} = y$ (the ground truth)
- Consistency **training objective:** $\mathscr{L} = \mathscr{L}_{\text{supervised}}(L) + \lambda \cdot \mathscr{L}_{\text{consistency}}(U)$
  - Consistency term **penalizes** the model for **changing predictions**

Mean Teacher

Mean Teacher

- Train two models: **student and teacher**

Mean Teacher

- Train two models: **student and teacher**
- Student trained normally but with **consistency loss**

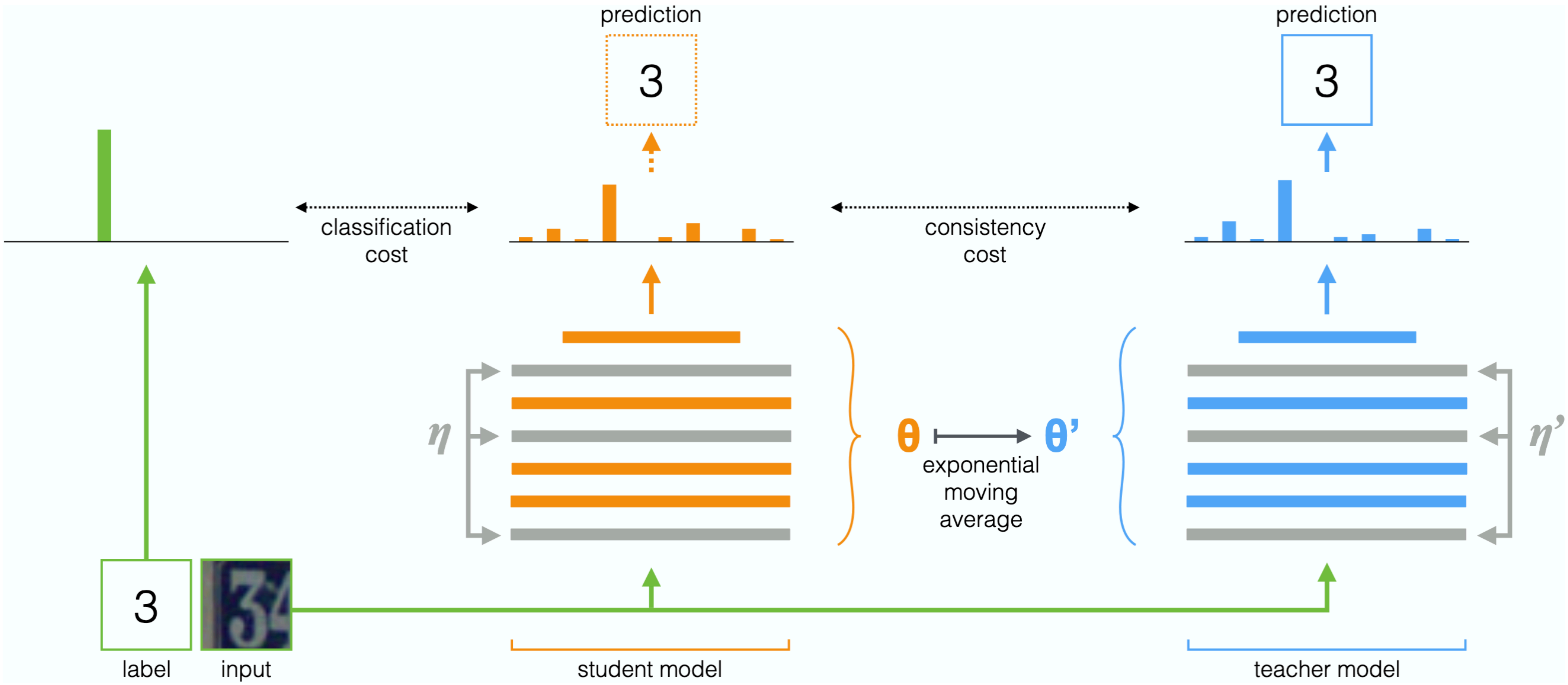
Mean Teacher

- Train two models: **student and teacher**
- Student trained normally but with **consistency loss**
- Teacher is **not trained directly:** it's an (Exponential) **Moving Average** (EMA) of the student model's parameters

Mean Teacher

- Train two models: **student and teacher**
- Student trained normally but with **consistency loss**
- Teacher is **not trained directly:** it's an (Exponential) **Moving Average** (EMA) of the student model's parameters
- Student's consistency loss is **based on the teacher**
  - Student's prediction should **match the teacher**
  - Intuition: teacher is a **more stable version** of the same model
  - **Discourages** the student from **chasing noisy labels**

Mean Teacher



Tarvainen and Valpola (2018)

FixMatch

FixMatch

- **Combines** consistency regularization with pseudo-labeling

FixMatch

- **Combines** consistency regularization with pseudo-labeling
- Generate a pseudo-label from a **weakly-augmented** unlabeled input
  - Only used if **confidence is high,** identifying a **high-quality label**

FixMatch

- **Combines** consistency regularization with pseudo-labeling
- Generate a pseudo-label from a **weakly-augmented** unlabeled input
  - Only used if **confidence is high,** identifying a **high-quality label**
- Train the model to **predict the same pseudo-label** on a **strongly-augmented** version of the same input (a hard task)
  - This forces **invariance** that we want the model to have

FixMatch

- **Combines** consistency regularization with pseudo-labeling
- Generate a pseudo-label from a **weakly-augmented** unlabeled input
  - Only used if **confidence is high,** identifying a **high-quality label**
- Train the model to **predict the same pseudo-label** on a **strongly-augmented** version of the same input (a hard task)
  - This forces **invariance** that we want the model to have
- Similar to **self-supervised techniques** we saw last week

FixMatch

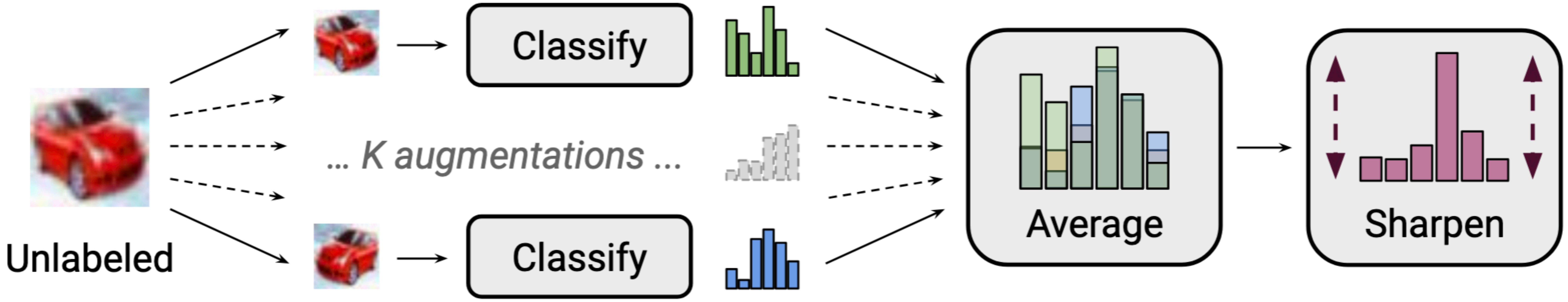


Figure 1: Diagram of the label guessing process used in MixMatch. Stochastic data augmentation is applied to an unlabeled image K times, and each augmented image is fed through the classifier. Then, the average of these K predictions is "sharpened" by adjusting the distribution's temperature. See algorithm 1 for a full description.