Carnegie Mellon University
HeinzCollege
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

Deep Kernel Aalen-Johansen Estimator: An Interpretable and Flexible Neural Net Framework for Competing Risks

Xiaobin Shen*
PhD Student
Carnegie Mellon University

George H. Chen*,😃
Associate Professor
Carnegie Mellon University

*equal contribution

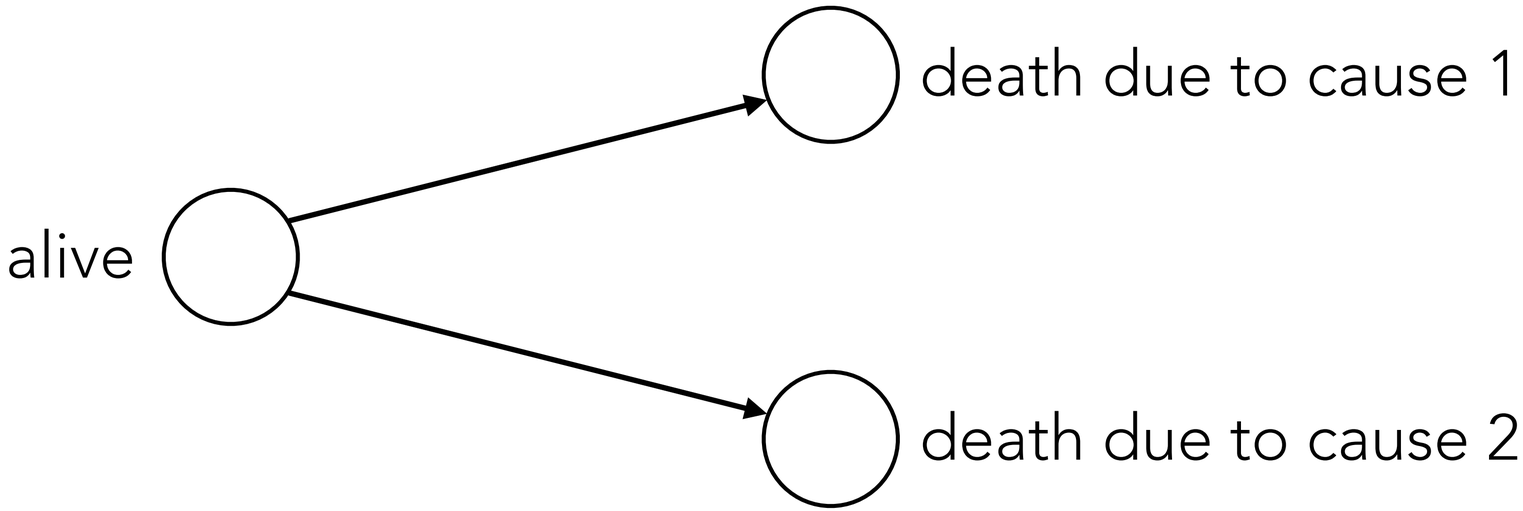
😃today's presenter

Published at *Machine Learning for Health* (ML4H) 2025

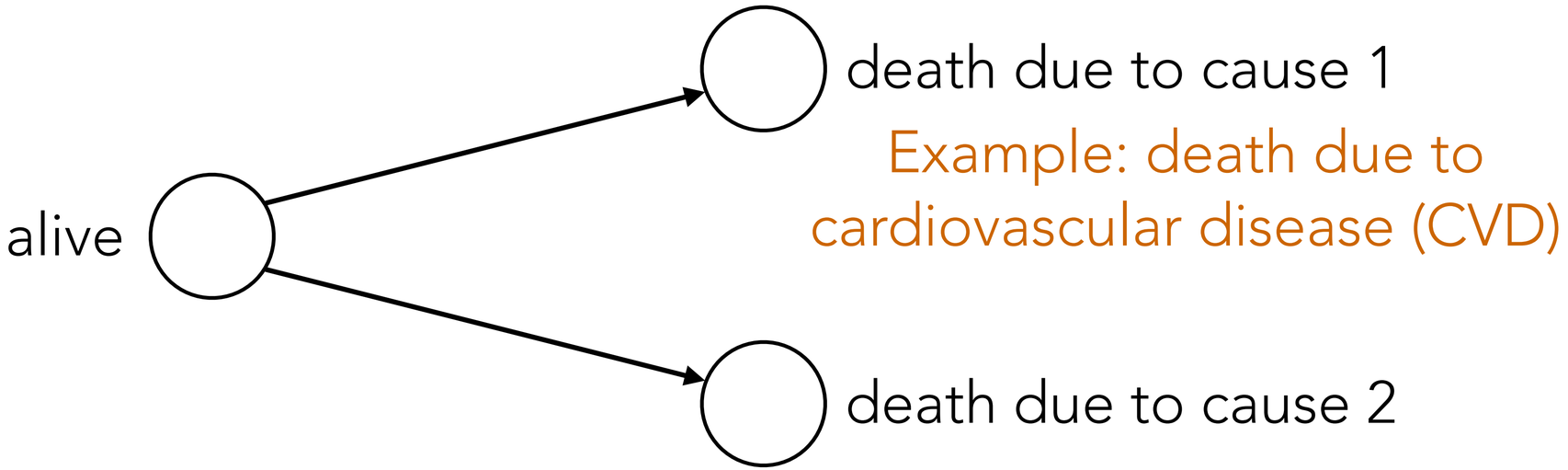
IMSI Advances in Quantitative Medical Care — February 2, 2026

Competing Risks (Simple Case)

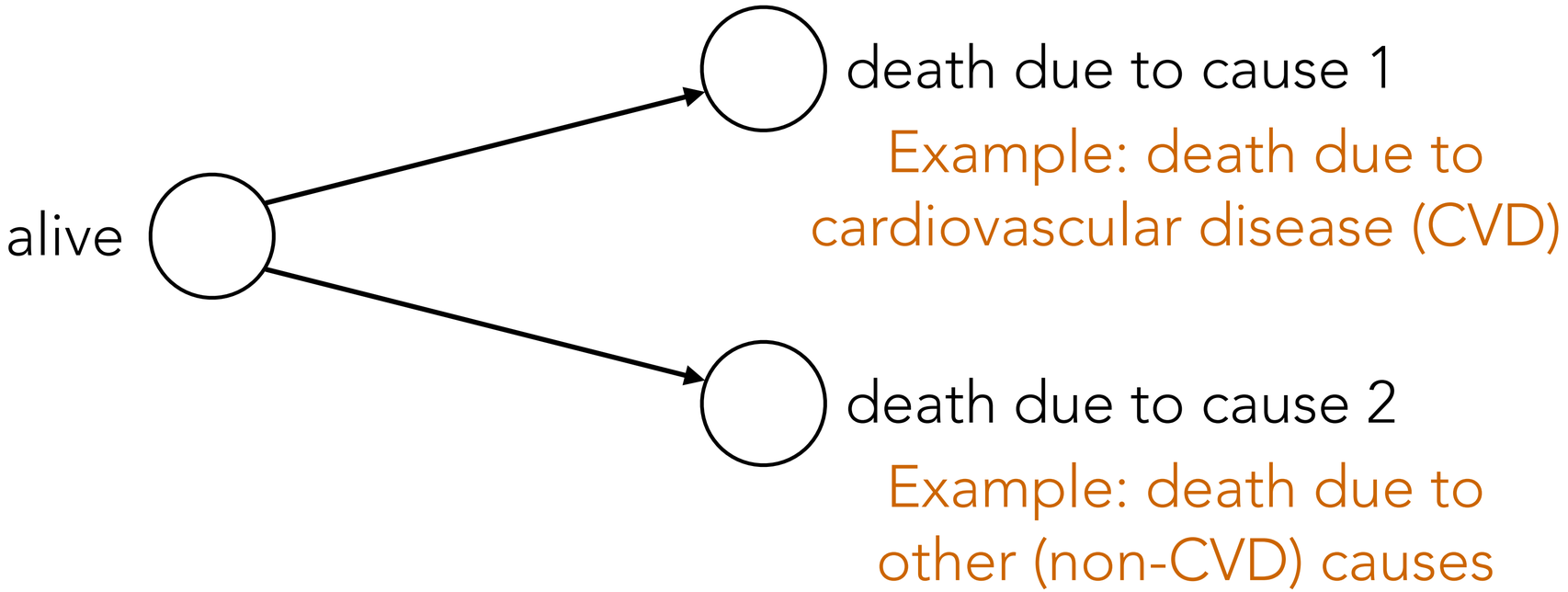
Competing Risks (Simple Case)



Competing Risks (Simple Case)

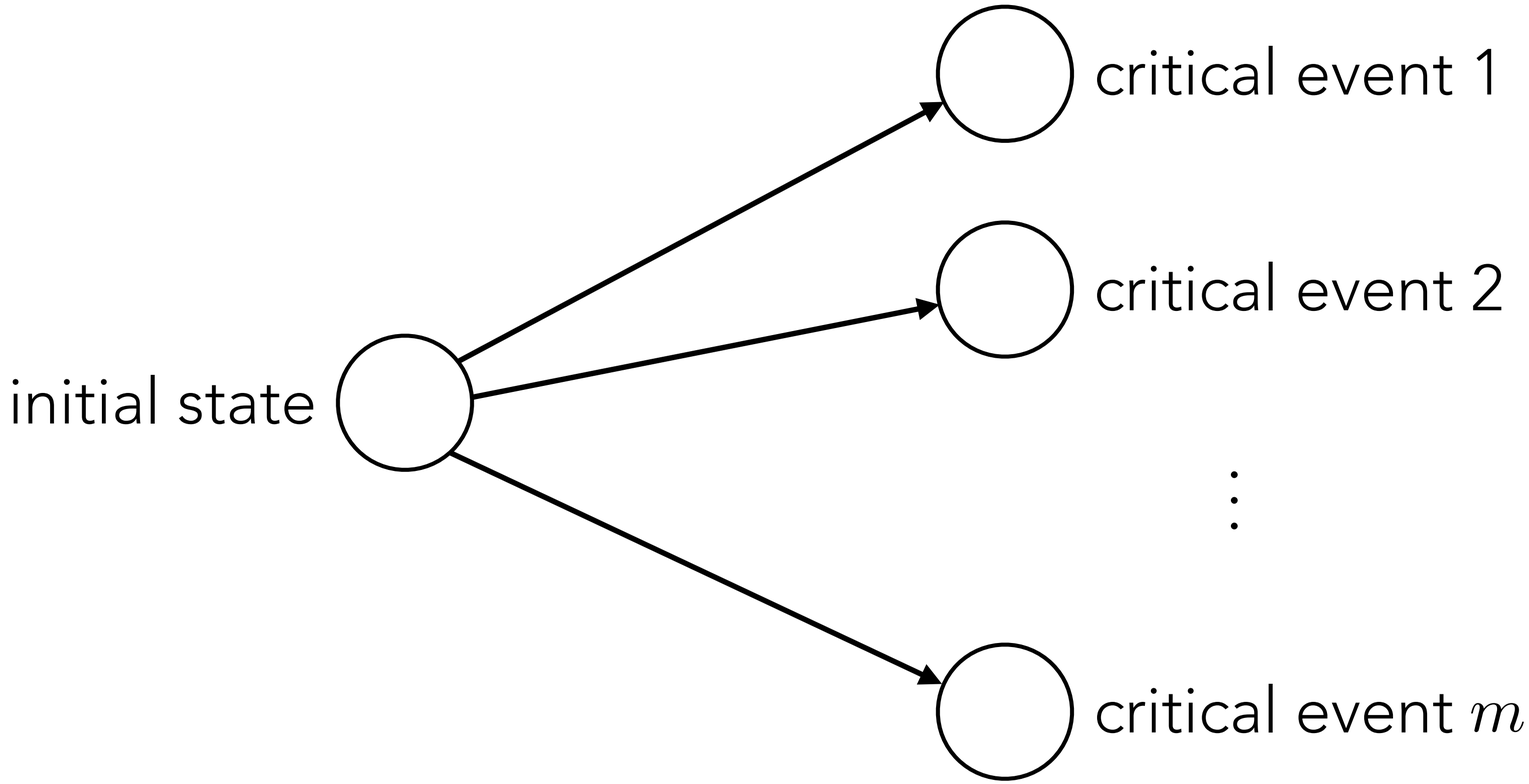


Competing Risks (Simple Case)

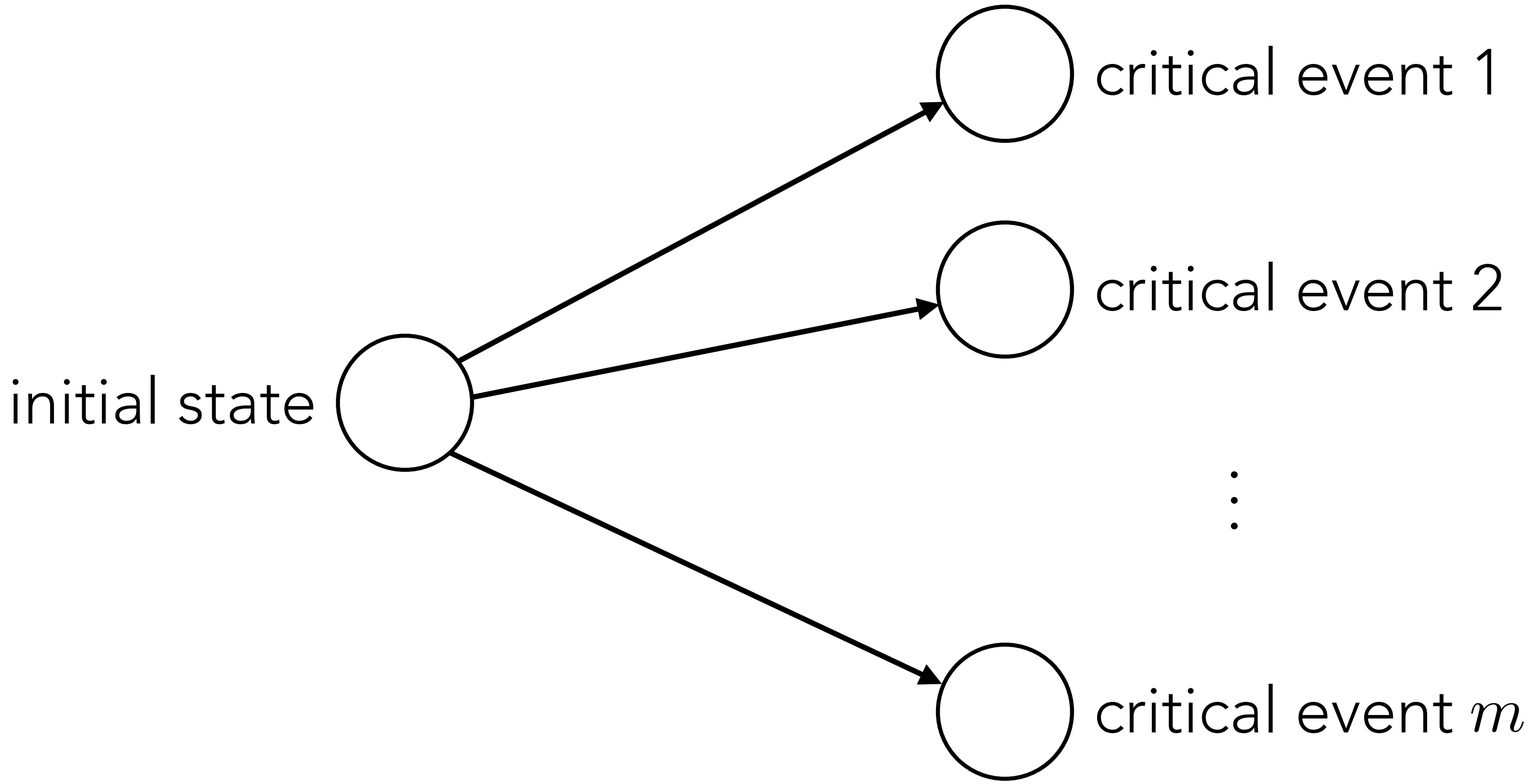


Competing Risks (General Case)

Competing Risks (General Case)

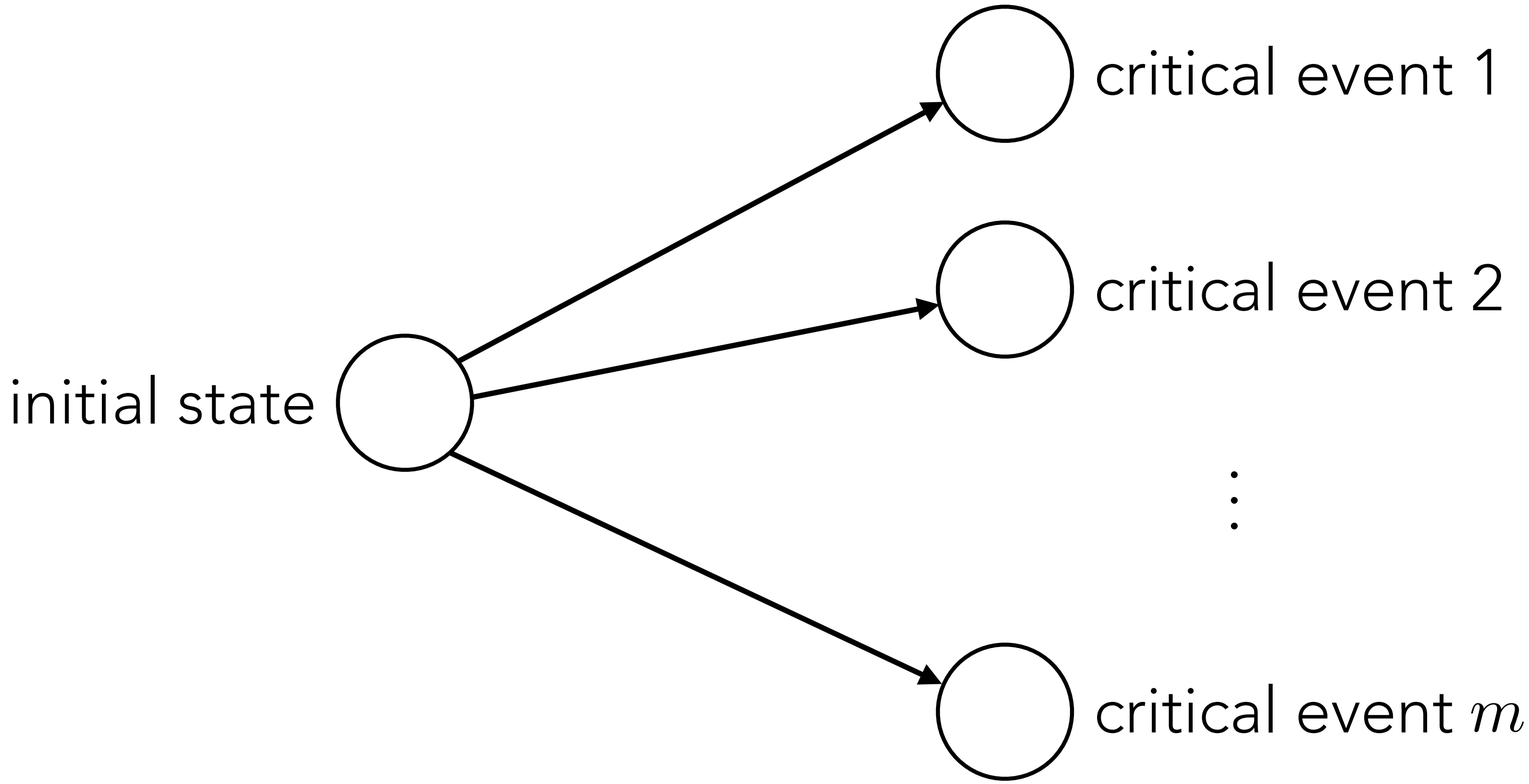


Competing Risks (General Case)



Want to reason about <u>which critical event will happen earliest</u> and <u>when</u>

Competing Risks (General Case)



Want to reason about <u>which critical event will happen earliest</u> and <u>when</u>

Note: standard survival analysis is a special case where $m = 1$

What Our Paper is About

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

by design & not via post hoc explanation tool

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

by design & not via post hoc explanation tool

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]
  - Competitive against various baselines

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

by design & not via post hoc explanation tool

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]
  - Competitive against various baselines
  - Provides a different notion of model interpretation compared to Fine & Gray [1999] or cause-specific Cox models [Prentice et al 1978]

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

by design & not via post hoc explanation tool

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]
  - Competitive against various baselines
  - Provides a different notion of model interpretation compared to Fine & Gray [1999] or cause-specific Cox models [Prentice et al 1978]

each data point is represented as a weighted combination of clusters

What Our Paper is About

- The ML community has recently developed many competing risks models (**DeepHit** [Lee et al 2018], **DSM** [Nagpal et al 2021], **DeSurv** [Danks & Yau 2022], **SurvivalBoost** [Alberge et al 2025], …)
  - The focus has largely been on prediction performance & *not* on interpretability

by design & not via post hoc explanation tool

- This paper: new interpretable deep competing risks model based on the classical Aalen-Johansen (AJ) estimator [1978]
  - Competitive against various baselines
  - Provides a different notion of model interpretation compared to Fine & Gray [1999] or cause-specific Cox models [Prentice et al 1978]

each data point is represented as a weighted combination of clusters

If a data point only has nonzero weight for 1 cluster
⇒ prediction corresponds to AJ estimator fitted to only data points from that cluster

Outline

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

  Technically I'll only be presenting the AJ estimator applied to competing risks
  (the AJ estimator more generally is for multistate processes)

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

  Technically I'll only be presenting the AJ estimator applied to competing risks (the AJ estimator more generally is for multistate processes)

- How to go from the AJ estimator (population level) to a *kernel* AJ estimator (individual level)

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

  Technically I'll only be presenting the AJ estimator applied to competing risks (the AJ estimator more generally is for multistate processes)

- How to go from the AJ estimator (population level) to a *kernel* AJ estimator (individual level)

- How to parameterize the kernel function as a neural net

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

  Technically I'll only be presenting the AJ estimator applied to competing risks (the AJ estimator more generally is for multistate processes)

- How to go from the AJ estimator (population level) to a *kernel* AJ estimator (individual level)
- How to parameterize the kernel function as a neural net
- The full deep kernel Aalen-Johansen (DKAJ) estimator

Outline

- Background: competing risks setup & the Aalen-Johansen (AJ) estimator

  Technically I'll only be presenting the AJ estimator applied to competing risks (the AJ estimator more generally is for multistate processes)

- How to go from the AJ estimator (population level) to a *kernel* AJ estimator (individual level)
- How to parameterize the kernel function as a neural net
- The full deep kernel Aalen-Johansen (DKAJ) estimator
- Numerical experiments

Competing Risks Problem Setup

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$

feature vector

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$

feature vector

$\Delta_i \in \{0, 1, \ldots, m\}$

indicates which event happened earliest
(0 means never left initial state, i.e., censored)

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

Common prediction task:

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time $t \mid x$)

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is $\delta \in \{1, 2, \ldots, m\}$ and happens within time t | x)

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

$\delta \in \{1, 2, \ldots, m\}$

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time $t \mid x$)

cumulative incidence function (CIF)

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

$\delta \in \{1, 2, \ldots, m\}$

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time t | x)

cumulative incidence function (CIF)

$$F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$$

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

$\delta \in \{1, 2, \ldots, m\}$

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time t | x)

cumulative incidence function (CIF)

$$F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$$

Δ^* refers to the earliest event type

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

$\delta \in \{1, 2, \ldots, m\}$

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time t | x)

cumulative incidence function (CIF)

$$F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$$

Δ^* refers to the earliest event type

T refers to the time of the earliest event

Competing Risks Problem Setup

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

$X_i \in \mathbb{R}^d$
feature vector

$Y_i \in [0, \infty)$
is the time until the earliest event
(or censoring)

$\Delta_i \in \{0, 1, \ldots, m\}$
indicates which event happened earliest
(0 means never left initial state, i.e., censored)

$\delta \in \{1, 2, \ldots, m\}$

Common prediction task:

For test feature vector x, estimate: $\mathbb{P}$(earliest event is δ and happens within time t | x)

cumulative incidence function (CIF)

$$F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$$

Δ^* refers to the earliest event type

T refers to the time of the earliest event

AJ estimator estimates population-level version

$$F_\delta^{\text{pop}}(t) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t\big)$$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened					

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$				

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$			

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$		

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

Intuition: $F_\delta^{\text{pop}}(t) \approx \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \mathbb{P}\left(\begin{smallmatrix}\text{experience critical event } \delta \\ \text{at time } t_\ell\end{smallmatrix}\right)$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

Intuition: $$F_\delta^{\text{pop}}(t) \approx \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \mathbb{P}\left(\begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right) \mathbb{P}\left(\begin{smallmatrix}\text{experience event type } \delta \\ \text{at time } t_\ell\end{smallmatrix} \;\middle|\; \begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right)$$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

Intuition: $F_\delta^{\text{pop}}(t) \approx \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \mathbb{P}\left(\begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right) \mathbb{P}\left(\begin{smallmatrix}\text{experience event type } \delta \\ \text{at time } t_\ell\end{smallmatrix} \;\middle|\; \begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right)$

$\approx \widehat{S}^{\text{KM}}(t_{\ell-1})$

classical Kaplan-Meier estimator [1958]

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

Intuition: $F_\delta^{\text{pop}}(t) \approx \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \mathbb{P}\left(\begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right) \mathbb{P}\left(\begin{smallmatrix}\text{experience event type } \delta \\ \text{at time } t_\ell\end{smallmatrix} \,\middle|\, \begin{smallmatrix}\text{earliest event} \\ \text{after time } t_{\ell-1}\end{smallmatrix}\right)$

$\approx \widehat{S}^{\text{KM}}(t_{\ell-1})$

classical Kaplan-Meier estimator [1958]

$\approx \frac{d_{\delta,\ell}}{n_\ell}$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

3. Output: $\widehat{F}_{\delta}^{\mathrm{AJ}}(t) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KM}}(t_{\ell-1}) \frac{d_{\delta,\ell}}{n_\ell}$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

AJ Estimator for Competing Risks [Aalen-Johansen 1978]

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$\cdots$	$d_{1,L}$
# times event 2 happened	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$\cdots$	$d_{2,L}$
$\vdots$					
# times event m happened	$d_{m,1}$	$d_{m,2}$	$d_{m,3}$	$\cdots$	$d_{m,L}$
# at risk	n_1	n_2	n_3	$\cdots$	n_L

3. Output: $\widehat{F}_\delta^{\mathrm{AJ}}(t) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}^{\mathrm{KM}}(t_{\ell-1}) \frac{d_{\delta,\ell}}{n_\ell}$

$$\widehat{S}^{\mathrm{KM}}(t) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}}{n_\ell}\right)$$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

Kernel Aalen-Johansen Estimator

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened	$d_{1,1}(x)$	$d_{1,2}(x)$	$d_{1,3}(x)$	$\cdots$	$d_{1,L}(x)$
# times event 2 happened	$d_{2,1}(x)$	$d_{2,2}(x)$	$d_{2,3}(x)$	$\cdots$	$d_{2,L}(x)$
$\vdots$					
# times event m happened	$d_{m,1}(x)$	$d_{m,2}(x)$	$d_{m,3}(x)$	$\cdots$	$d_{m,L}(x)$
# at risk	$n_1(x)$	$n_2(x)$	$n_3(x)$	$\cdots$	$n_L(x)$

3. Output: $\widehat{F}_{\delta}^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

Training data: $(X_1, Y_1, \Delta_1), (X_2, Y_2, \Delta_2), \ldots, (X_n, Y_n, \Delta_n)$

Kernel Aalen-Johansen Estimator

1. Find all unique times in which any critical event occurred

$$0 < t_1 < t_2 < \cdots < t_L \qquad L = \text{\# unique times of any critical event}$$

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened among patients who look like x	$d_{1,1}(x)$	$d_{1,2}(x)$	$d_{1,3}(x)$	$\cdots$	$d_{1,L}(x)$
# times event 2 happened among patients who look like x	$d_{2,1}(x)$	$d_{2,2}(x)$	$d_{2,3}(x)$	$\cdots$	$d_{2,L}(x)$
$\vdots$					
# times event m happened among patients who look like x	$d_{m,1}(x)$	$d_{m,2}(x)$	$d_{m,3}(x)$	$\cdots$	$d_{m,L}(x)$
# at risk among patients who look like x	$n_1(x)$	$n_2(x)$	$n_3(x)$	$\cdots$	$n_L(x)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened among patients who look like x	$d_{1,1}(x)$	$d_{1,2}(x)$	$d_{1,3}(x)$	$\cdots$	$d_{1,L}(x)$
# times event 2 happened among patients who look like x	$d_{2,1}(x)$	$d_{2,2}(x)$	$d_{2,3}(x)$	$\cdots$	$d_{2,L}(x)$
$\vdots$					
# times event m happened among patients who look like x	$d_{m,1}(x)$	$d_{m,2}(x)$	$d_{m,3}(x)$	$\cdots$	$d_{m,L}(x)$
# at risk among patients who look like x	$n_1(x)$	$n_2(x)$	$n_3(x)$	$\cdots$	$n_L(x)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened among patients who look like x	$d_{1,1}(x)$	$d_{1,2}(x)$	$d_{1,3}(x)$	$\cdots$	$d_{1,L}(x)$
# times event 2 happened among patients who look like x	$d_{2,1}(x)$	$d_{2,2}(x)$	$d_{2,3}(x)$	$\cdots$	$d_{2,L}(x)$
$\vdots$					
# times event m happened among patients who look like x	$d_{m,1}(x)$	$d_{m,2}(x)$	$d_{m,3}(x)$	$\cdots$	$d_{m,L}(x)$
# at risk among patients who look like x	$n_1(x)$	$n_2(x)$	$n_3(x)$	$\cdots$	$n_L(x)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

2. Build table below

	t_1	t_2	t_3	$\cdots$	t_L
# times event 1 happened among patients who look like x	$d_{1,1}(x)$	$d_{1,2}(x)$	$d_{1,3}(x)$	$\cdots$	$d_{1,L}(x)$
# times event 2 happened among patients who look like x	$d_{2,1}(x)$	$d_{2,2}(x)$	$d_{2,3}(x)$	$\cdots$	$d_{2,L}(x)$
$\vdots$					
# times event m happened among patients who look like x	$d_{m,1}(x)$	$d_{m,2}(x)$	$d_{m,3}(x)$	$\cdots$	$d_{m,L}(x)$
# at risk among patients who look like x	$n_1(x)$	$n_2(x)$	$n_3(x)$	$\cdots$	$n_L(x)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

Example kernel functions:

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

Example kernel functions:

$K(x, x') = 1$ for all x, x'

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

Example kernel functions:

$K(x, x') = 1$ for all x, x' $\Rightarrow$ recover classical (population-level) AJ estimator

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

Example kernel functions:

$K(x, x') = 1$ for all x, x' $\Rightarrow$ recover classical (population-level) AJ estimator

$K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

kernel function parameterized by an "encoder" neural net $f(\cdot;\theta)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

$$K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$$

kernel function parameterized by an "encoder" neural net $f(\cdot;\theta)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

Can learn neural net parameters using maximum likelihood (train in minibatches via minibatch gradient descent)

$$K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$$

kernel function parameterized by an "encoder" neural net $f(\cdot;\theta)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

DKAJ High-Level Idea

DKAJ High-Level Idea

$$f(\cdot;\theta)$$

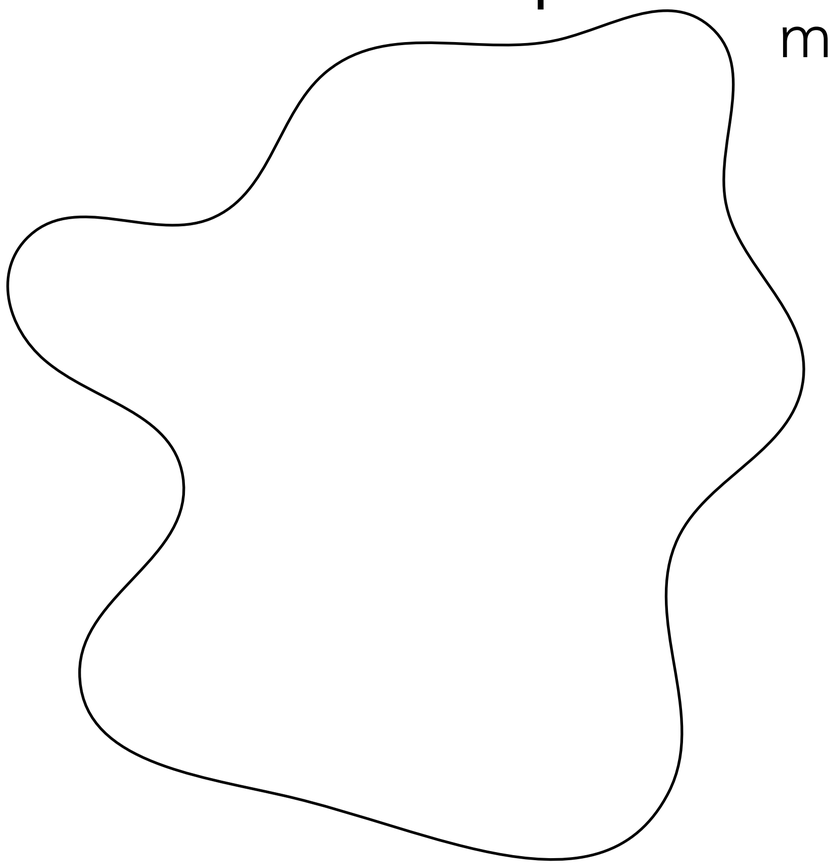
DKAJ High-Level Idea

encoder learned via
maximum likelihood

$f(\cdot;\widehat{\theta})$

DKAJ High-Level Idea

Raw feature space

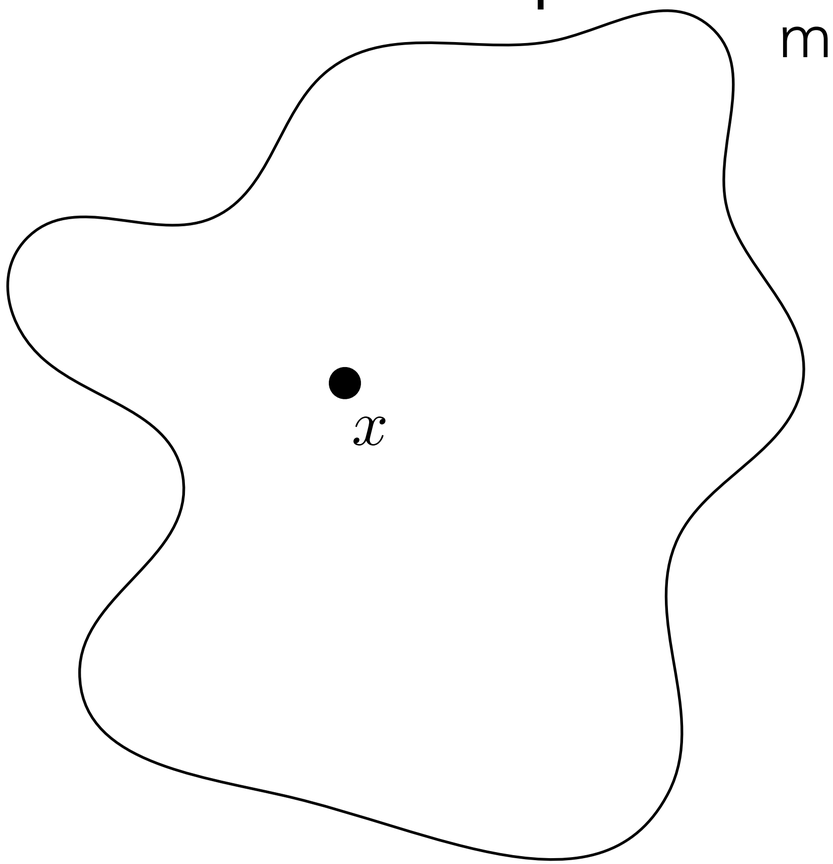


encoder learned via maximum likelihood

$$f(\cdot;\widehat{\theta})$$

DKAJ High-Level Idea

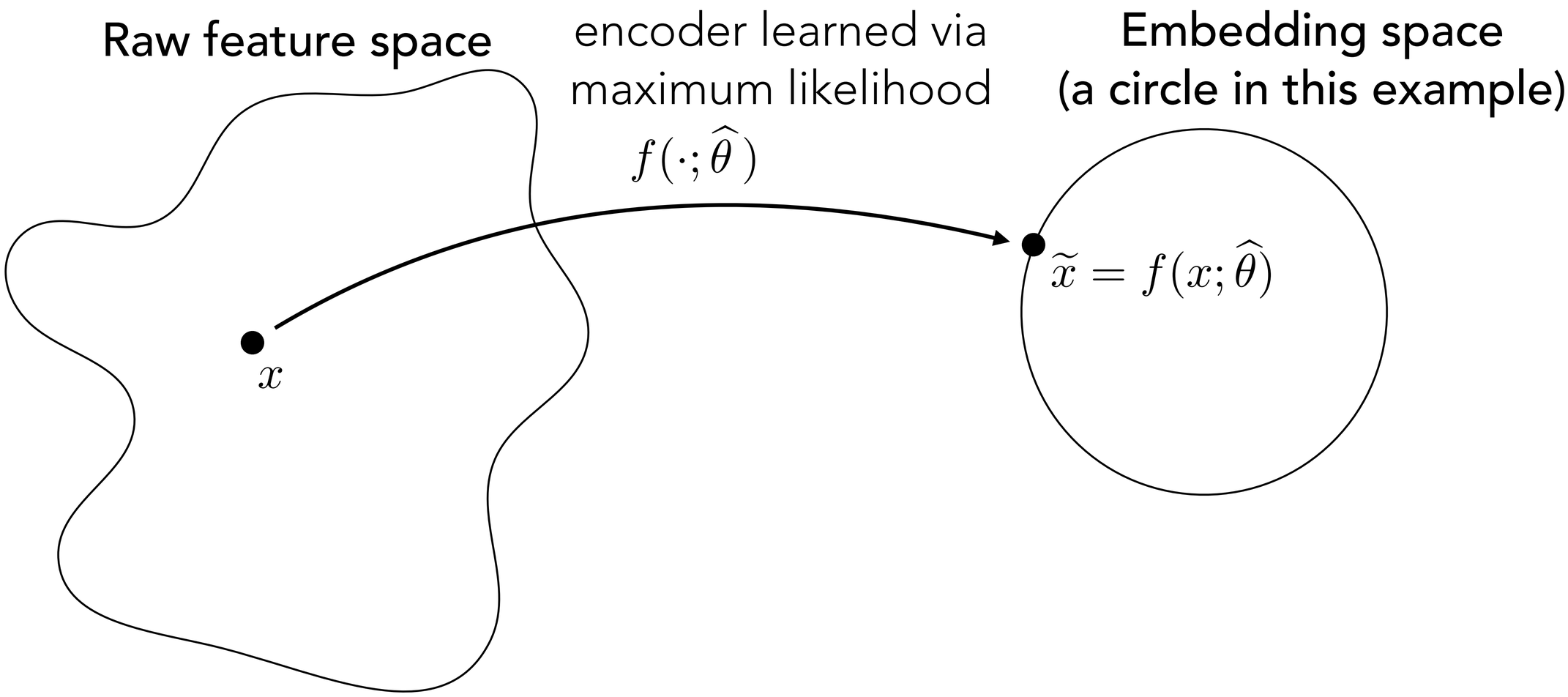
Raw feature space



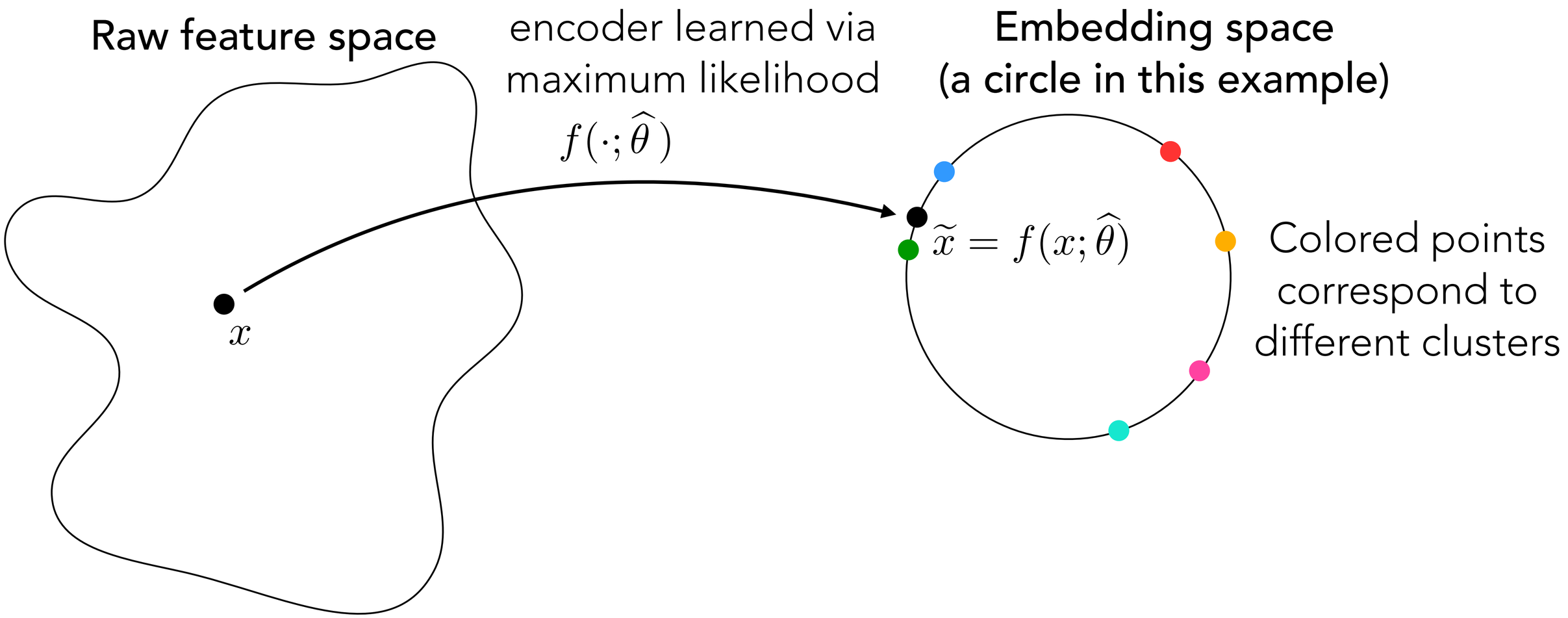
encoder learned via maximum likelihood

$f(\cdot;\widehat{\theta})$

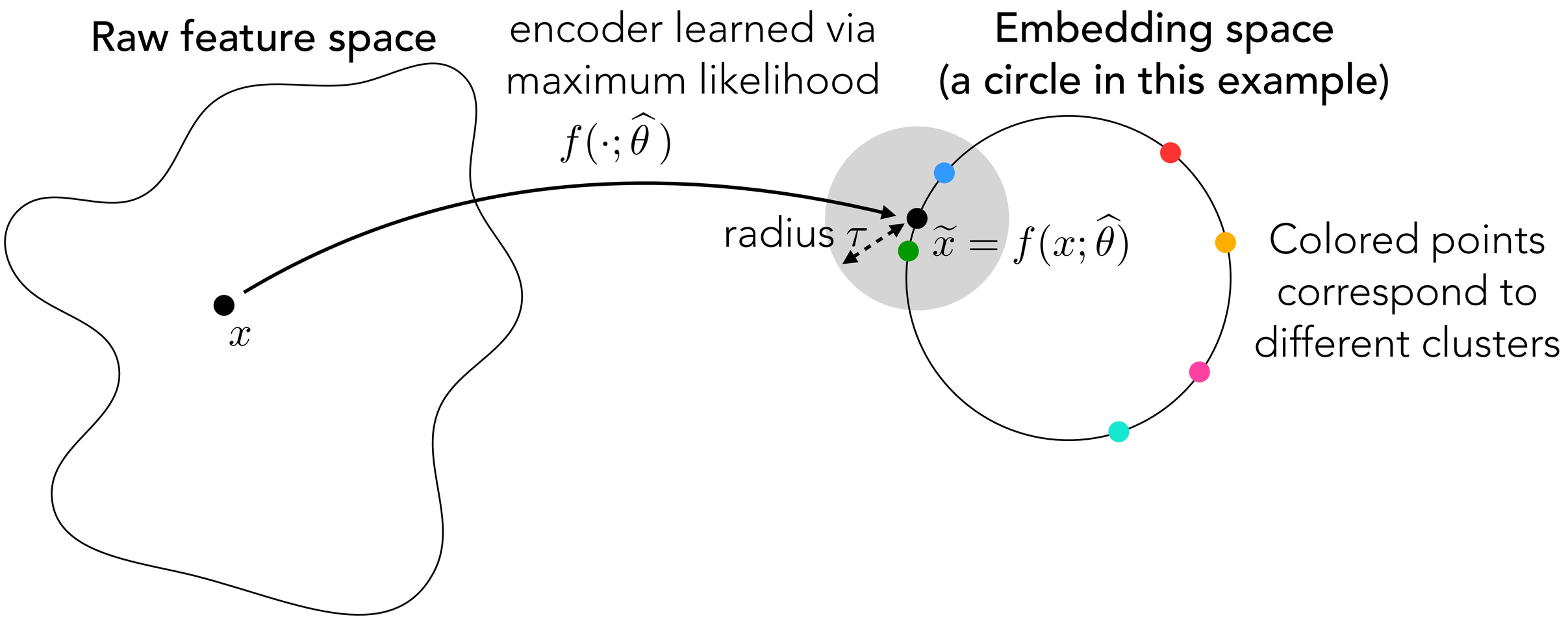
DKAJ High-Level Idea



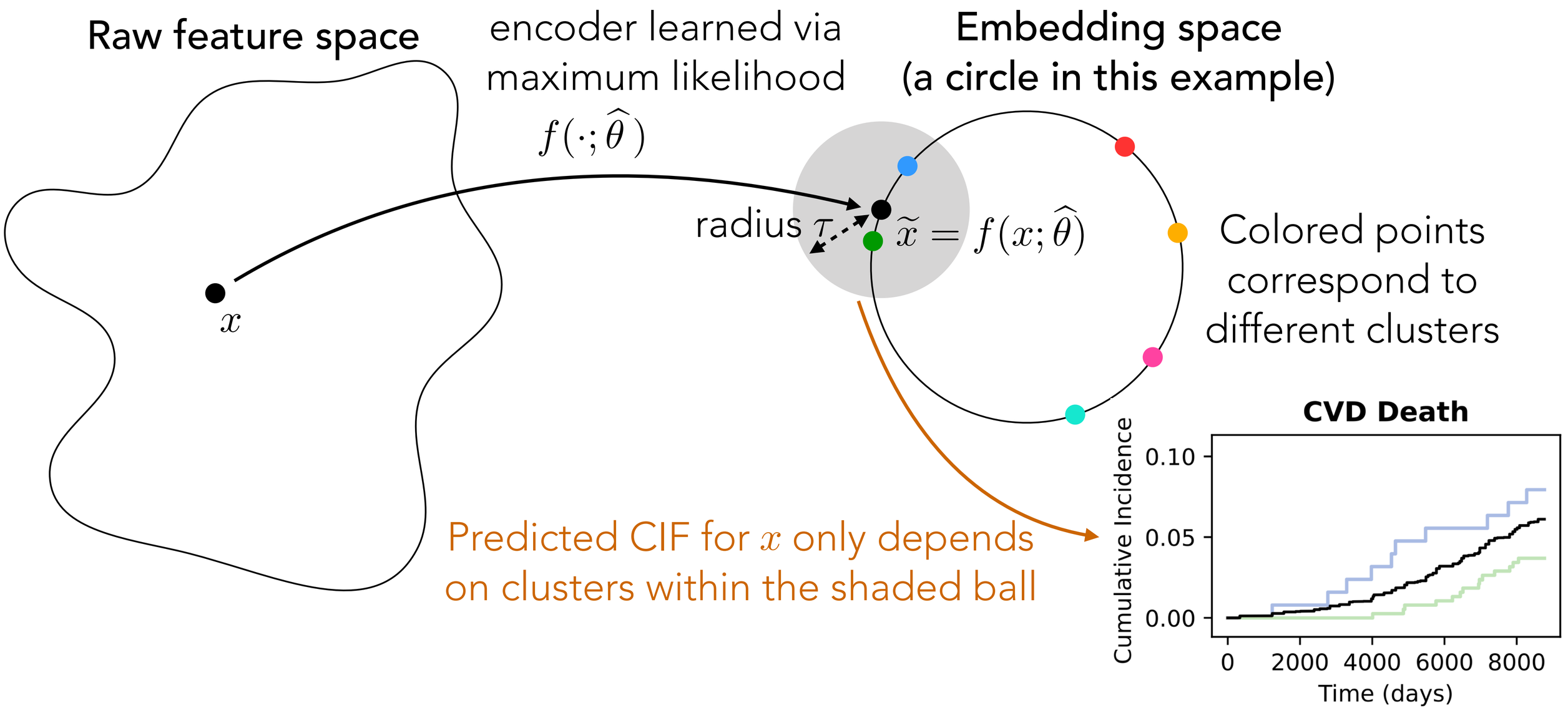
DKAJ High-Level Idea



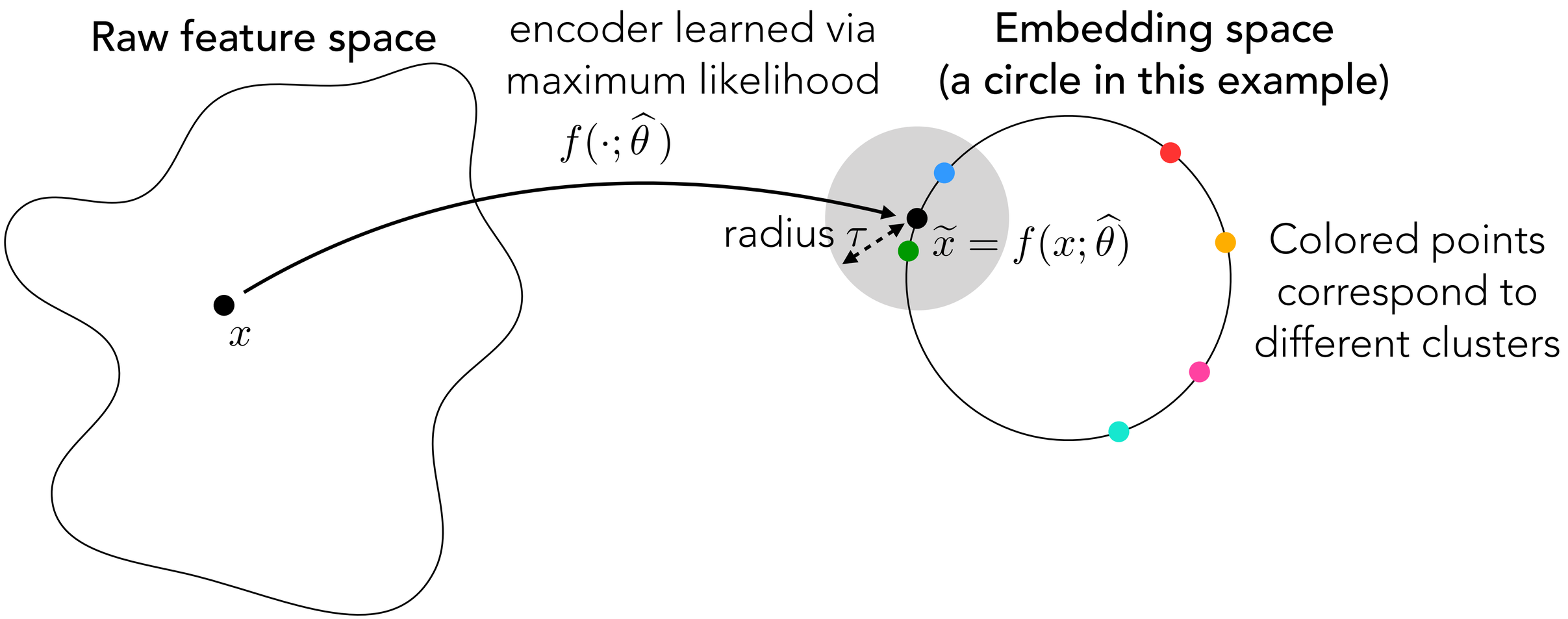
DKAJ High-Level Idea



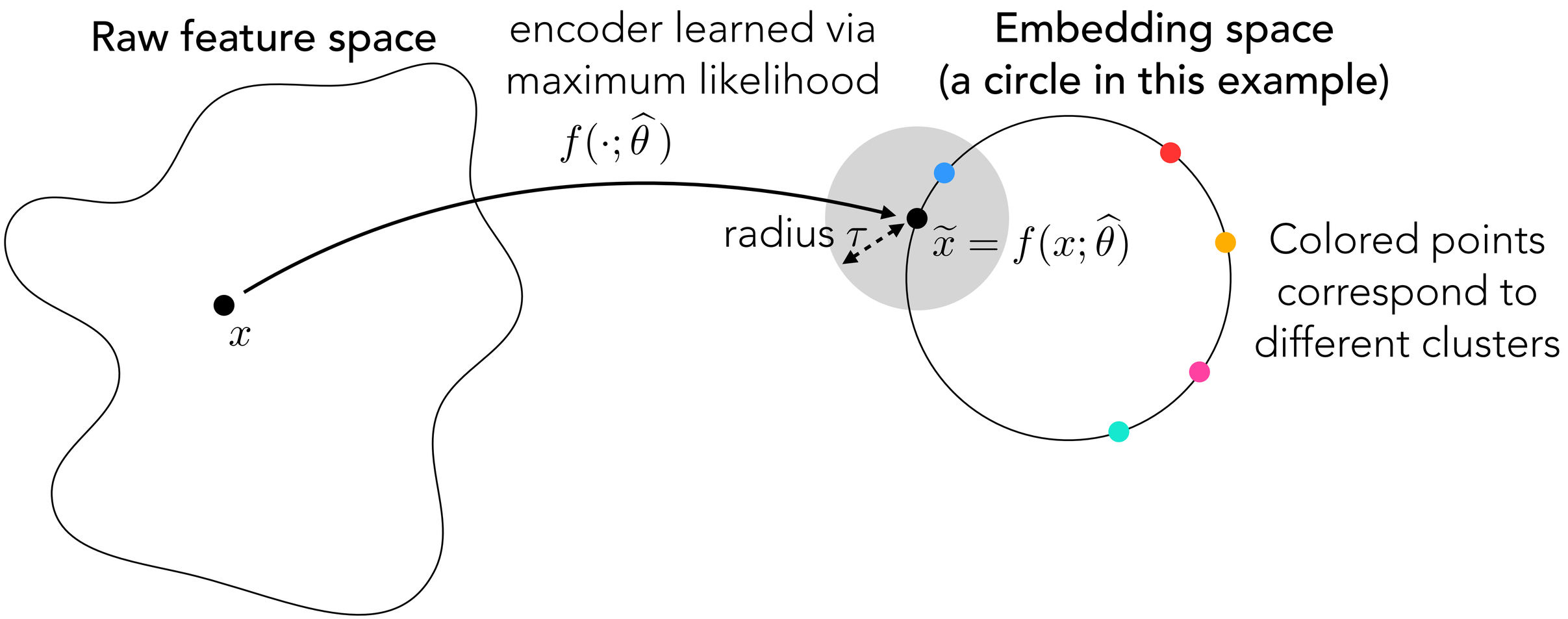
DKAJ High-Level Idea



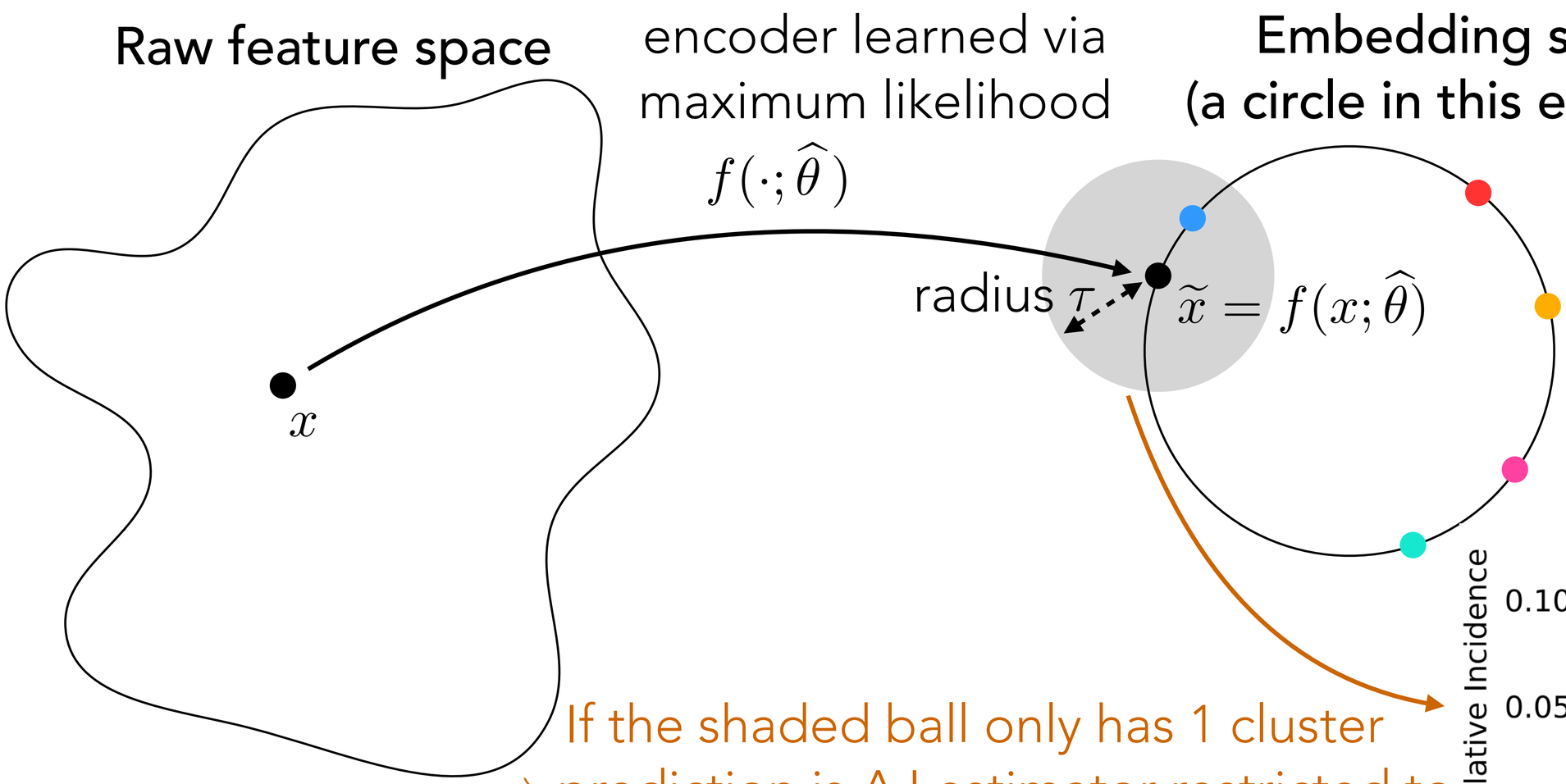
DKAJ High-Level Idea

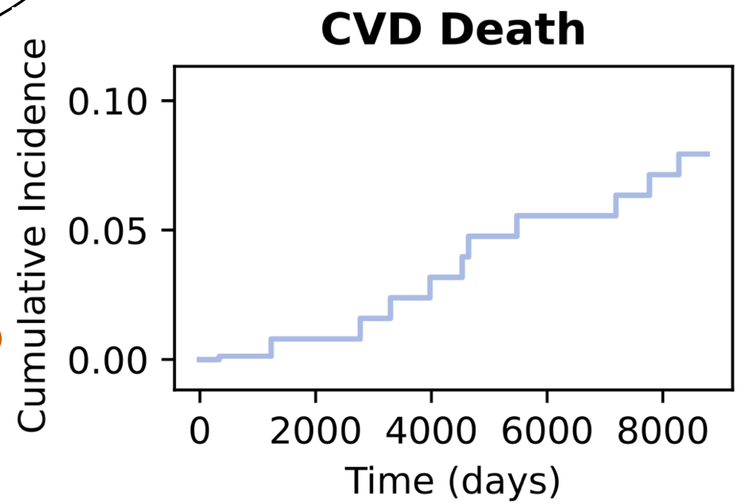


DKAJ High-Level Idea



DKAJ High-Level Idea





Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training

Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

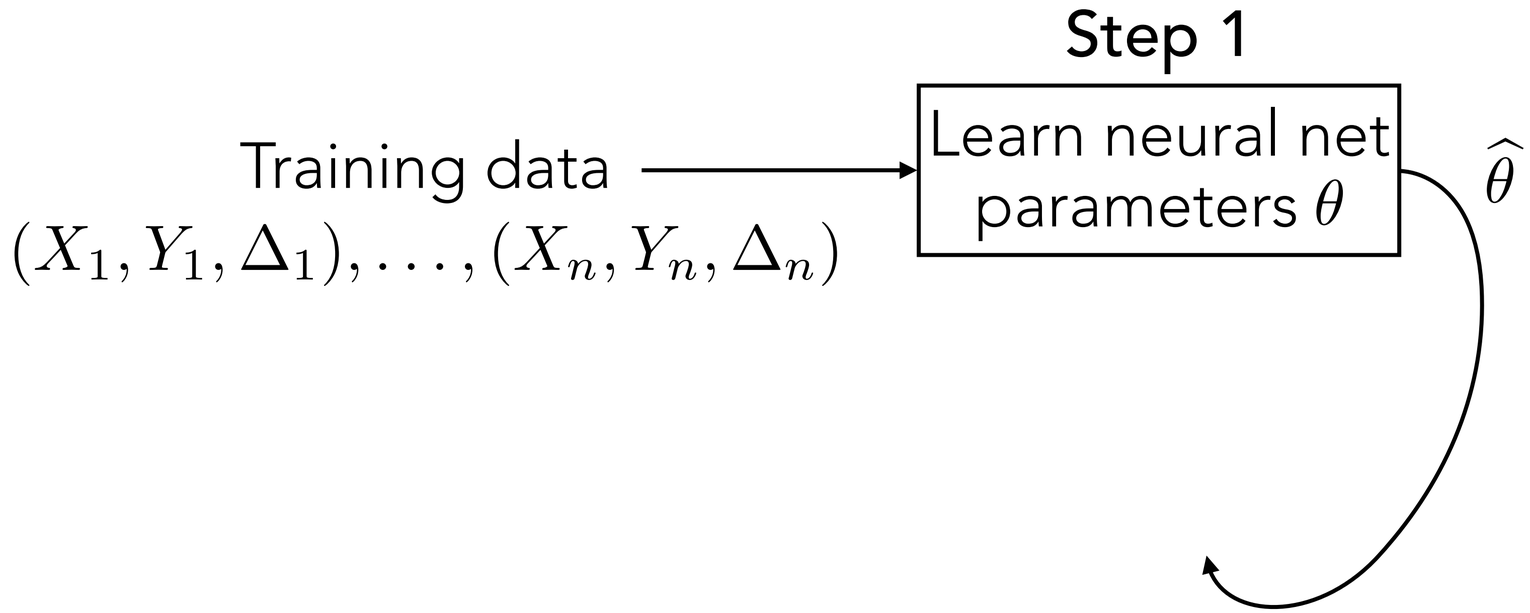
DKAJ Training

Training data

$(X_1, Y_1, \Delta_1), \ldots, (X_n, Y_n, \Delta_n)$

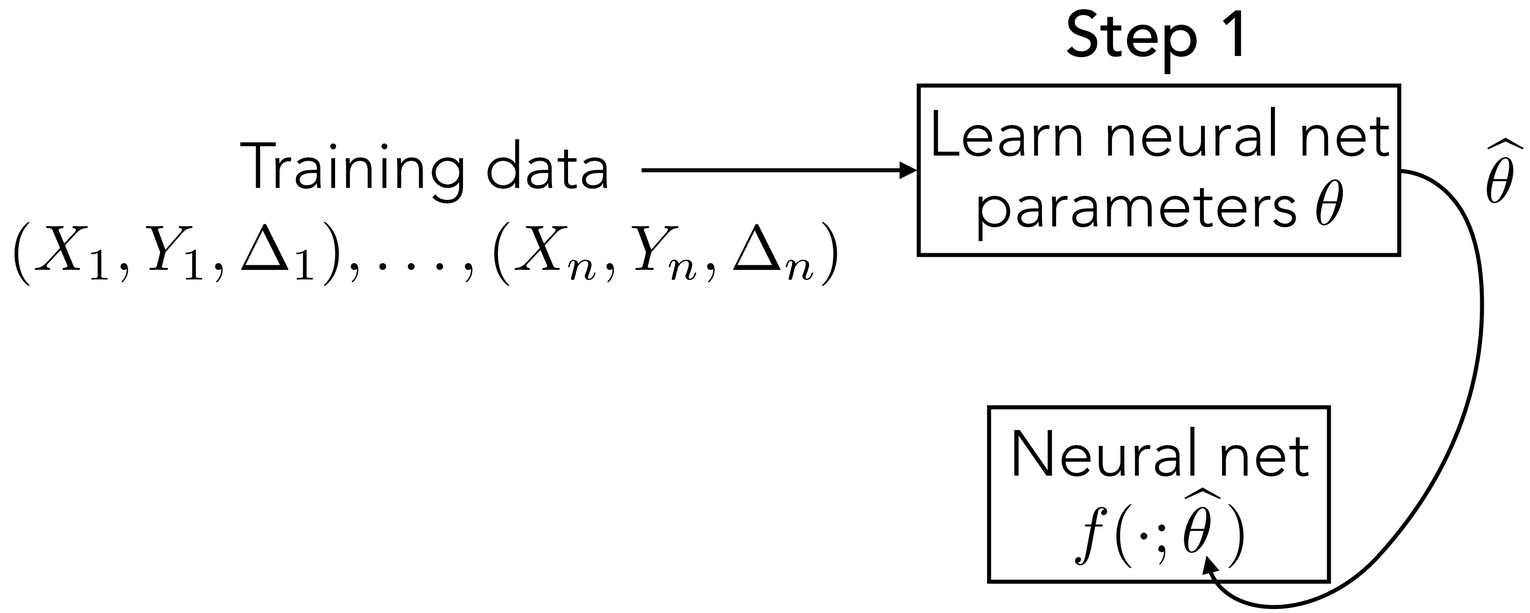
Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



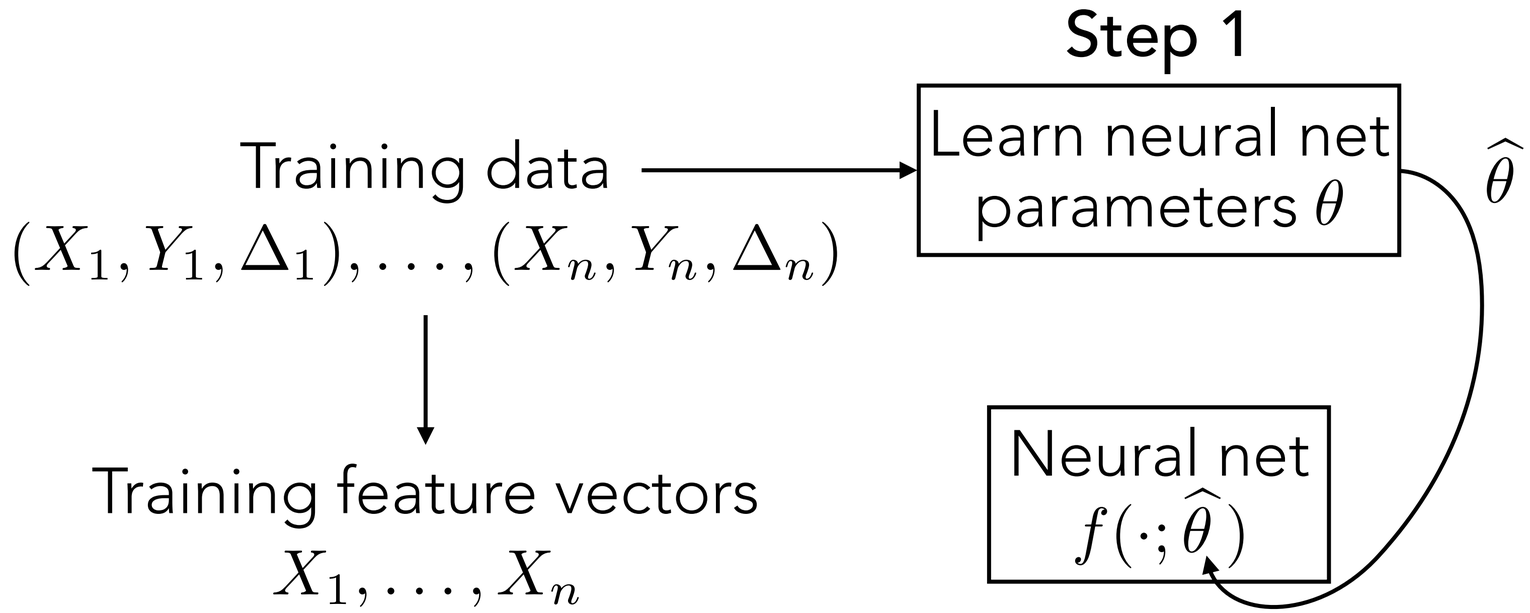
Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



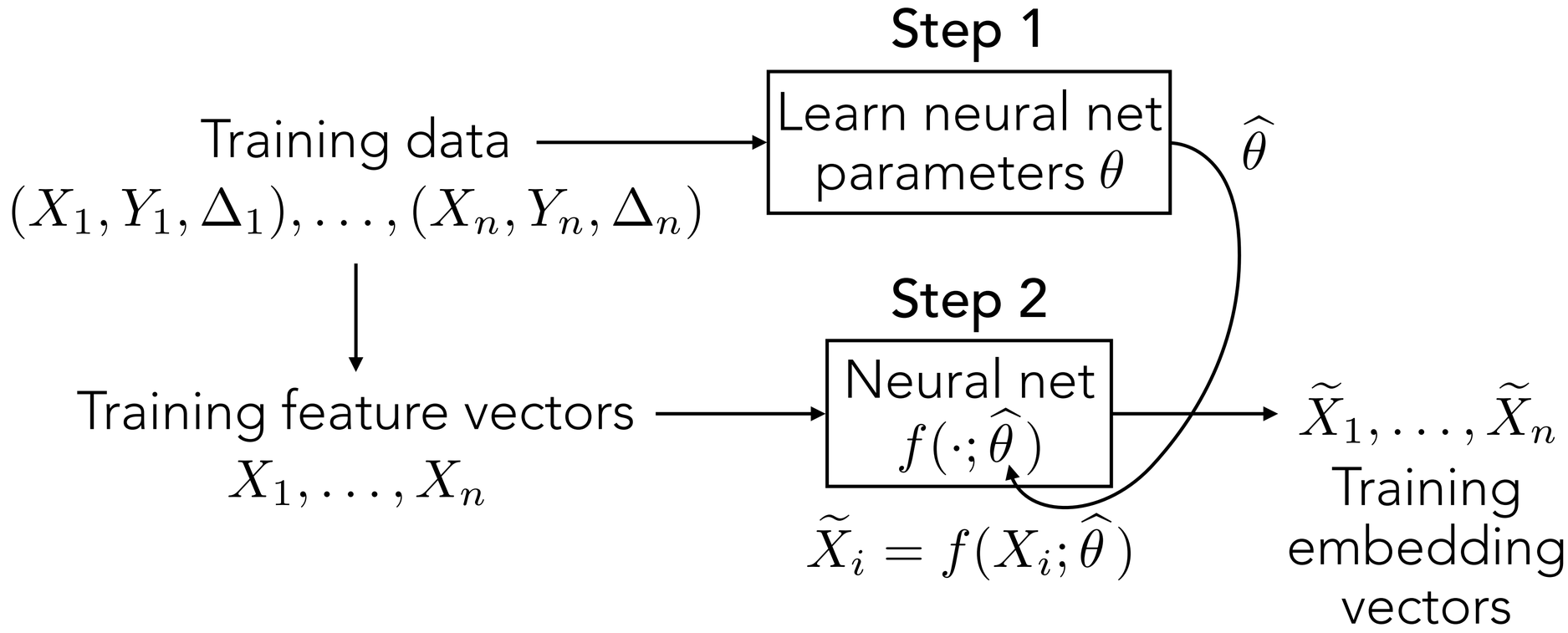
Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



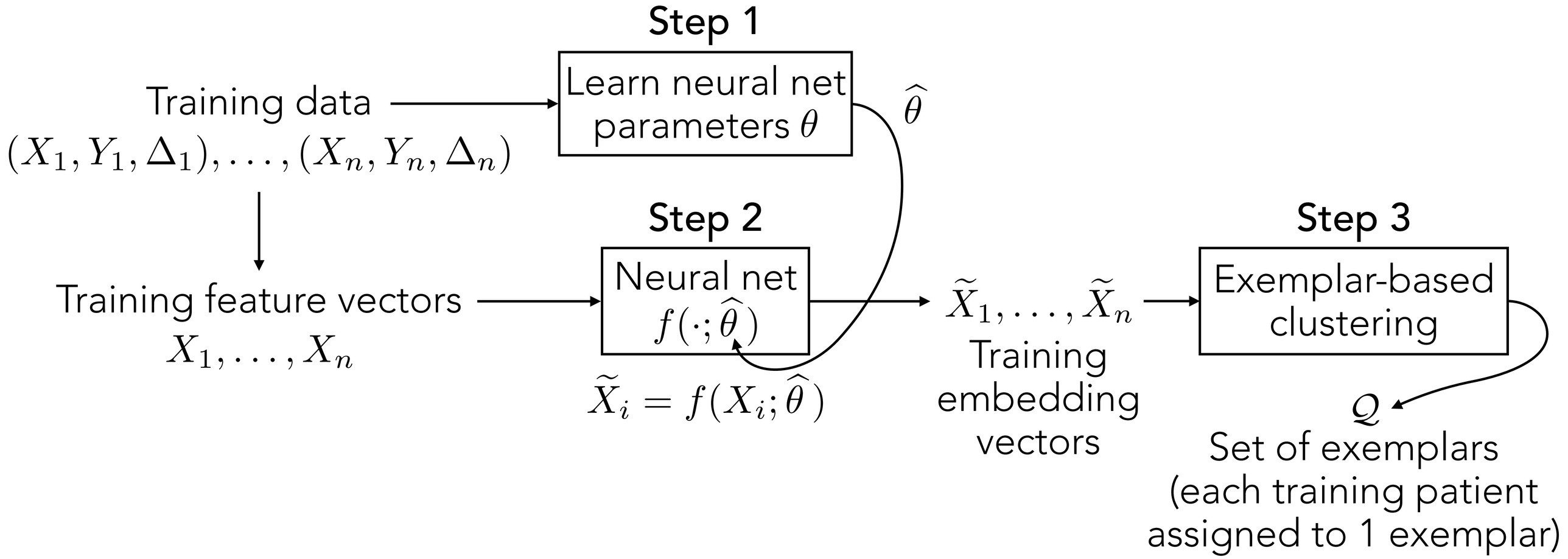
Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



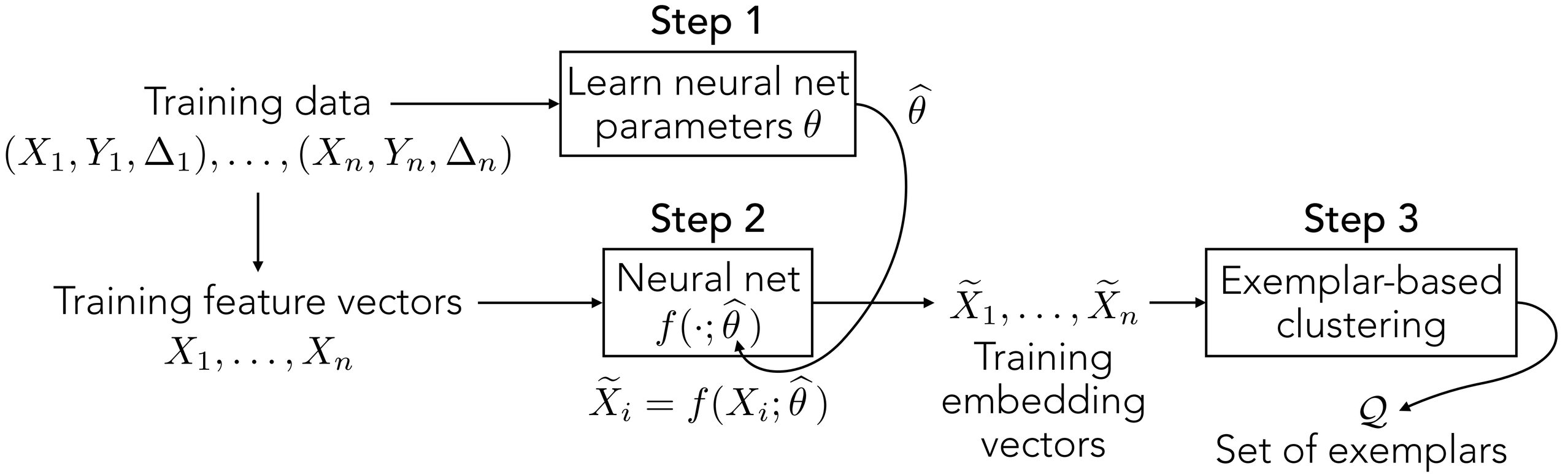
Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



Step 4

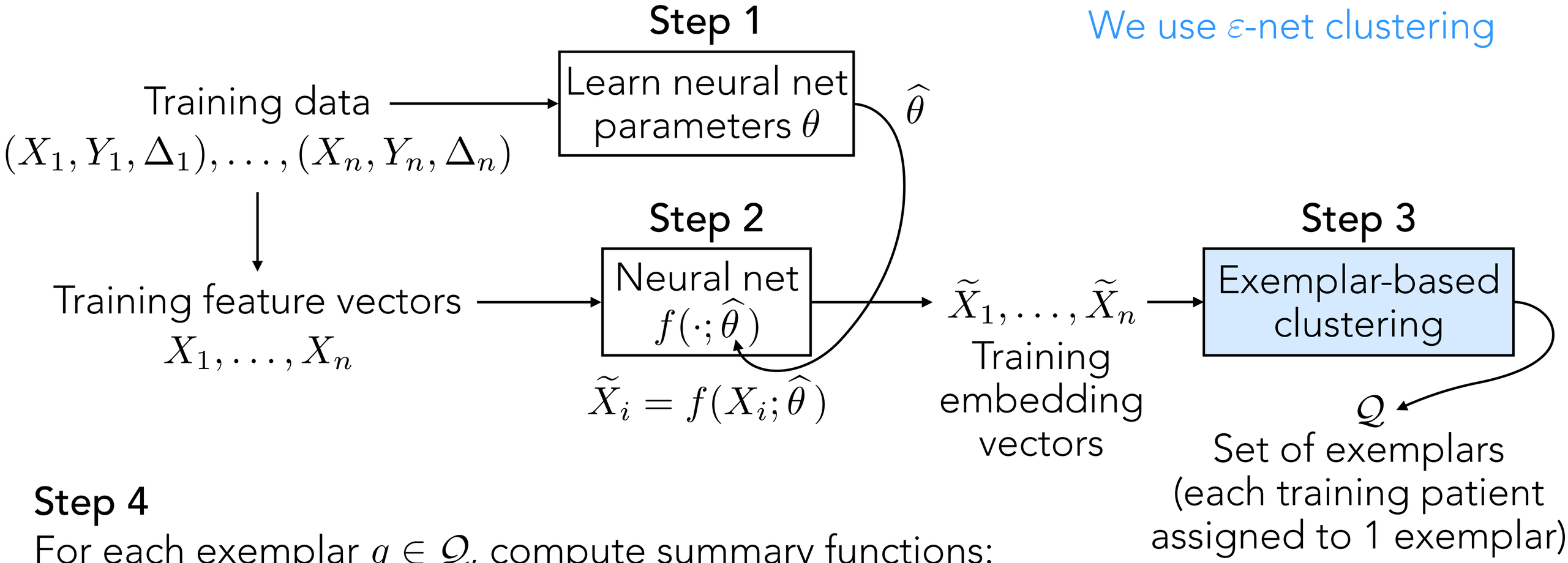
For each exemplar $q \in \mathcal{Q}$, compute summary functions:

$d_{\delta,\ell}^{\text{cluster}}(q)$ = # times event δ happened at time t_ℓ among patients in q's cluster

$n_{\ell}^{\text{cluster}}(q)$ = # at risk at time t_ℓ among patients in q's cluster

Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



We use ε-net clustering

Step 4

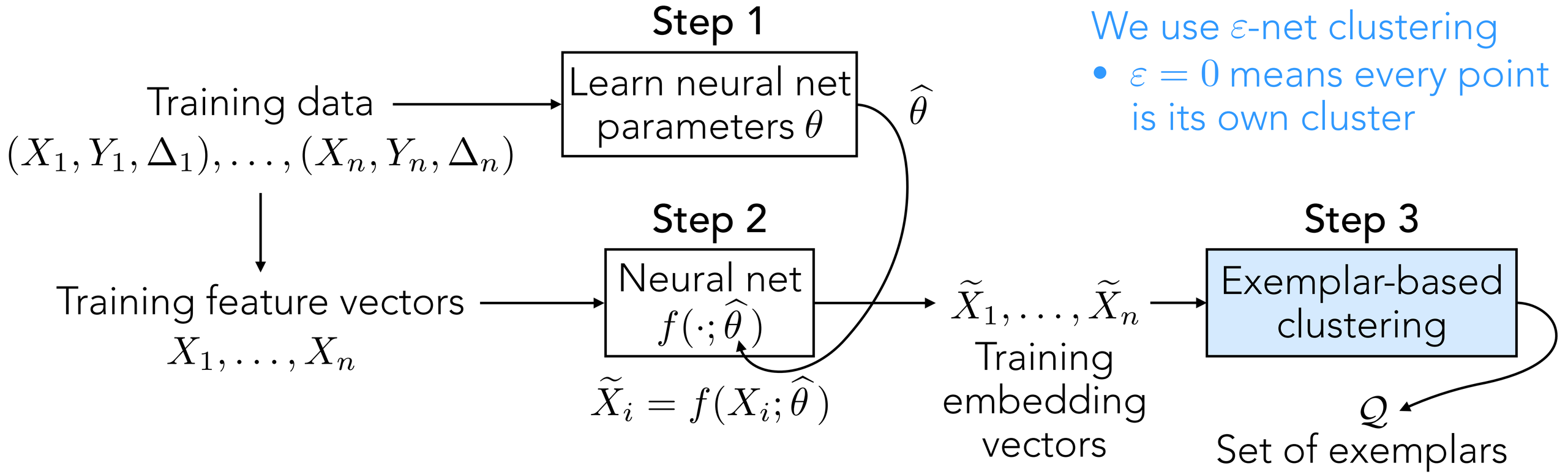
For each exemplar $q \in \mathcal{Q}$, compute summary functions:

$d_{\delta,\ell}^{\text{cluster}}(q)$ = # times event δ happened at time t_ℓ among patients in q's cluster

$n_\ell^{\text{cluster}}(q)$ = # at risk at time t_ℓ among patients in q's cluster

Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



We use ε-net clustering

- $\varepsilon = 0$ means every point is its own cluster

Step 4

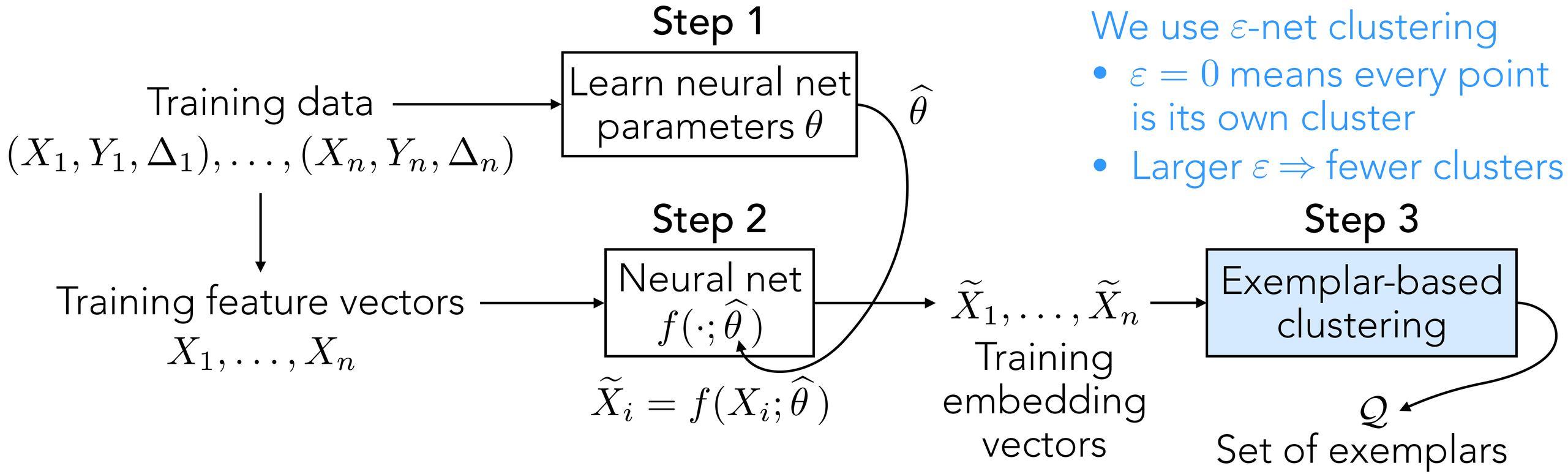
For each exemplar $q \in \mathcal{Q}$, compute summary functions:

$d_{\delta,\ell}^{\text{cluster}}(q)$ = # times event δ happened at time t_ℓ among patients in q's cluster

$n_\ell^{\text{cluster}}(q)$ = # at risk at time t_ℓ among patients in q's cluster

Recall: kernel function is $K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$

DKAJ Training



We use ε-net clustering

- $\varepsilon = 0$ means every point is its own cluster
- Larger $\varepsilon \Rightarrow$ fewer clusters

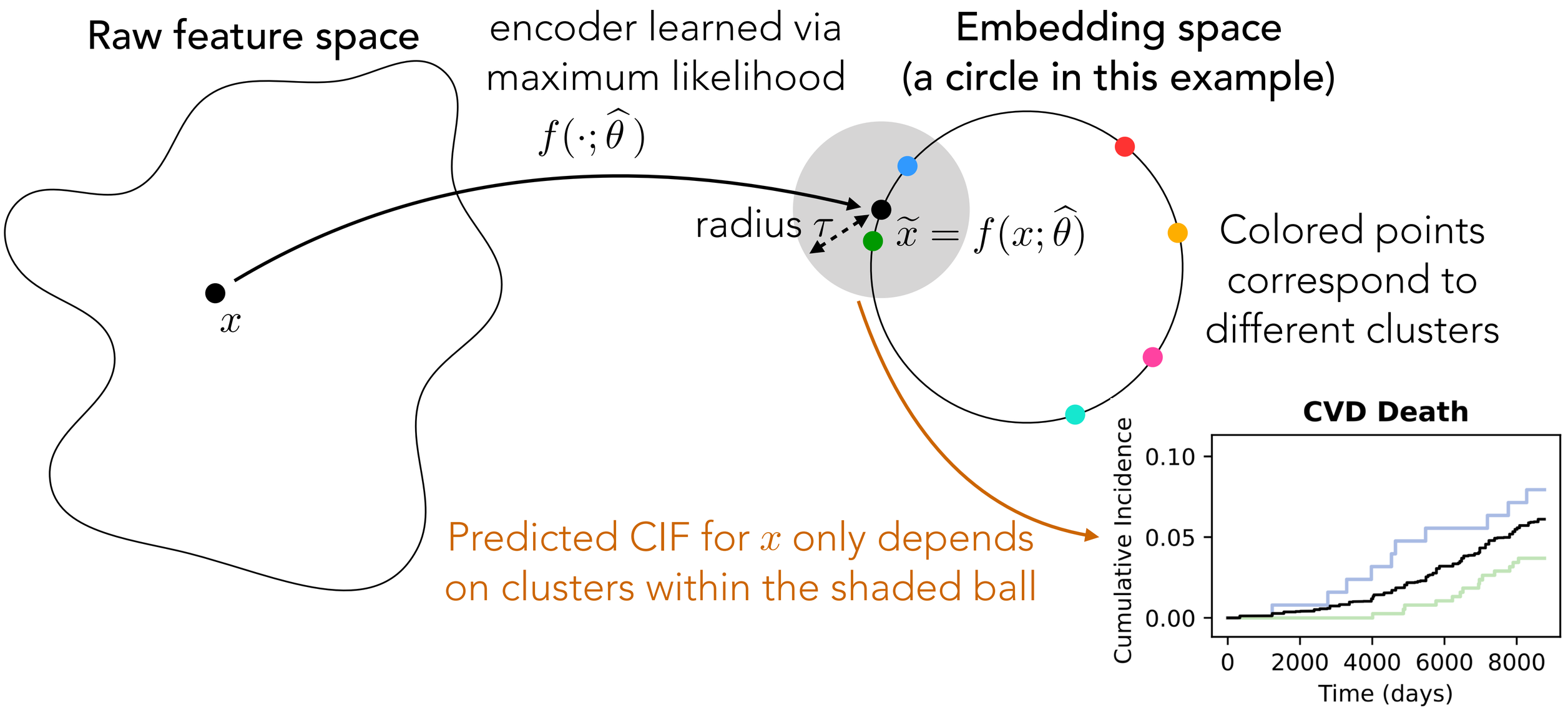
Step 4

For each exemplar $q \in \mathcal{Q}$, compute summary functions:

$d_{\delta,\ell}^{\text{cluster}}(q) =$ # times event δ happened at time t_ℓ among patients in q's cluster

$n_\ell^{\text{cluster}}(q) =$ # at risk at time t_ℓ among patients in q's cluster

Reminder: DKAJ High-Level Idea



DKAJ Prediction

DKAJ Prediction

$$\widehat{F}_{\delta}^{\mathrm{DKAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}(t_{\ell-1}|x) \frac{d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)} \qquad \widehat{S}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)}\right)$$

DKAJ Prediction

$$\widehat{F}_{\delta}^{\mathrm{DKAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}(t_{\ell-1}|x) \frac{d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)} \qquad \widehat{S}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)}\right)$$

$$d_{\delta,\ell}^{\mathrm{DKAJ}}(x) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) d_{\delta,\ell}^{\mathrm{cluster}}(q)$$

DKAJ Prediction

$$\widehat{F}_{\delta}^{\mathrm{DKAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}(t_{\ell-1}|x) \frac{d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)} \qquad \widehat{S}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)}\right)$$

$$d_{\delta,\ell}^{\mathrm{DKAJ}}(x) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) d_{\delta,\ell}^{\mathrm{cluster}}(q)$$

DKAJ Prediction

$$\widehat{F}_{\delta}^{\mathrm{DKAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \widehat{S}(t_{\ell-1}|x) \frac{d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)} \qquad \widehat{S}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \le t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)}\right)$$

$$d_{\delta,\ell}^{\mathrm{DKAJ}}(x) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) d_{\delta,\ell}^{\mathrm{cluster}}(q)$$

$$n_\ell^{\mathrm{DKAJ}}(q) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) n_\ell^{\mathrm{cluster}}(q)$$

DKAJ Prediction

$$\widehat{F}_{\delta}^{\mathrm{DKAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}(t_{\ell-1}|x) \frac{d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)} \qquad \widehat{S}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}^{\mathrm{DKAJ}}(x)}{n_\ell^{\mathrm{DKAJ}}(x)}\right)$$

$$d_{\delta,\ell}^{\mathrm{DKAJ}}(x) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) d_{\delta,\ell}^{\mathrm{cluster}}(q)$$

$$n_\ell^{\mathrm{DKAJ}}(q) \triangleq \sum_{\substack{\text{exemplar } q \\ \text{within distance } \tau \text{ of } x \\ \text{(in embedding space)}}} K(x, X_q) n_\ell^{\mathrm{cluster}}(q)$$

Takeaway: CIFs for x are predicted using only close-enough exemplars (in embedding space)

Numerical Experiments

Benchmark

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters
- Evaluation metrics: C^{td} [Antolini et al 2015], paper also has IBS

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters
- Evaluation metrics: C^{td} [Antolini et al 2015], paper also has IBS

Test set C^{td} index (mean±std dev across 10 experimental repeats) for 4 datasets (each with 2 event types): **bold** = best, blue = 2nd best

Dataset	Method	Critical Event 1	Critical Event 2
PBC [Fleming & Harrington 1991] N=1945, # features=15	Fine & Gray	0.8212±0.0119	0.8769±0.0203
	Cause-specific Cox	0.8292±0.0100	0.9091±0.0134
	DeepHit	0.8407±0.0131	0.9060±0.0130
	DSM	0.8319±0.0115	0.9039±0.0189
	NeuralFG	0.8363±0.0150	0.9120±0.0149
	SurvivalBoost	**0.8719±0.0107**	**0.9360±0.0114**
	DKAJ (ours)	0.8351±0.0106	0.8782±0.0412
Framingham [Kannel & McGee 1979] N=4434, # features=18	Fine & Gray	0.7733±0.0106	0.7144±0.0182
	Cause-specific Cox	**0.7751±0.0107**	**0.7160±0.0174**
	DeepHit	0.7423±0.0198	0.6957±0.0195
	DSM	0.7664±0.0170	0.7095±0.0184
	NeuralFG	0.6833±0.0619	0.6978±0.0287
	SurvivalBoost	0.7645±0.0152	0.7031±0.0156
	DKAJ (ours)	0.7656±0.0147	0.7034±0.0185
SEER [https://seer.cancer.gov] N=24907, # features=22	Fine & Gray	0.8151±0.0021	0.8478±0.0099
	Cause-specific Cox	0.7630±0.0085	0.8413±0.0132
	DeepHit	0.8239±0.0028	0.8548±0.0120
	DSM	0.7807±0.0088	0.8422±0.0119
	NeuralFG	0.7743±0.0090	0.8362±0.0138
	SurvivalBoost	**0.8418±0.0038**	**0.8583±0.0081**
	DKAJ (ours)	0.8247±0.0033	0.8549±0.0069
Synthetic [Lee et al 2018] N=30000, # features=12	Fine & Gray	0.5823±0.0051	0.5917±0.0071
	Cause-specific Cox	0.5808±0.0052	0.5903±0.0072
	DeepHit	0.7401±0.0067	0.7437±0.0043
	DSM	0.7280±0.0055	0.7320±0.0042
	NeuralFG	**0.7481±0.0073**	**0.7513±0.0045**
	SurvivalBoost	0.7160±0.0083	0.7190±0.0039
	DKAJ (ours)	0.7399±0.0062	0.7446±0.0048

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters
- Evaluation metrics: C^{td} [Antolini et al 2015], paper also has IBS

Main takeaways:

Test set C^{td} index (mean±std dev across 10 experimental repeats) for 4 datasets (each with 2 event types): **bold** = best, blue = 2nd best

Dataset	Method	Critical Event 1	Critical Event 2
PBC [Fleming & Harrington 1991] N=1945, # features=15	Fine & Gray	0.8212±0.0119	0.8769±0.0203
	Cause-specific Cox	0.8292±0.0100	0.9091±0.0134
	DeepHit	0.8407±0.0131	0.9060±0.0130
	DSM	0.8319±0.0115	0.9039±0.0189
	NeuralFG	0.8363±0.0150	0.9120±0.0149
	SurvivalBoost	**0.8719±0.0107**	**0.9360±0.0114**
	DKAJ (ours)	0.8351±0.0106	0.8782±0.0412
Framingham [Kannel & McGee 1979] N=4434, # features=18	Fine & Gray	0.7733±0.0106	0.7144±0.0182
	Cause-specific Cox	**0.7751±0.0107**	**0.7160±0.0174**
	DeepHit	0.7423±0.0198	0.6957±0.0195
	DSM	0.7664±0.0170	0.7095±0.0184
	NeuralFG	0.6833±0.0619	0.6978±0.0287
	SurvivalBoost	0.7645±0.0152	0.7031±0.0156
	DKAJ (ours)	0.7656±0.0147	0.7034±0.0185
SEER [https://seer.cancer.gov] N=24907, # features=22	Fine & Gray	0.8151±0.0021	0.8478±0.0099
	Cause-specific Cox	0.7630±0.0085	0.8413±0.0132
	DeepHit	0.8239±0.0028	0.8548±0.0120
	DSM	0.7807±0.0088	0.8422±0.0119
	NeuralFG	0.7743±0.0090	0.8362±0.0138
	SurvivalBoost	**0.8418±0.0038**	**0.8583±0.0081**
	DKAJ (ours)	0.8247±0.0033	0.8549±0.0069
Synthetic [Lee et al 2018] N=30000, # features=12	Fine & Gray	0.5823±0.0051	0.5917±0.0071
	Cause-specific Cox	0.5808±0.0052	0.5903±0.0072
	DeepHit	0.7401±0.0067	0.7437±0.0043
	DSM	0.7280±0.0055	0.7320±0.0042
	NeuralFG	**0.7481±0.0073**	**0.7513±0.0045**
	SurvivalBoost	0.7160±0.0083	0.7190±0.0039
	DKAJ (ours)	0.7399±0.0062	0.7446±0.0048

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters
- Evaluation metrics: C^{td} [Antolini et al 2015], paper also has IBS

Main takeaways:

- No single model is best across all datasets

Test set C^{td} index (mean±std dev across 10 experimental repeats) for 4 datasets (each with 2 event types): **bold** = best, blue = 2nd best

Dataset	Method	Critical Event 1	Critical Event 2
PBC [Fleming & Harrington 1991] N=1945, # features=15	Fine & Gray	0.8212±0.0119	0.8769±0.0203
	Cause-specific Cox	0.8292±0.0100	0.9091±0.0134
	DeepHit	0.8407±0.0131	0.9060±0.0130
	DSM	0.8319±0.0115	0.9039±0.0189
	NeuralFG	0.8363±0.0150	0.9120±0.0149
	SurvivalBoost	**0.8719±0.0107**	**0.9360±0.0114**
	DKAJ (ours)	0.8351±0.0106	0.8782±0.0412
Framingham [Kannel & McGee 1979] N=4434, # features=18	Fine & Gray	0.7733±0.0106	0.7144±0.0182
	Cause-specific Cox	**0.7751±0.0107**	**0.7160±0.0174**
	DeepHit	0.7423±0.0198	0.6957±0.0195
	DSM	0.7664±0.0170	0.7095±0.0184
	NeuralFG	0.6833±0.0619	0.6978±0.0287
	SurvivalBoost	0.7645±0.0152	0.7031±0.0156
	DKAJ (ours)	0.7656±0.0147	0.7034±0.0185
SEER [https://seer.cancer.gov] N=24907, # features=22	Fine & Gray	0.8151±0.0021	0.8478±0.0099
	Cause-specific Cox	0.7630±0.0085	0.8413±0.0132
	DeepHit	0.8239±0.0028	0.8548±0.0120
	DSM	0.7807±0.0088	0.8422±0.0119
	NeuralFG	0.7743±0.0090	0.8362±0.0138
	SurvivalBoost	**0.8418±0.0038**	**0.8583±0.0081**
	DKAJ (ours)	0.8247±0.0033	0.8549±0.0069
Synthetic [Lee et al 2018] N=30000, # features=12	Fine & Gray	0.5823±0.0051	0.5917±0.0071
	Cause-specific Cox	0.5808±0.0052	0.5903±0.0072
	DeepHit	0.7401±0.0067	0.7437±0.0043
	DSM	0.7280±0.0055	0.7320±0.0042
	NeuralFG	**0.7481±0.0073**	**0.7513±0.0045**
	SurvivalBoost	0.7160±0.0083	0.7190±0.0039
	DKAJ (ours)	0.7399±0.0062	0.7446±0.0048

Benchmark

- 4 standard datasets each with 56/14/30 train/val/test split
  - Val. set is for tuning hyperparameters
- Evaluation metrics: C^{td} [Antolini et al 2015], paper also has IBS

Main takeaways:

- No single model is best across all datasets
- DKAJ is competitive with various baselines

Test set C^{td} index (mean±std dev across 10 experimental repeats) for 4 datasets (each with 2 event types): **bold** = best, blue = 2nd best

Dataset	Method	Critical Event 1	Critical Event 2
PBC [Fleming & Harrington 1991] N=1945, # features=15	Fine & Gray	0.8212±0.0119	0.8769±0.0203
	Cause-specific Cox	0.8292±0.0100	0.9091±0.0134
	DeepHit	0.8407±0.0131	0.9060±0.0130
	DSM	0.8319±0.0115	0.9039±0.0189
	NeuralFG	0.8363±0.0150	0.9120±0.0149
	SurvivalBoost	**0.8719±0.0107**	**0.9360±0.0114**
	DKAJ (ours)	0.8351±0.0106	0.8782±0.0412
Framingham [Kannel & McGee 1979] N=4434, # features=18	Fine & Gray	0.7733±0.0106	0.7144±0.0182
	Cause-specific Cox	**0.7751±0.0107**	**0.7160±0.0174**
	DeepHit	0.7423±0.0198	0.6957±0.0195
	DSM	0.7664±0.0170	0.7095±0.0184
	NeuralFG	0.6833±0.0619	0.6978±0.0287
	SurvivalBoost	0.7645±0.0152	0.7031±0.0156
	DKAJ (ours)	0.7656±0.0147	0.7034±0.0185
SEER [https://seer.cancer.gov] N=24907, # features=22	Fine & Gray	0.8151±0.0021	0.8478±0.0099
	Cause-specific Cox	0.7630±0.0085	0.8413±0.0132
	DeepHit	0.8239±0.0028	0.8548±0.0120
	DSM	0.7807±0.0088	0.8422±0.0119
	NeuralFG	0.7743±0.0090	0.8362±0.0138
	SurvivalBoost	**0.8418±0.0038**	**0.8583±0.0081**
	DKAJ (ours)	0.8247±0.0033	0.8549±0.0069
Synthetic [Lee et al 2018] N=30000, # features=12	Fine & Gray	0.5823±0.0051	0.5917±0.0071
	Cause-specific Cox	0.5808±0.0052	0.5903±0.0072
	DeepHit	0.7401±0.0067	0.7437±0.0043
	DSM	0.7280±0.0055	0.7320±0.0042
	NeuralFG	**0.7481±0.0073**	**0.7513±0.0045**
	SurvivalBoost	0.7160±0.0083	0.7190±0.0039
	DKAJ (ours)	0.7399±0.0062	0.7446±0.0048

Illustration of DKAJ Model Interpretation: Framingham

Illustration of DKAJ Model Interpretation: Framingham

2 critical events:
CVD (cardiovascular disease) death,
Non-CVD death

Illustration of DKAJ Model Interpretation: Framingham

2 critical events:
CVD (cardiovascular disease) death,
Non-CVD death

Choose which clusters to focus on

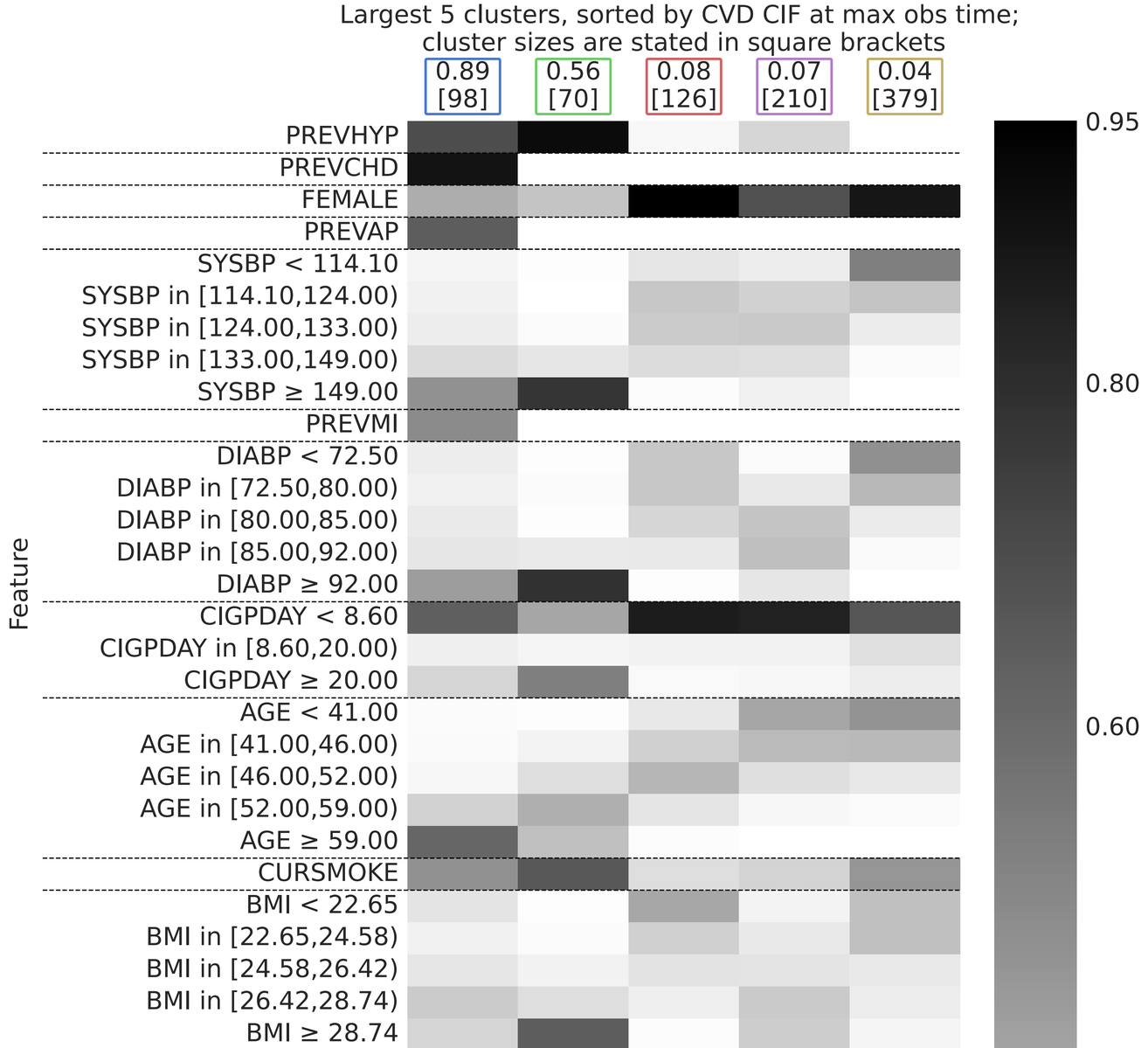
Illustration of DKAJ Model Interpretation: Framingham

2 critical events:
CVD (cardiovascular disease) death,
Non-CVD death

Choose which clusters to focus on

Concrete example:
focus on the 5 largest ones
(clusters with the most training
patients assigned to them)

Illustration of DKAJ Model Interpretation: Framingham



2 critical events:
CVD (cardiovascular disease) death,
Non-CVD death

Choose which clusters to focus on

Concrete example:
focus on the 5 largest ones
(clusters with the most training patients assigned to them)

Illustration of DKAJ Model Interpretation: Framingham

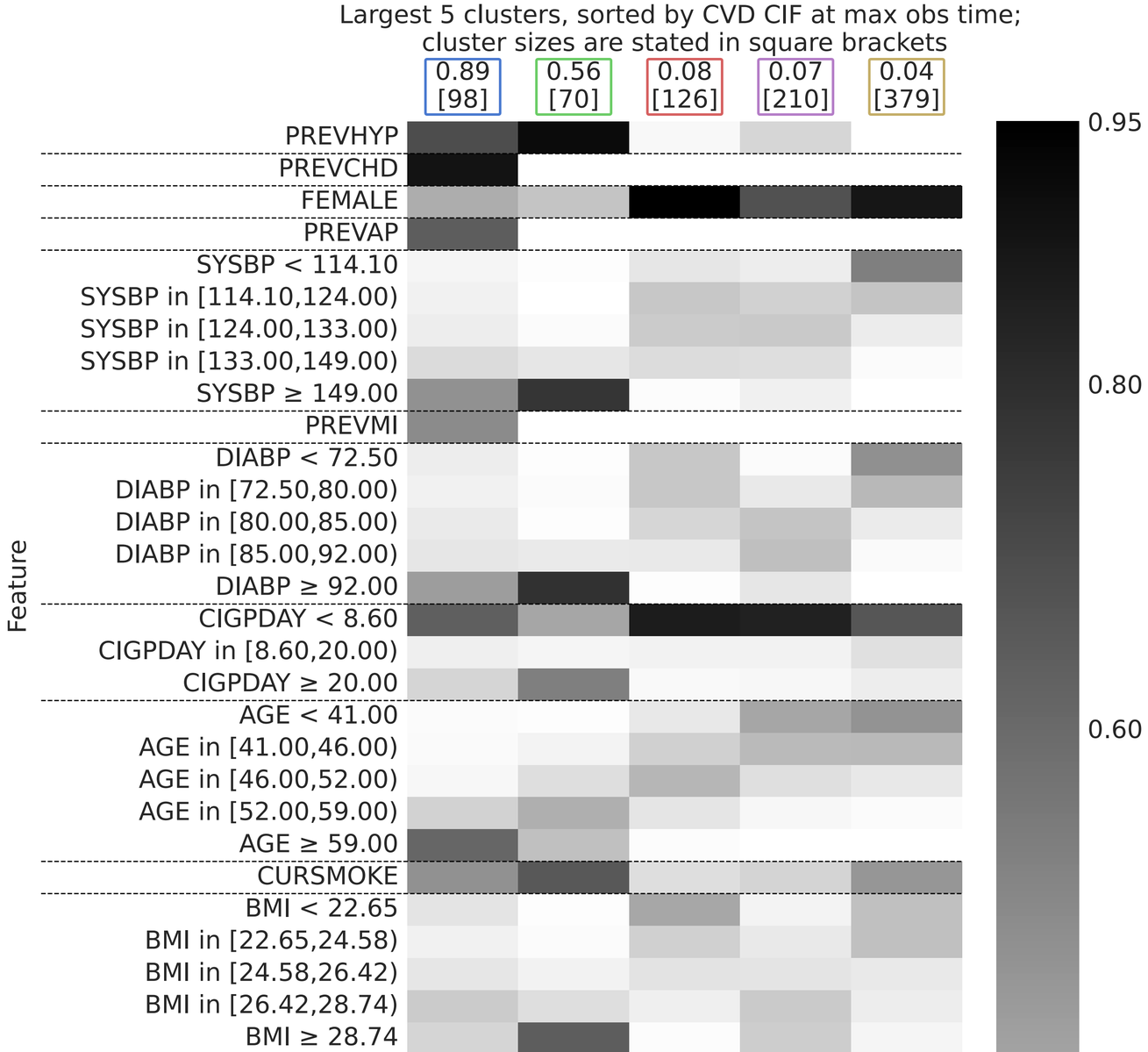
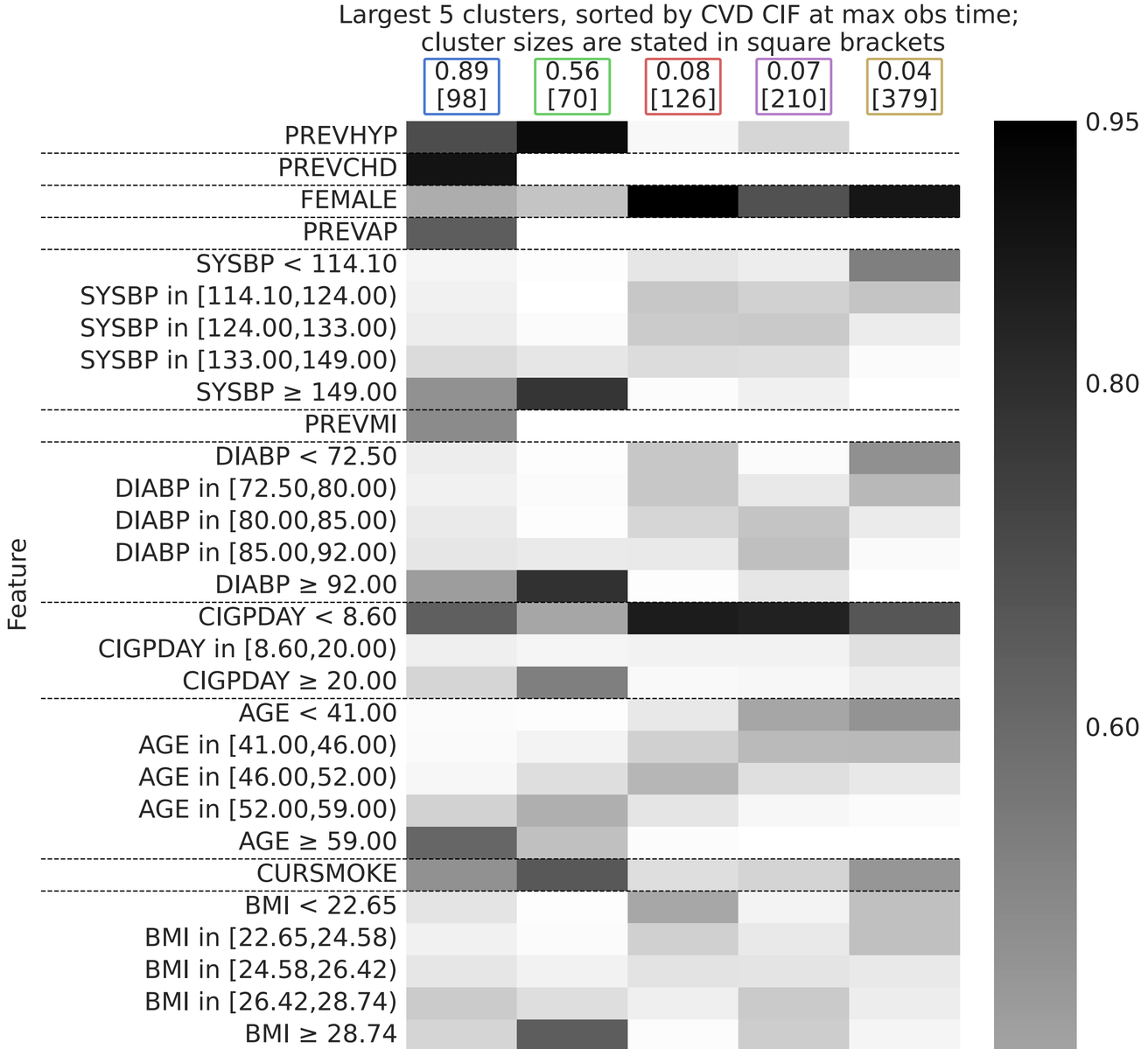
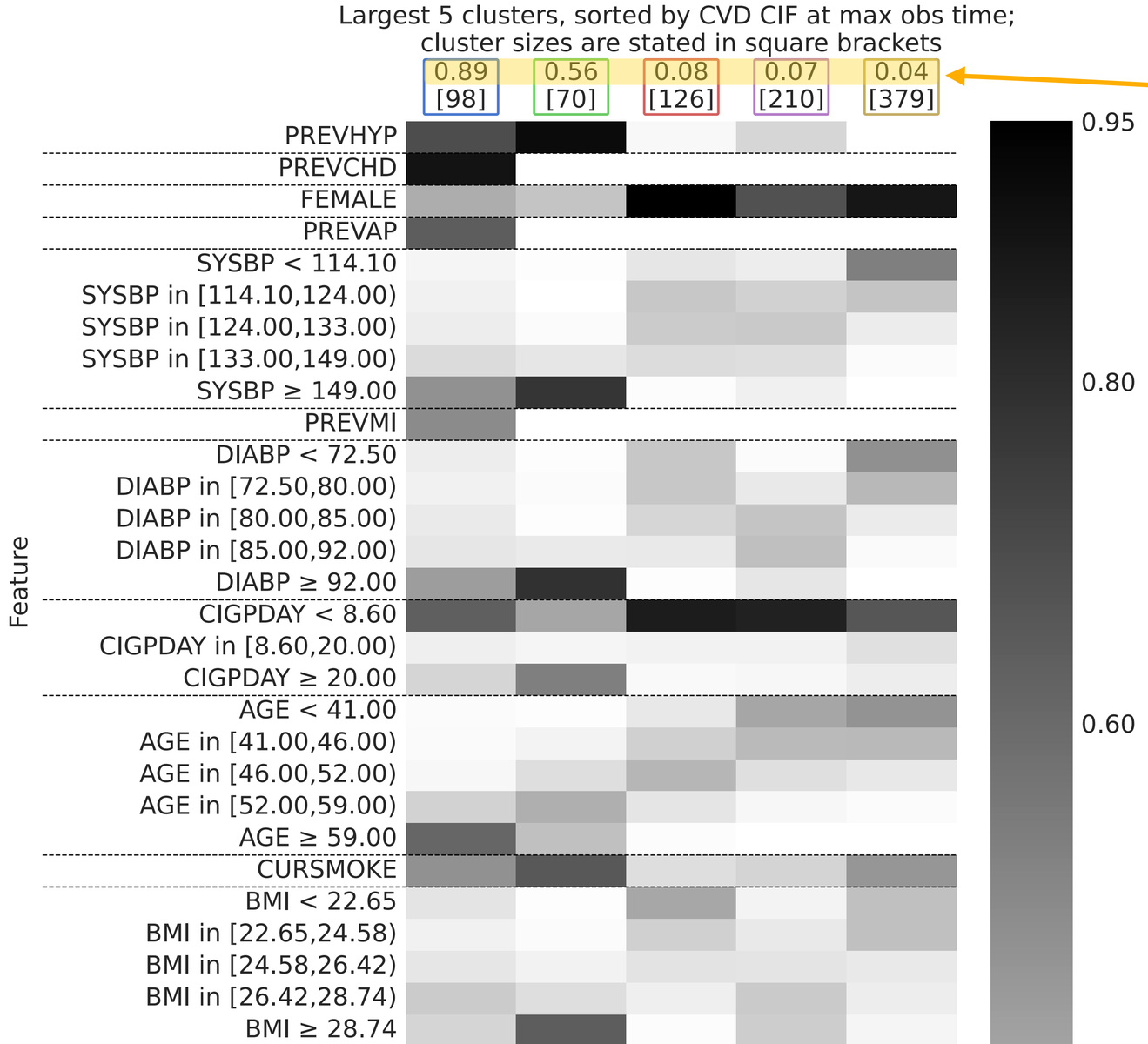


Illustration of DKAJ Model Interpretation: Framingham



Each column: a different cluster
columns sorted by "**risk of CVD death**"

Illustration of DKAJ Model Interpretation: Framingham

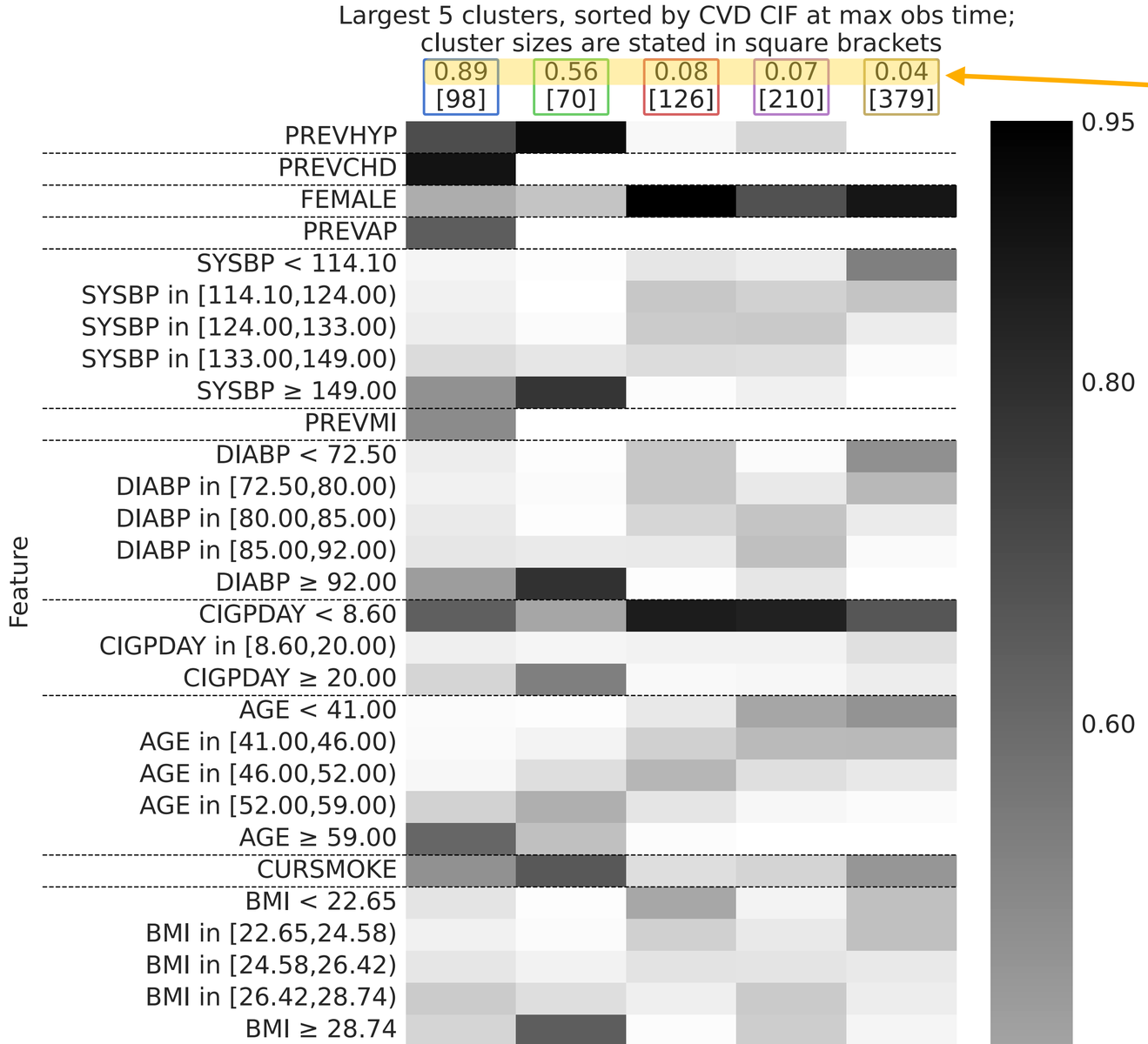


Each column: a different cluster

columns sorted by **"risk of CVD death"**

technically: CIF of CVD death evaluated at 24 years (max obs. time)

Illustration of DKAJ Model Interpretation: Framingham



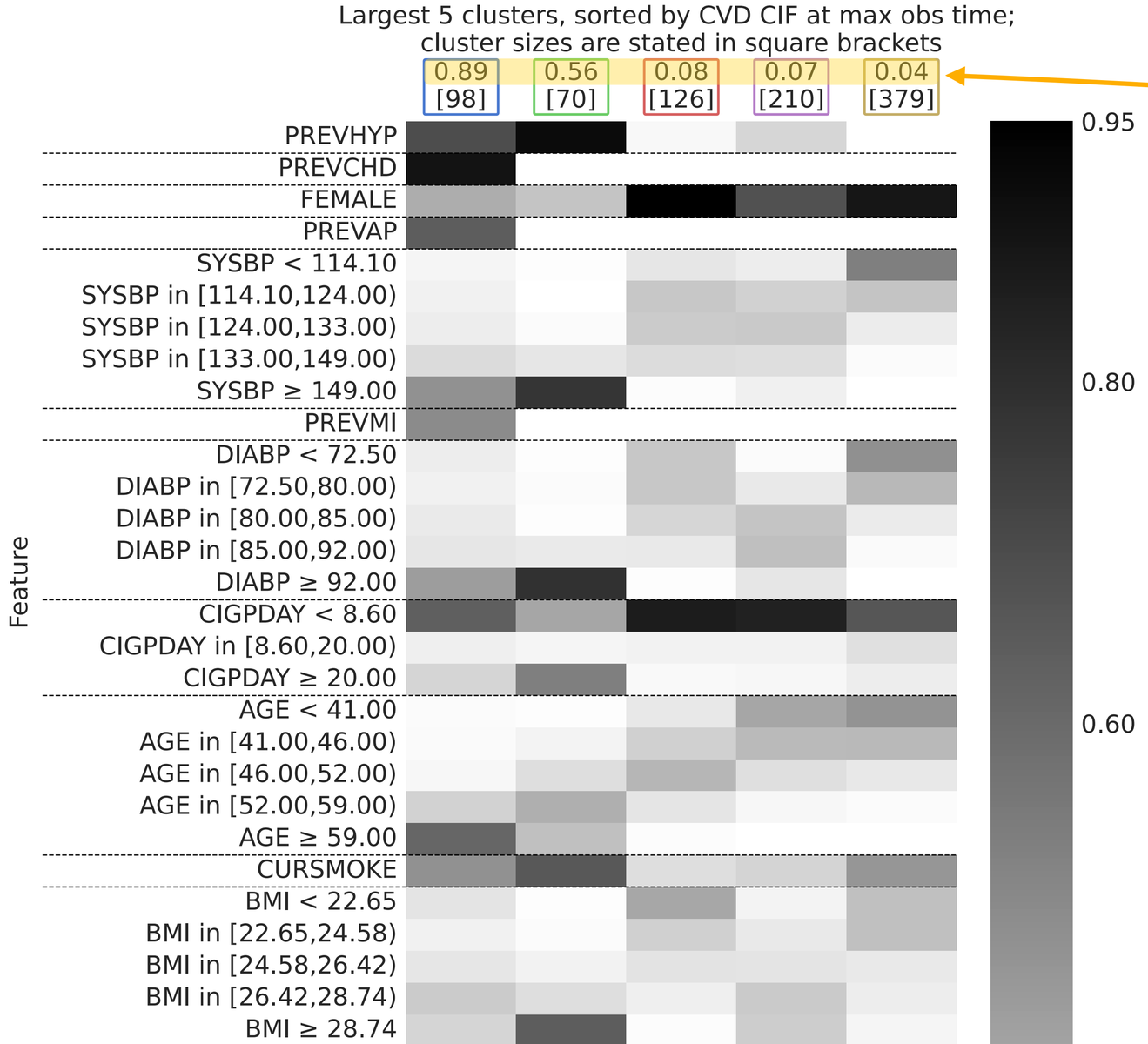
Each column: a different cluster

columns sorted by **"risk of CVD death"**

technically: CIF of CVD death evaluated at 24 years (max obs. time)

Rows: raw features

Illustration of DKAJ Model Interpretation: Framingham



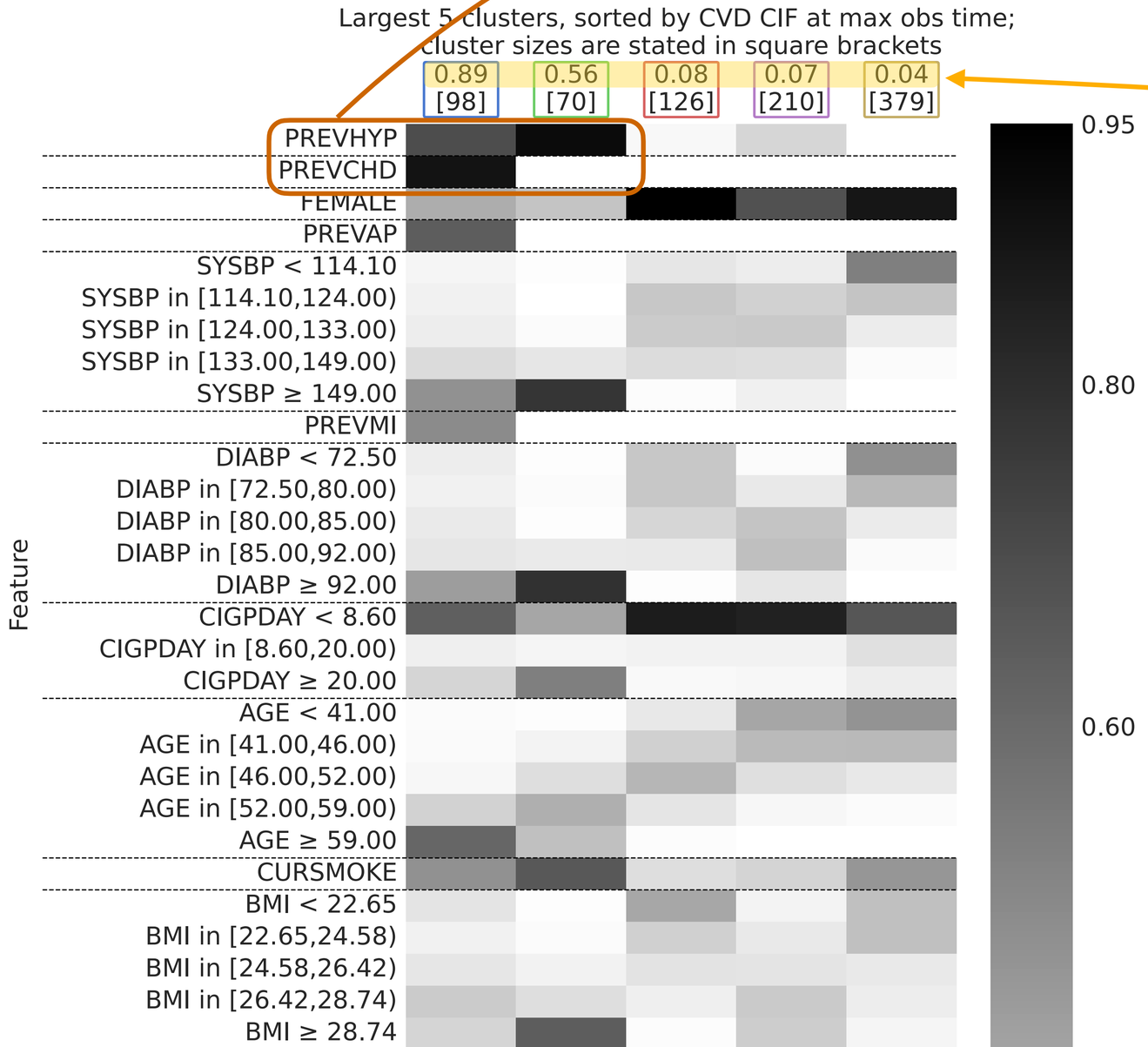
Each column: a different cluster
columns sorted by **"risk of CVD death"**
technically: CIF of CVD death
evaluated at 24 years (max obs. time)

Rows: raw features

Intensity value: fraction of people in a cluster with a particular raw feature

hypertension, coronary heart disease

Illustration of DKAJ Model Interpretation: Framingham



Each column: a different cluster

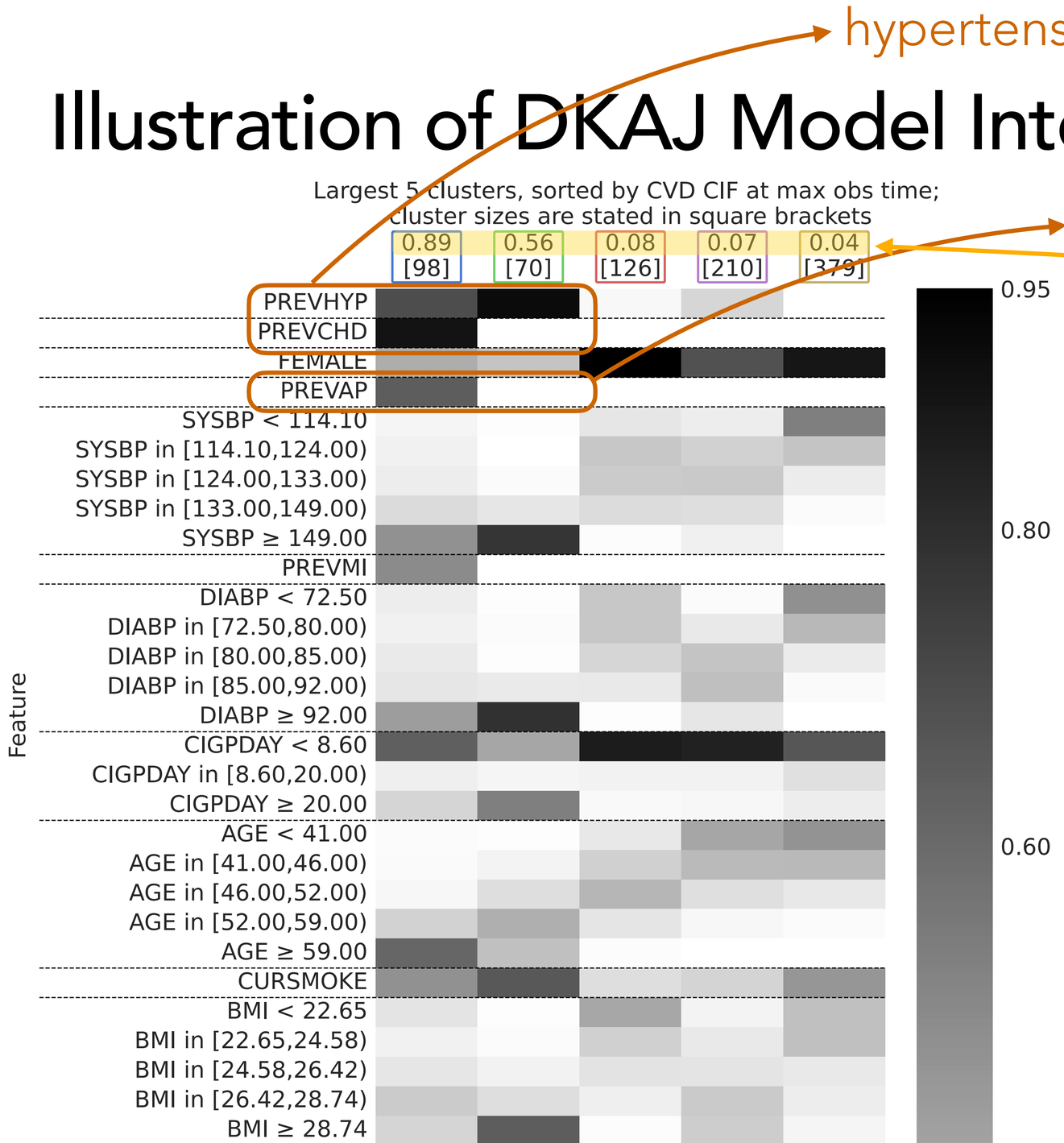
columns sorted by **"risk of CVD death"**

technically: CIF of CVD death evaluated at 24 years (max obs. time)

Rows: raw features

Intensity value: fraction of people in a cluster with a particular raw feature

Illustration of DKAJ Model Interpretation: Framingham



hypertension, coronary heart disease

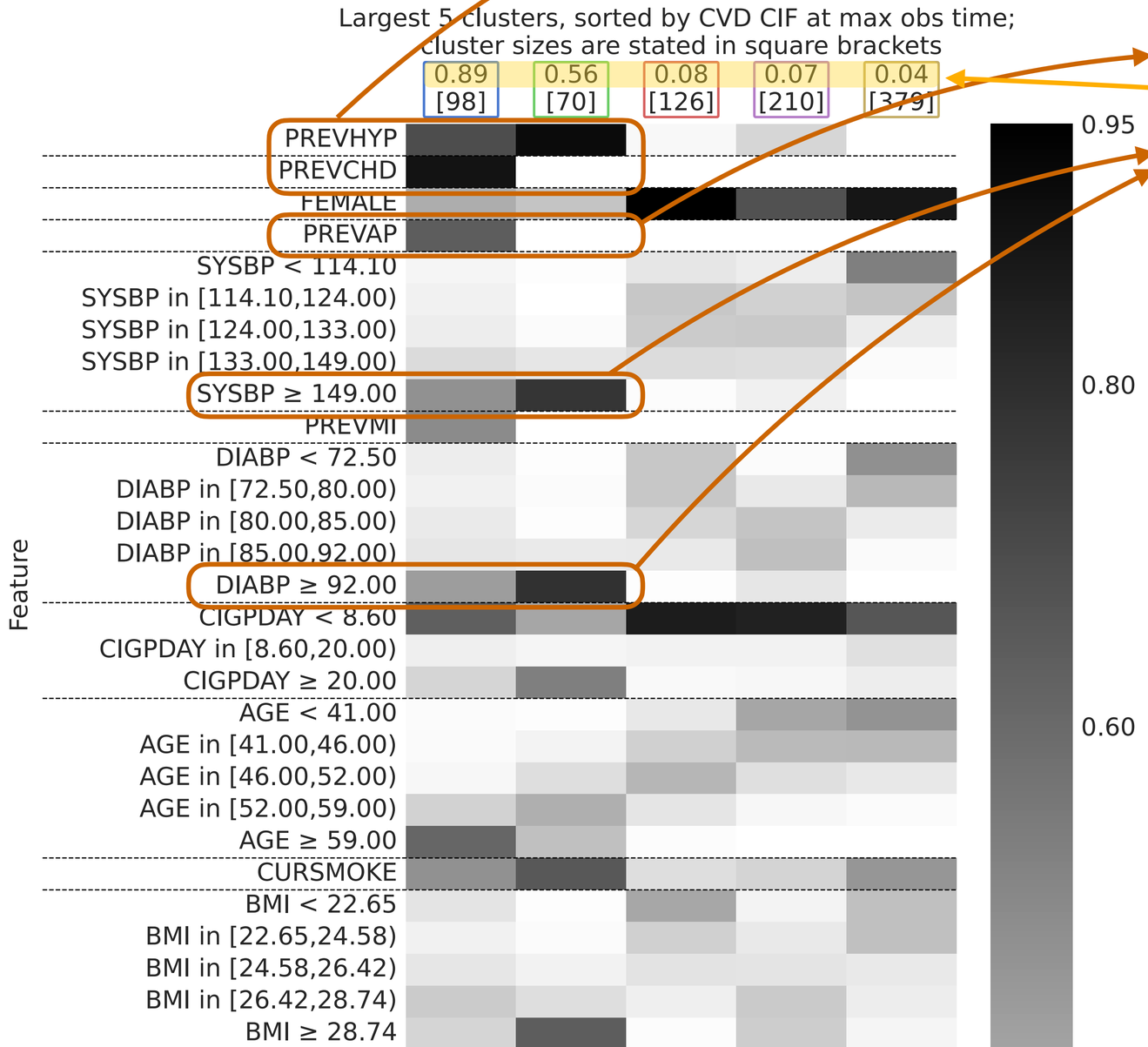
angina (chest pain/discomfort)

Each column: a different cluster
columns sorted by **"risk of CVD death"**
technically: CIF of CVD death
evaluated at 24 years (max obs. time)

Rows: raw features

Intensity value: fraction of people in a cluster with a particular raw feature

Illustration of DKAJ Model Interpretation: Framingham



hypertension, coronary heart disease

angina (chest pain/discomfort)

high blood pressure

Each column: a different cluster

columns sorted by **"risk of CVD death"**

technically: CIF of CVD death evaluated at 24 years (max obs. time)

Rows: raw features

Intensity value: fraction of people in a cluster with a particular raw feature

Illustration of DKAJ Model Interpretation: Framingham

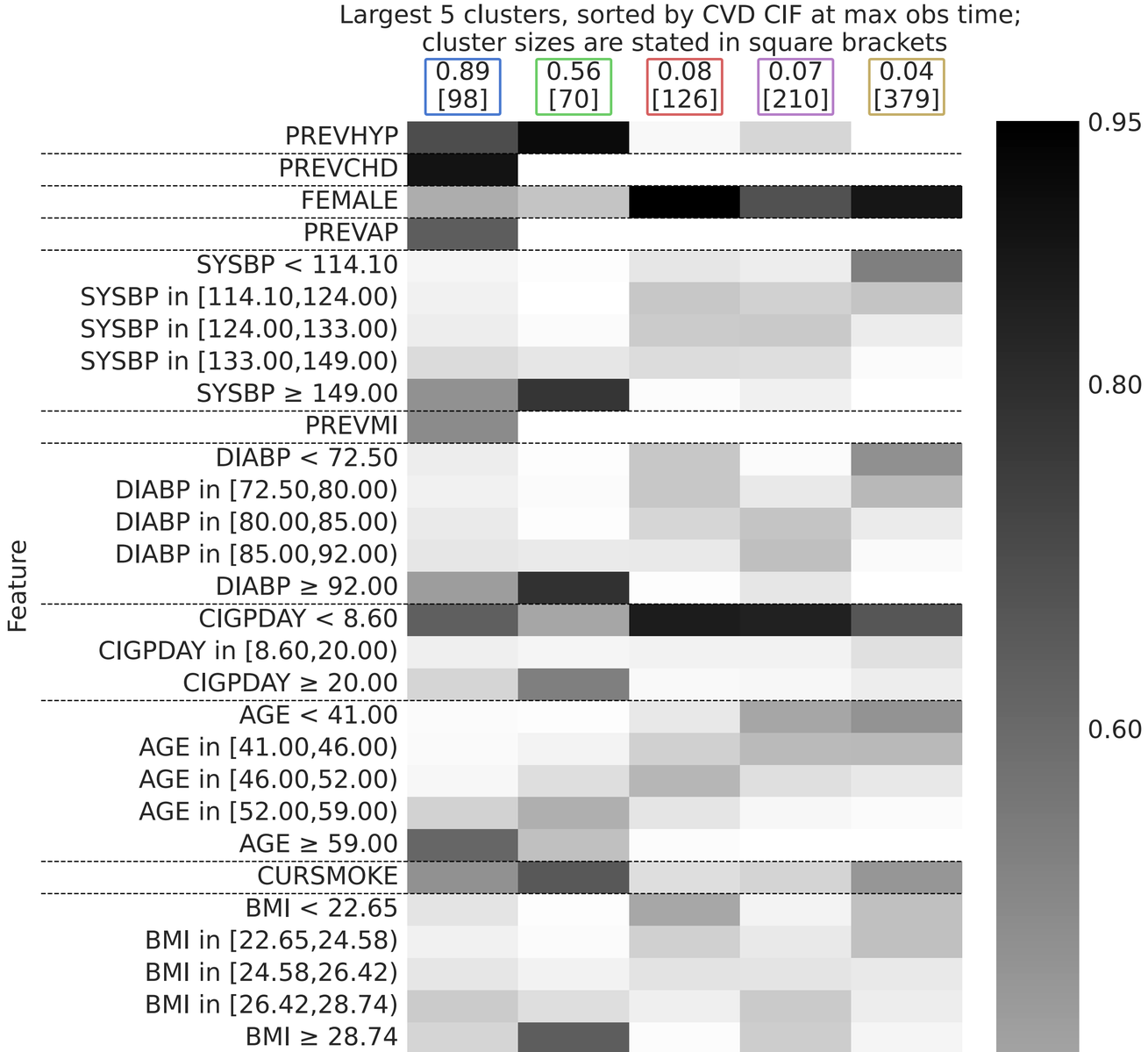
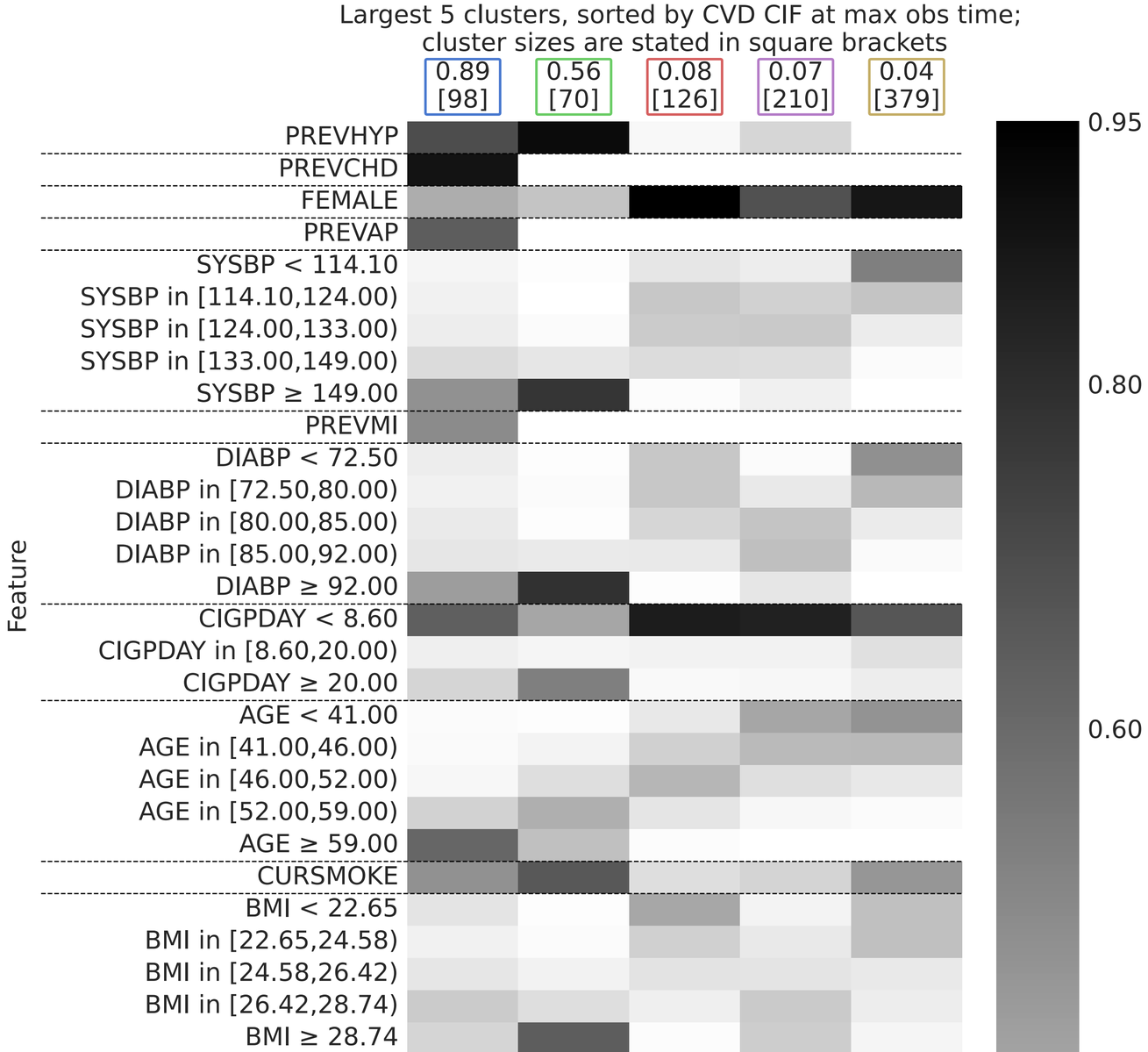
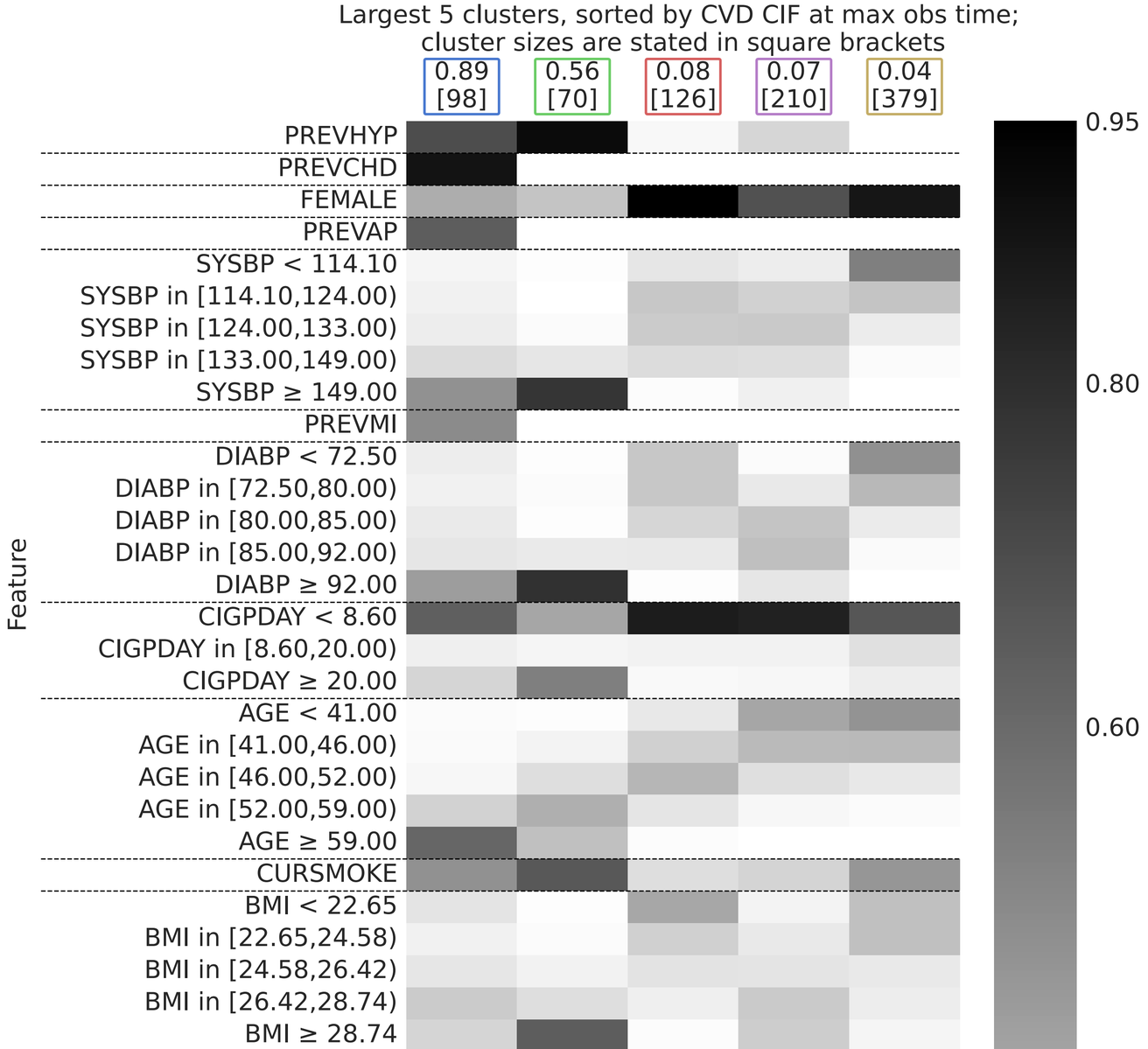


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

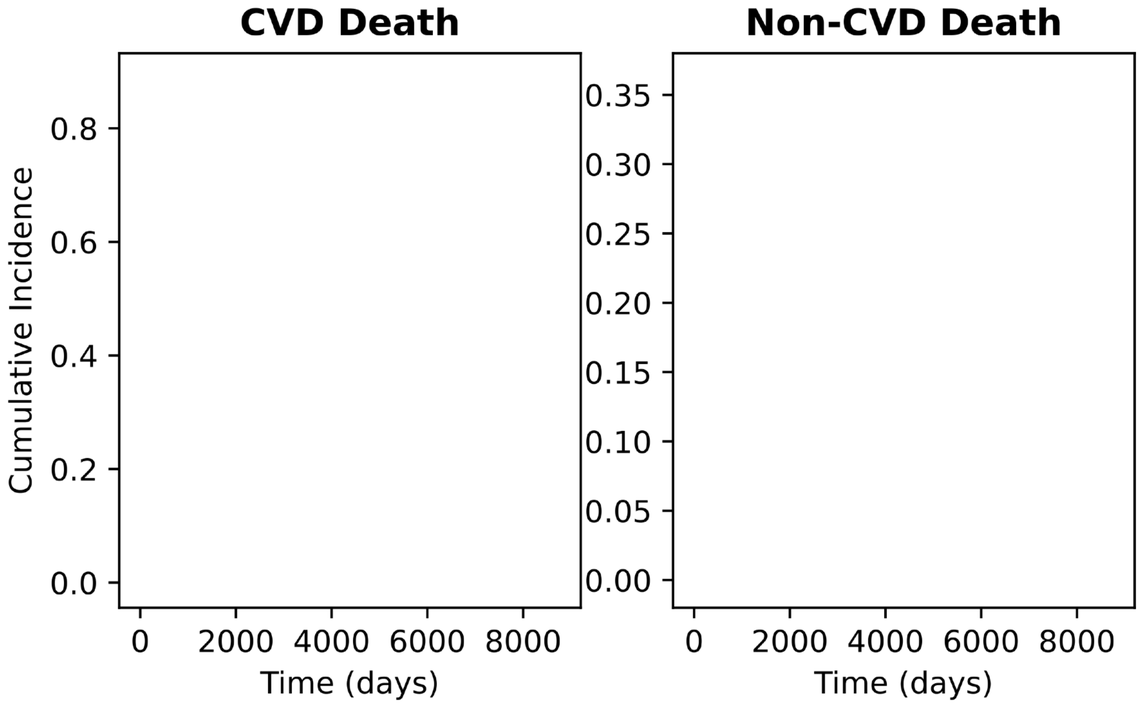
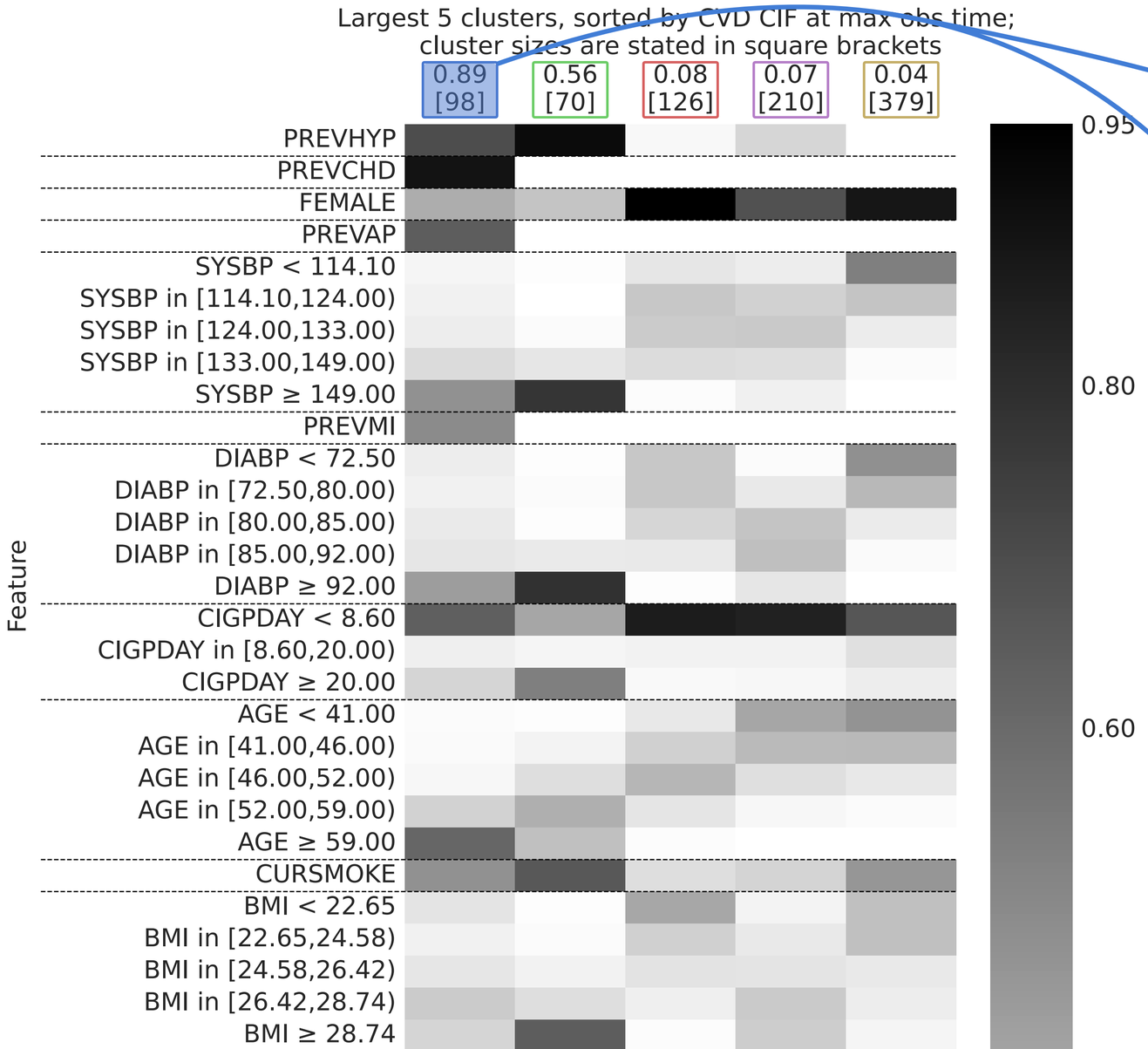


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

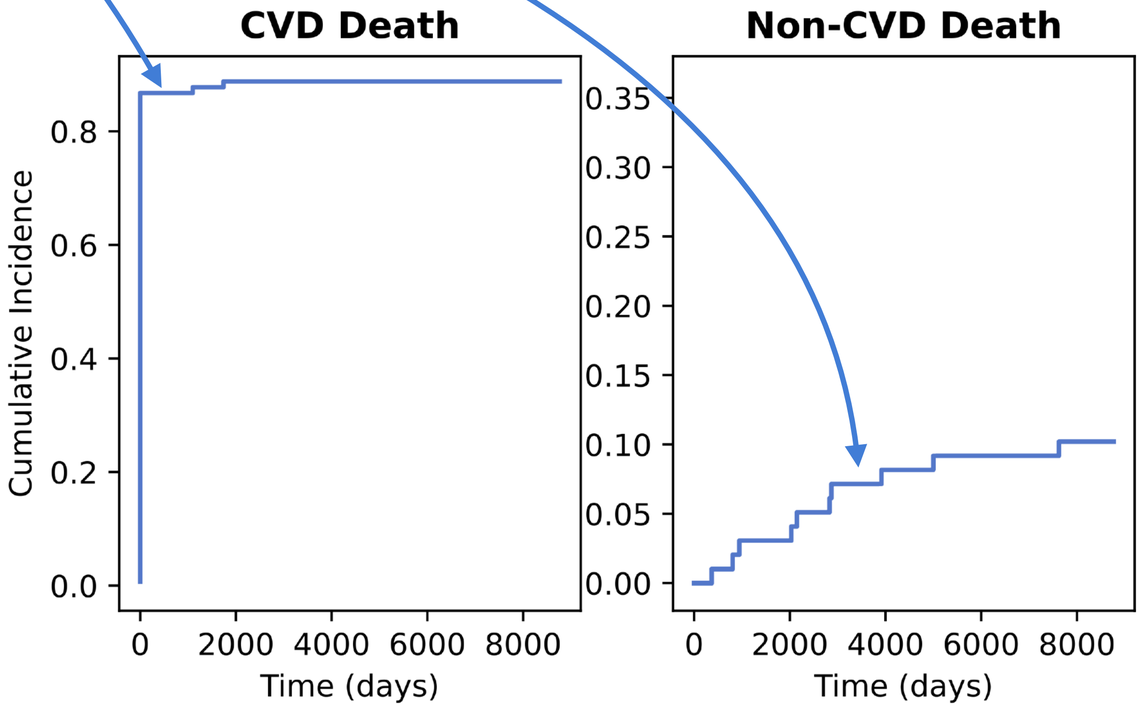
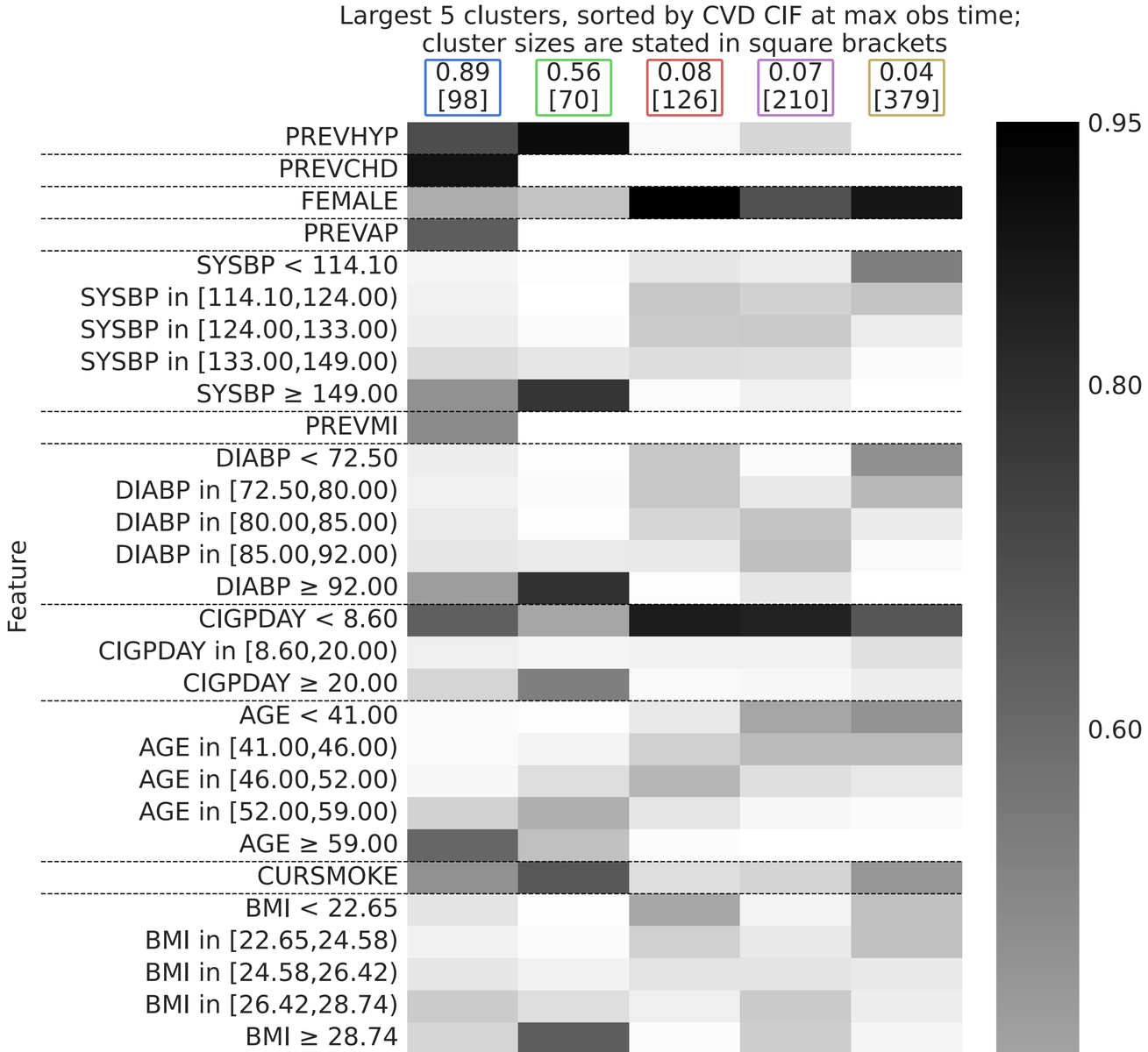


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

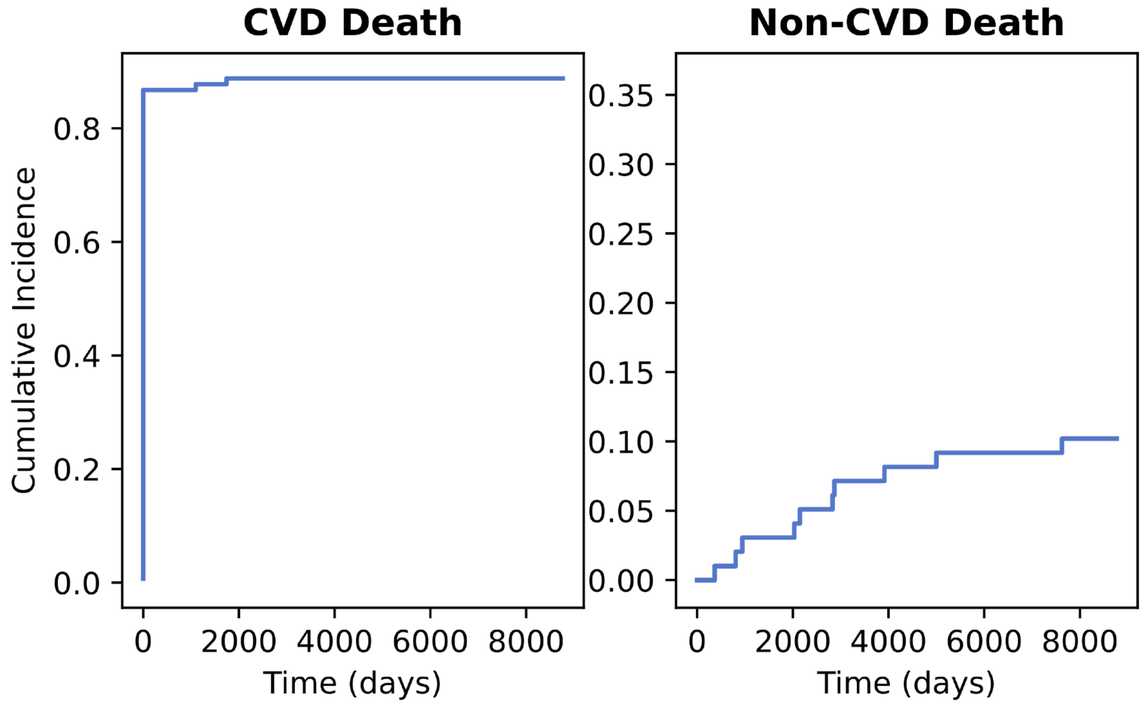
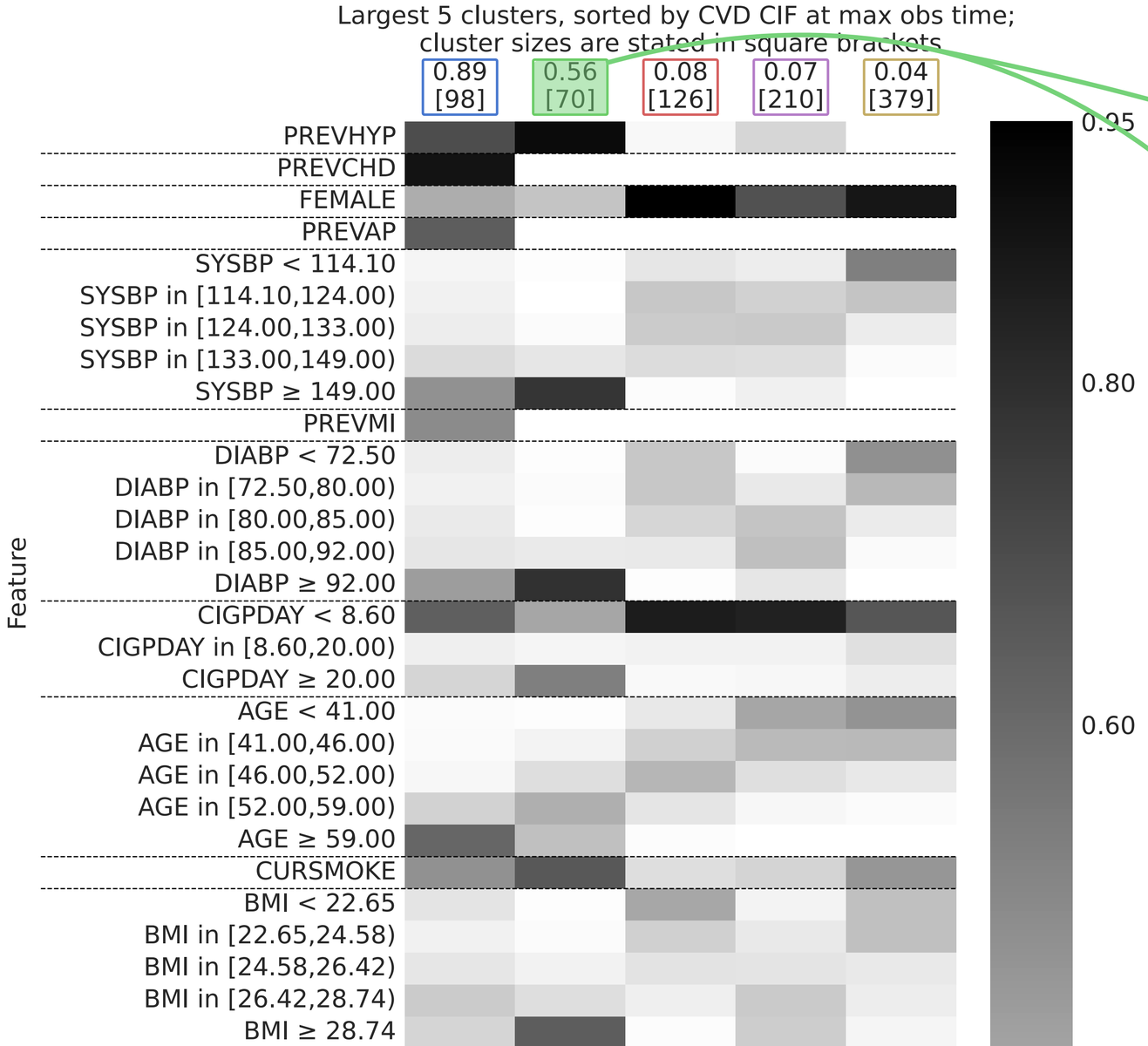


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

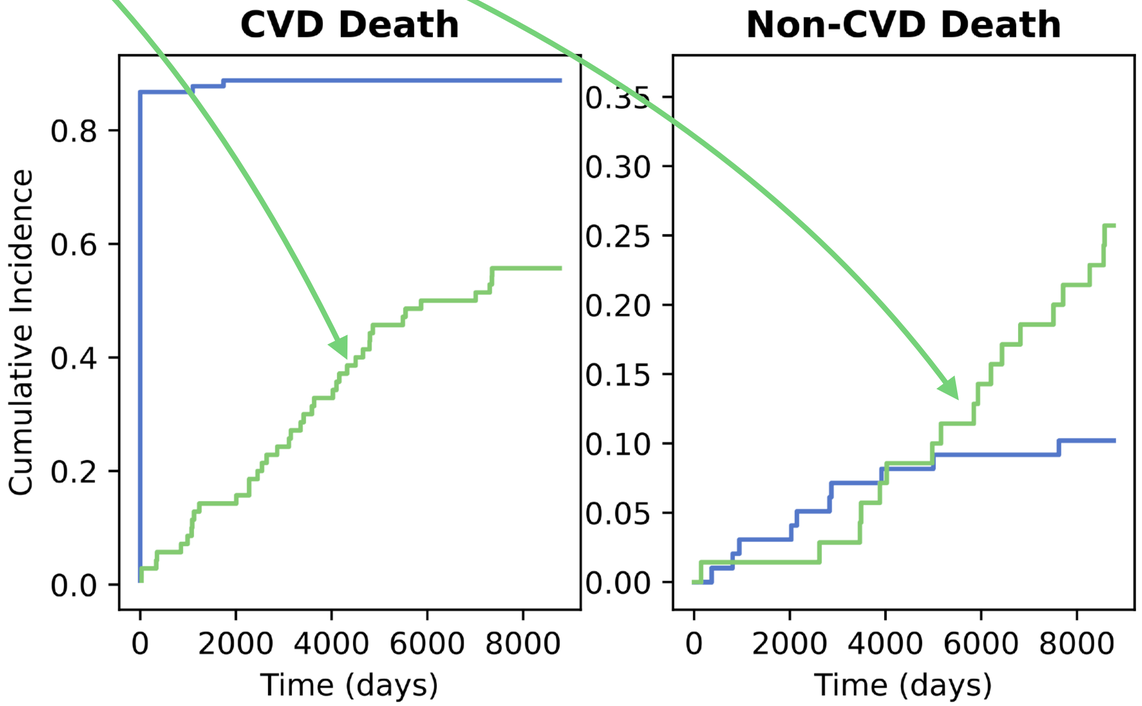
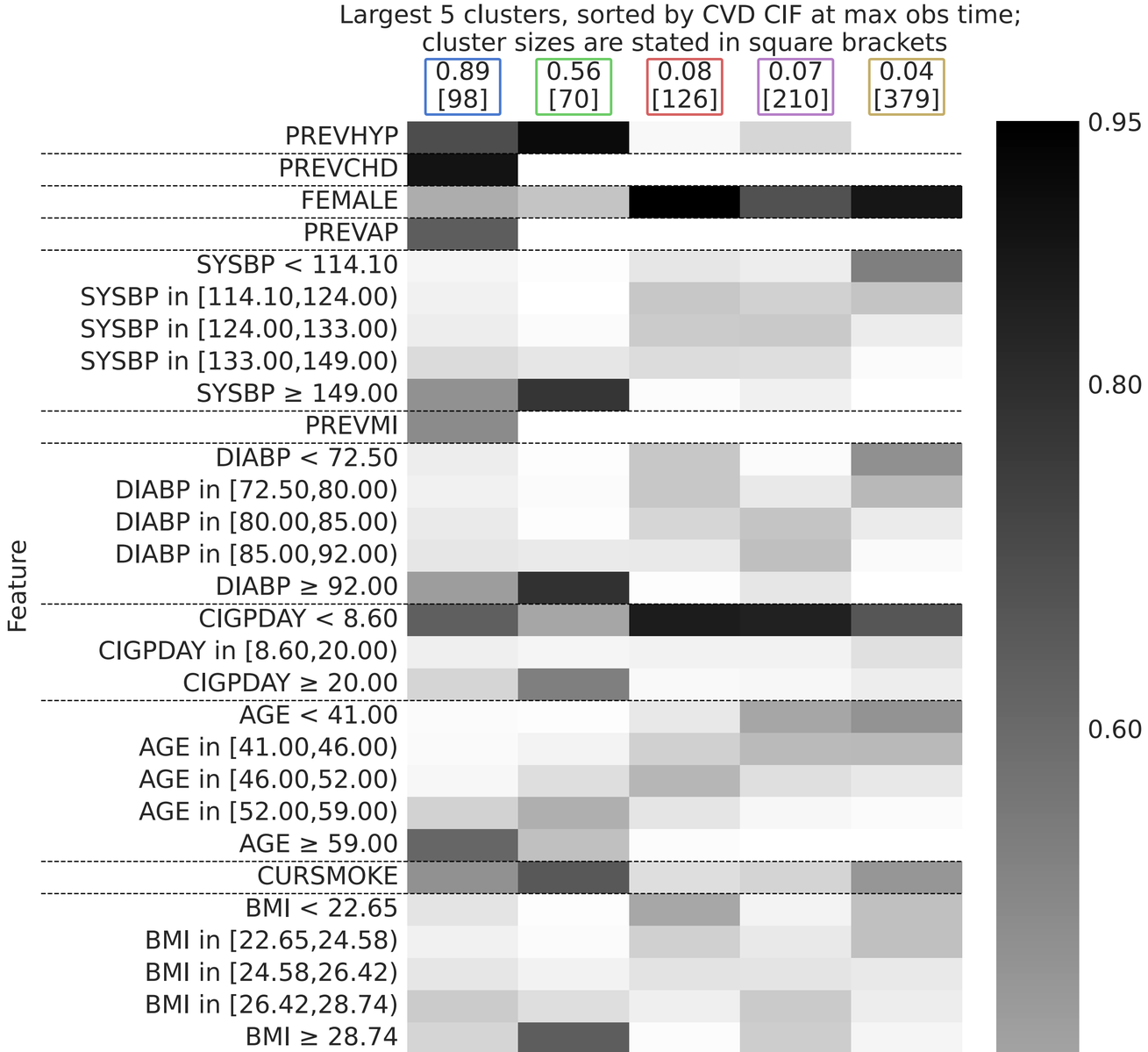


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:

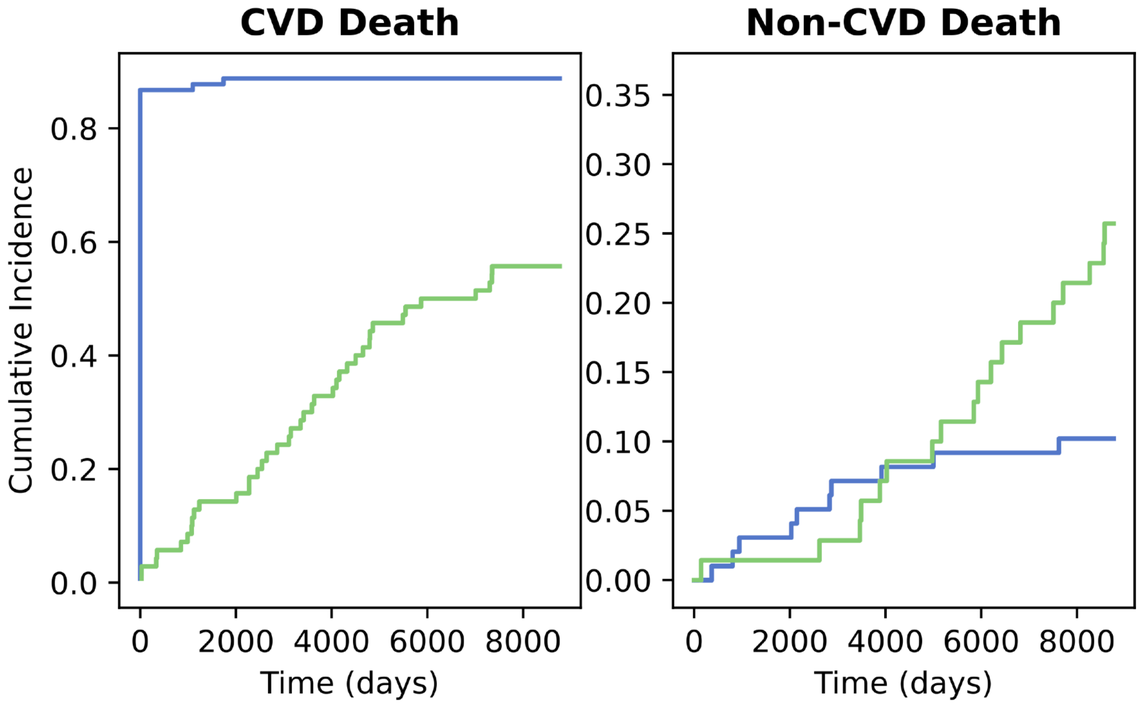
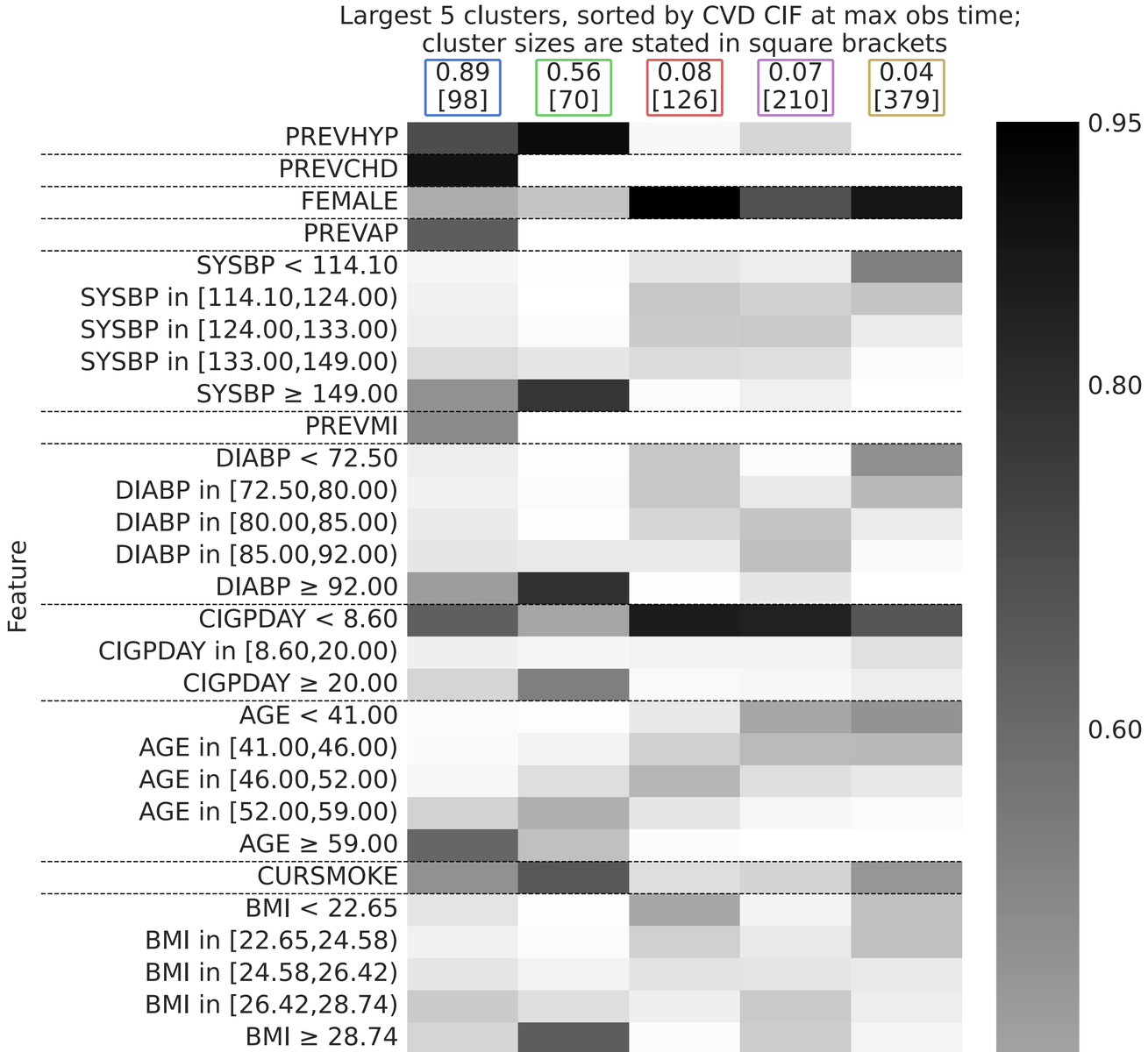
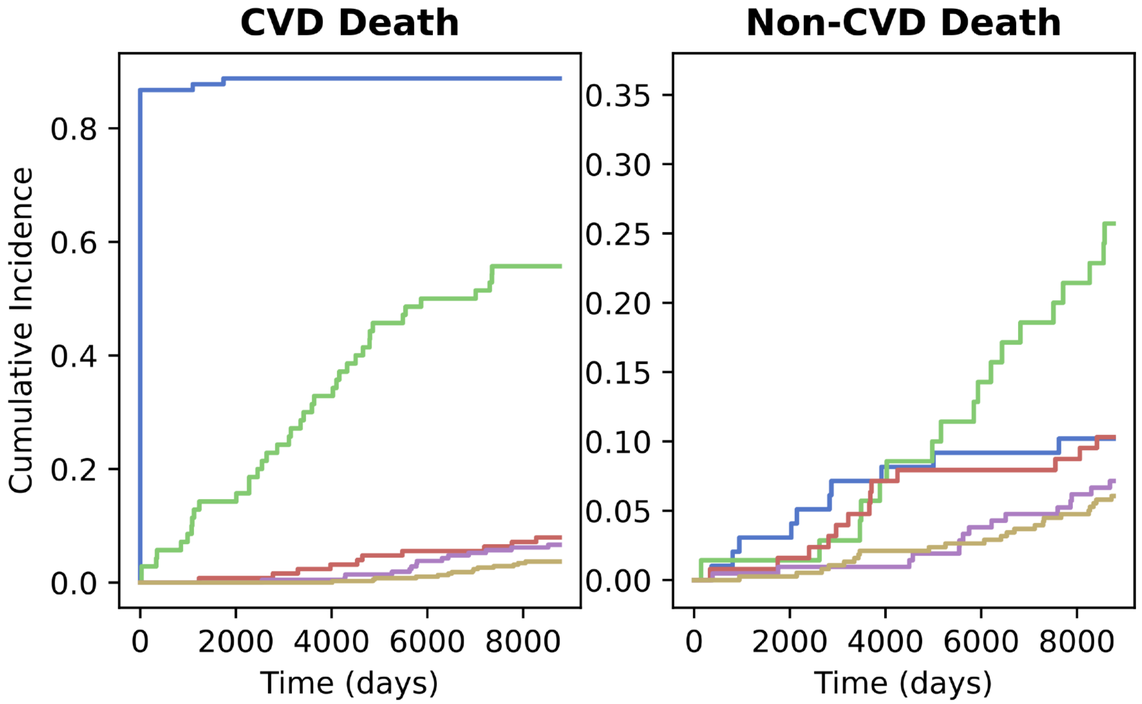


Illustration of DKAJ Model Interpretation: Framingham



Can also plot CIFs of these clusters:



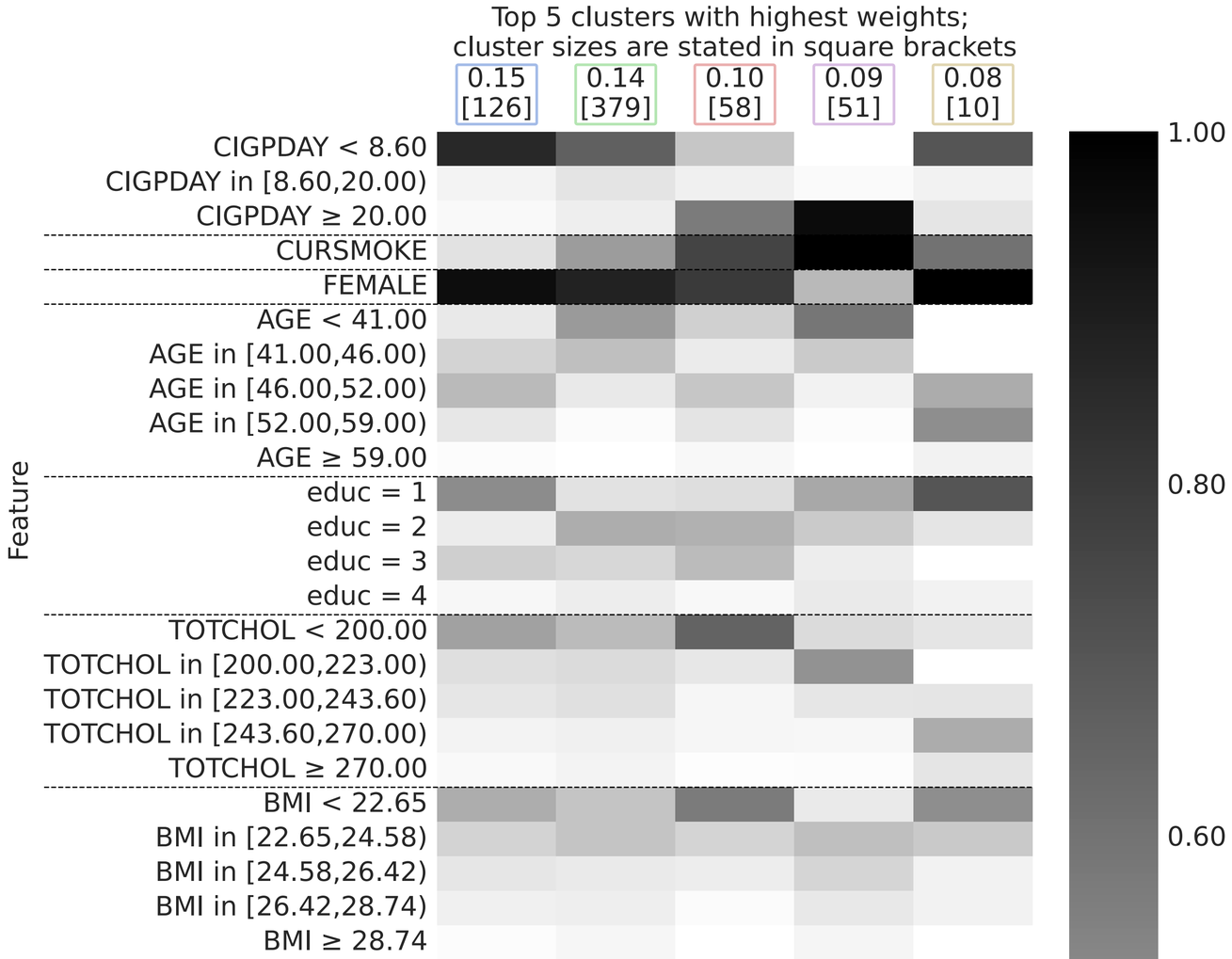
DKAJ Individual-Level Prediction Visualizations

DKAJ Individual-Level Prediction Visualizations

For randomly chosen test patient, can look at which clusters contribute the most

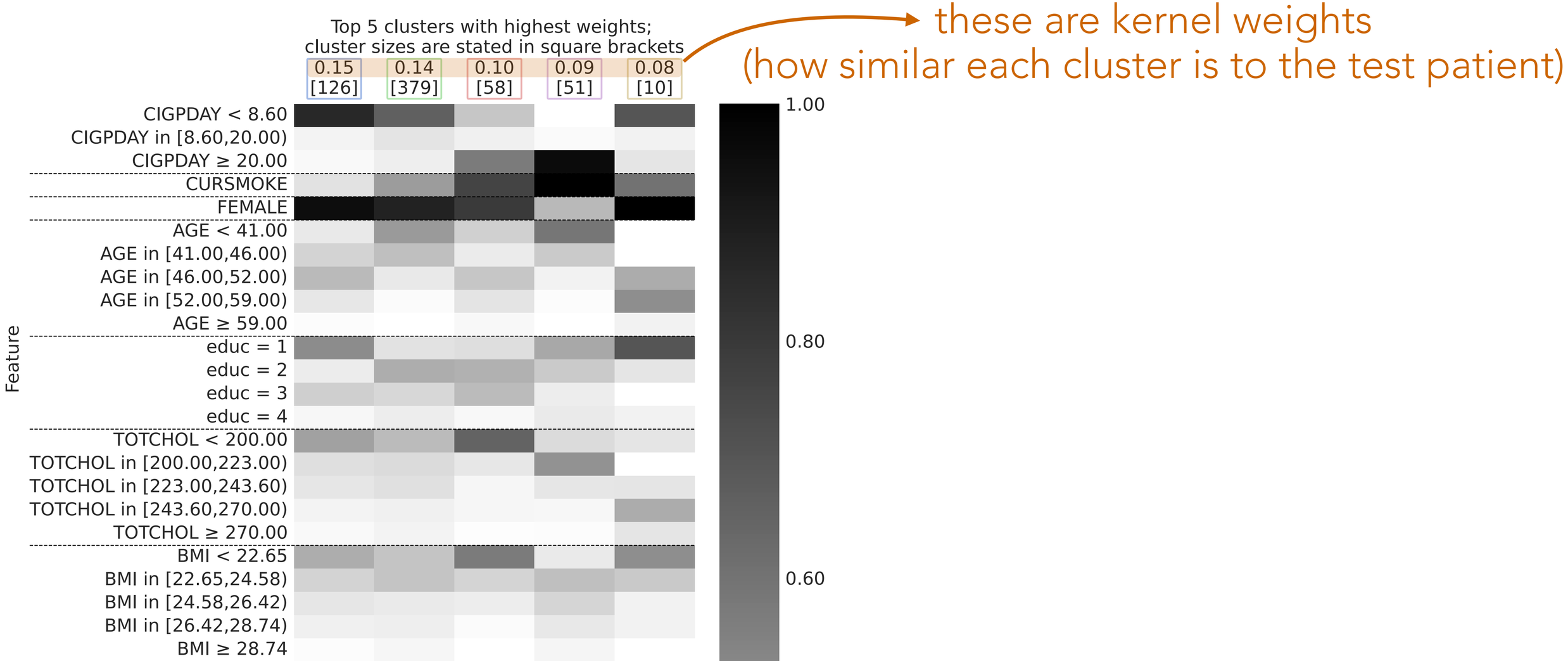
DKAJ Individual-Level Prediction Visualizations

For randomly chosen test patient, can look at which clusters contribute the most



DKAJ Individual-Level Prediction Visualizations

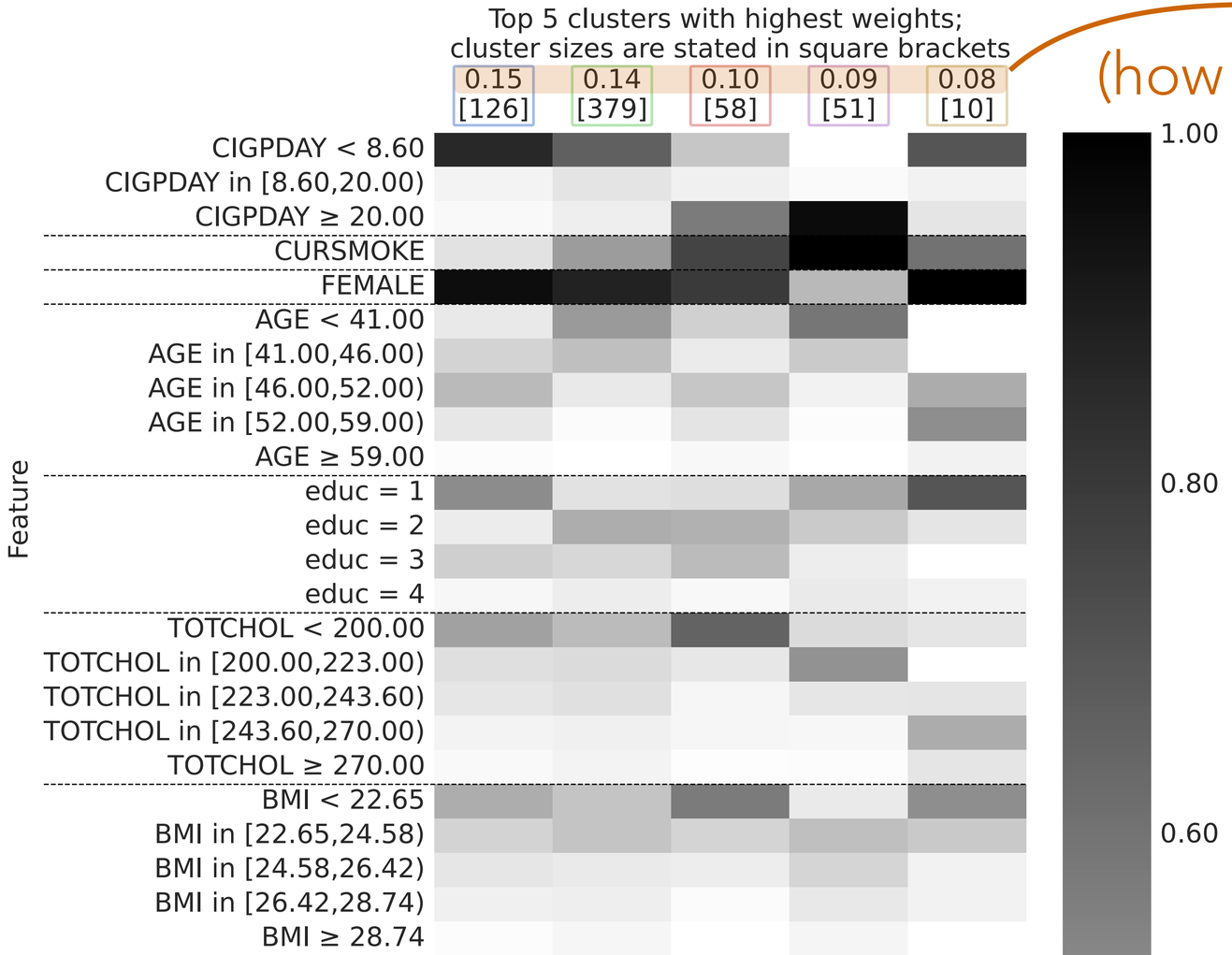
For randomly chosen test patient, can look at which clusters contribute the most

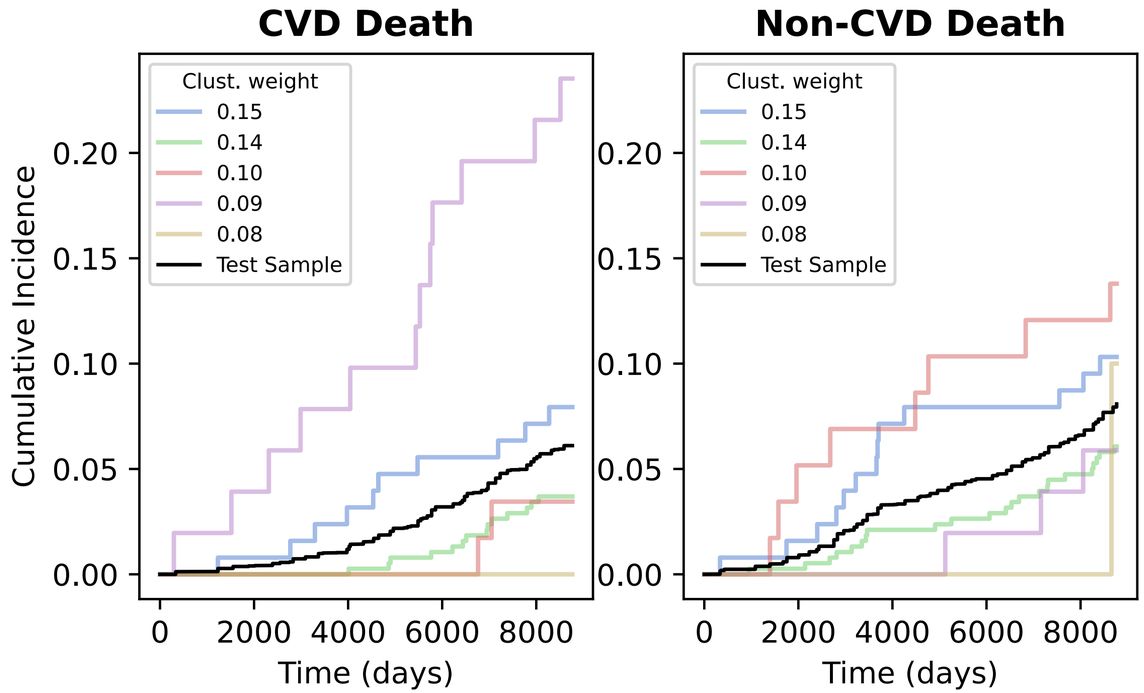


DKAJ Individual-Level Prediction Visualizations

For randomly chosen test patient, can look at which clusters contribute the most

these are kernel weights
(how similar each cluster is to the test patient)





Summary & Extensions

Summary & Extensions

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])
- Try other clustering methods (aside from one based on ε-nets)

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])
- Try other clustering methods (aside from one based on ε-nets)
- Try other kernel functions
  (e.g., other parametric form, time-dependent, event-specific)

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])
- Try other clustering methods (aside from one based on ε-nets)
- Try other kernel functions
  (e.g., other parametric form, time-dependent, event-specific)
- Generalize to multistate processes (the general setting AJ was designed for)

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])
- Try other clustering methods (aside from one based on ε-nets)
- Try other kernel functions
  (e.g., other parametric form, time-dependent, event-specific)
- Generalize to multistate processes (the general setting AJ was designed for)

Paper: https://arxiv.org/abs/2512.08063
Code: https://github.com/xiaobin-xs/dkaj

Summary & Extensions

can make cluster-level & individual-level visualizations + probe kernel function structure

Main contribution: new interpretable deep competing risks model that is competitive with various baselines

Some possible extensions:

- Establish convergence rate guarantee (only have theory for $m = 1$ [Chen 2024])
- Train with other loss functions (e.g., proper scoring rule by Alberge et al [2025])
- Try other clustering methods (aside from one based on ε-nets)
- Try other kernel functions
  (e.g., other parametric form, time-dependent, event-specific)
- Generalize to multistate processes (the general setting AJ was designed for)

Paper: https://arxiv.org/abs/2512.08063
Code: https://github.com/xiaobin-xs/dkaj

Funded by NSF CAREER award #2047981

You have reached the end of the main talk slides

Backup Slides

Also Important: The Survival Function

The survival function also shows up in later equations:

$$S(t|x) \triangleq \mathbb{P}(T > t \mid X = x)$$

This is the probability that a patient with feature vector x
experiences their earliest critical event (any of the m events) after time t

Knowing CIFs enables us to recover the survival function: $S(t|x) = 1 - \sum_{\delta=1}^{m} F_\delta(t|x)$

The AJ estimator will depend on the population-level survival function:

$$S^{\text{pop}}(t) \triangleq \mathbb{P}(T > t)$$

This can be estimated by the
Kaplan-Meier estimator [1958]

Loss Function: Negative Log Likelihood

Recall that the CIF for event δ is $F_\delta(t|x) = \mathbb{P}(\Delta^* = \delta, T \leq t \mid X = x)$

A transformed version of the CIF yields the *cause-specific hazard function*:

$$\lambda_\delta(t|x) = \frac{1}{S(t|x)} \frac{\mathrm{d}F_\delta(t|x)}{\mathrm{d}t}$$

$$\text{where } S(t|x) = 1 - \sum_{\widetilde{\delta}=1}^{m} F_{\widetilde{\delta}}(t|x)$$

Standard competing risks likelihood (under uninformative censoring):

$$\mathcal{L} = \prod_{i=1}^{n} \left(\lambda_{\Delta_i}(Y_i|X_i)\right)^{\mathbb{1}\{\Delta_i \neq 0\}} S(Y_i|X_i) \quad \Rightarrow \text{can use loss } -\log \mathcal{L}$$

If we exclude feature vectors, the classical AJ estimator maximizes this likelihood (with some pre- and post-processing steps)

measures how similar x and X_j are

$$d_{\delta,\ell}(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} K(x, X_j)$$

times event δ occurs at time t_ℓ among those who look like x

$$n_\ell(x) \triangleq \sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} K(x, X_j)$$

at risk at time t_ℓ among those who look like x

What loss function do we use to train this neural net?

$$\text{loss} = -\log \mathcal{L} = -\log \left\{ \prod_{i=1}^{n} \left(\lambda_{\Delta_i}(Y_i|X_i)\right)^{\mathbb{1}\{\Delta_i \neq 0\}} S(Y_i|X_i) \right\}$$

$$K(x, x') = \exp(-\|f(x;\theta) - f(x';\theta)\|^2)$$

kernel function parameterized by a neural net $f(\cdot;\theta)$

3. Output: $\widehat{F}_\delta^{\mathrm{KAJ}}(t|x) \triangleq \sum_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \widehat{S}^{\mathrm{KKM}}(t_{\ell-1}|x) \frac{d_{\delta,\ell}(x)}{n_\ell(x)}$

$$\widehat{S}^{\mathrm{KKM}}(t|x) \triangleq \prod_{\substack{\ell \in \{1,\ldots,L\} \\ \text{s.t. } t_\ell \leq t}} \left(1 - \frac{\sum_{\delta=1}^{m} d_{\delta,\ell}(x)}{n_\ell(x)}\right)$$

By just coding these equations up in standard neural net software (e.g., PyTorch), we can use minibatch gradient descent to learn neural net parameters θ

Training Loss (First Attempt)

Discrete time index corresponding to Y_i

With a bit of algebra, can write the negative log likelihood loss:

$$\text{loss} = \sum_{i=1}^{n} \sum_{\delta=1}^{m} \left[-\mathbb{1}\{\Delta_i = \delta\} \log \psi_{\delta,\kappa(Y_i)}(X_i;\theta) + \sum_{\ell=1}^{\kappa(Y_i)} \psi_{\delta,\ell}(X_i;\theta) \right] + \text{constant}$$

where

$$\psi_{\delta,\ell}(x;\theta) = \frac{\sum_{j=1}^{n} \mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\} \exp(-\|f(x;\theta) - f(X_j;\theta)\|^2)}{\sum_{j=1}^{n} \mathbb{1}\{Y_j \geq t_\ell\} \exp(-\|f(x;\theta) - f(X_j;\theta)\|^2)}$$

(this is a predicted cause-specific hazard value)

Problem: i-th training point's loss function uses hazard function prediction that has access to the i-th training point's ground truth

Training Loss

Discrete time index corresponding to Y_i

With a bit of algebra, can write the negative log likelihood loss:

$$\text{LOO loss} = \sum_{i=1}^{n}\sum_{\delta=1}^{m}\left[-\mathbb{1}\{\Delta_i = \delta\} \log \psi^{\neg i}_{\delta,\kappa(Y_i)}(X_i;\theta) + \sum_{\ell=1}^{\kappa(Y_i)} \psi^{\neg i}_{\delta,\ell}(X_i;\theta)\right] + \text{constant}$$

where

$$\psi^{\neg i}_{\delta,\ell}(x;\theta) = \frac{\sum_{j\neq i}\mathbb{1}\{\Delta_j = \delta, Y_j = t_\ell\}\exp(-\|f(x;\theta) - f(X_j;\theta)\|^2)}{\sum_{j\neq i}\mathbb{1}\{Y_j \geq t_\ell\}\exp(-\|f(x;\theta) - f(X_j;\theta)\|^2)}$$

(this is a predicted cause-specific hazard value)

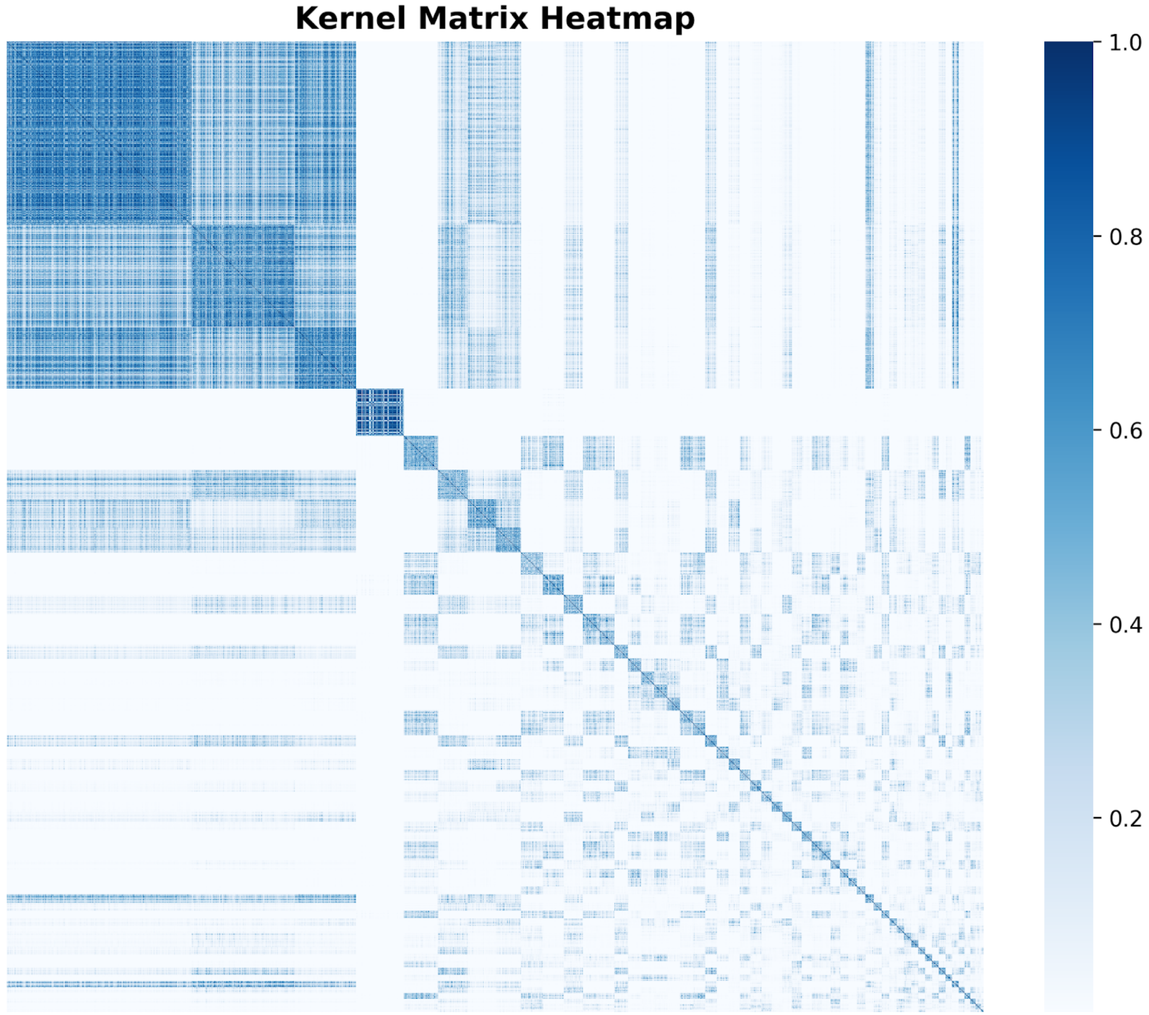
Solution: use a leave-one-out estimate instead

DKAJ Special Cases

- If $\tau = \infty$ (shaded ball has infinite radius):
  - If $\varepsilon = \infty$ for ε-net clustering (all training points are in the same cluster), we get the **classical AJ estimator** [Aalen-Johansen 1978]
  - If $\varepsilon = 0$ (each training point is in its own cluster), we get an approach that is almost the same as that of the **conditional AJ estimator** by Bladt & Furrer [2025]
    - If furthermore the kernel function always outputs 1, we also get the **classical AJ estimator** [Aalen-Johansen 1978]
- If # event types $m = 1$, we get my earlier **survival kernets** model [Chen 2024] (a deep kernel Kaplan-Meier estimator)

DKAJ Visualization of the Learned Kernel Function

- After learning the kernel function, can apply it to all training points to produce a kernel matrix
- Sort rows & columns by cluster index
- Finding: very clear block structure
- Also possible (although we haven't experimented with this yet):
  can understand hierarchical structure via hierarchical clustering algorithms that take a kernel matrix as input



Statistical Setup for Training Data

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_X$

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1

time until critical event m

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 time until critical event m

Note: it is possible for these times to be correlated!

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

   time until critical event 1 — time until critical event m

   Note: it is possible for these times to be correlated!

   Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

   and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$
3. Sample a censoring time C from $\mathbb{P}_{\mathsf{C}|\mathsf{X}}(\cdot|X)$

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

3. Sample a censoring time C from $\mathbb{P}_{\mathsf{C}|\mathsf{X}}(\cdot|X)$

assume ties in times happen with prob. 0, which we get if, e.g., $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|x)$ & $\mathbb{P}_{\mathsf{C}|\mathsf{X}}(\cdot|x)$ are absolutely continuous for all x

Statistical Setup for Training Data

Model each generic training point (X, Y, Δ) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

3. Sample a censoring time C from $\mathbb{P}_{\mathsf{C}|\mathsf{X}}(\cdot|X)$
4. Finally, set $Y \triangleq \min\{T, C\}$, and $\Delta \triangleq \begin{cases} 0 & \text{if } Y = C \\ \Delta^* & \text{otherwise} \end{cases}$

assume ties in times happen with prob. 0, which we get if, e.g., $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|x)$ & $\mathbb{P}_{\mathsf{C}|\mathsf{X}}(\cdot|x)$ are absolutely continuous for all x

Statistical Setup for Test Data

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Statistical Setup for Test Data

For test data, we do not model censoring times

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Statistical Setup for Test Data

For test data, we do not model censoring times

Model each generic test point (X, T, Δ^*) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Statistical Setup for Test Data

For test data, we do not model censoring times

Model each generic test point (X, T, Δ^*) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Formally, the CIF is defined as: $F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$

Statistical Setup for Test Data

For test data, we do not model censoring times

Model each generic test point (X, T, Δ^*) to be sampled as follows:

1. Sample feature vector X from $\mathbb{P}_{\mathsf{X}}$
2. Sample the time until each of the m events happens; we do this in a "joint" fashion by sampling the length-m vector $(T_1, \ldots, T_m)$ from $\mathbb{P}_{\mathsf{T}|\mathsf{X}}(\cdot|X)$

time until critical event 1 — time until critical event m

Note: it is possible for these times to be correlated!

Denote the time until the earliest event by $T \triangleq \min_{\delta \in \{1,\ldots,m\}} T_\delta$

and the earliest event by $\Delta^* \triangleq \arg \min_{\delta \in \{1,\ldots,m\}} T_\delta$

Formally, the CIF is defined as: $F_\delta(t|x) \triangleq \mathbb{P}\big(\Delta^* = \delta, T \leq t \;\big|\; X = x\big)$

The AJ estimator estimates the population-level version: $F_\delta^{\text{pop}}(t) \triangleq \mathbb{P}(\Delta^* = \delta, T \leq t)$

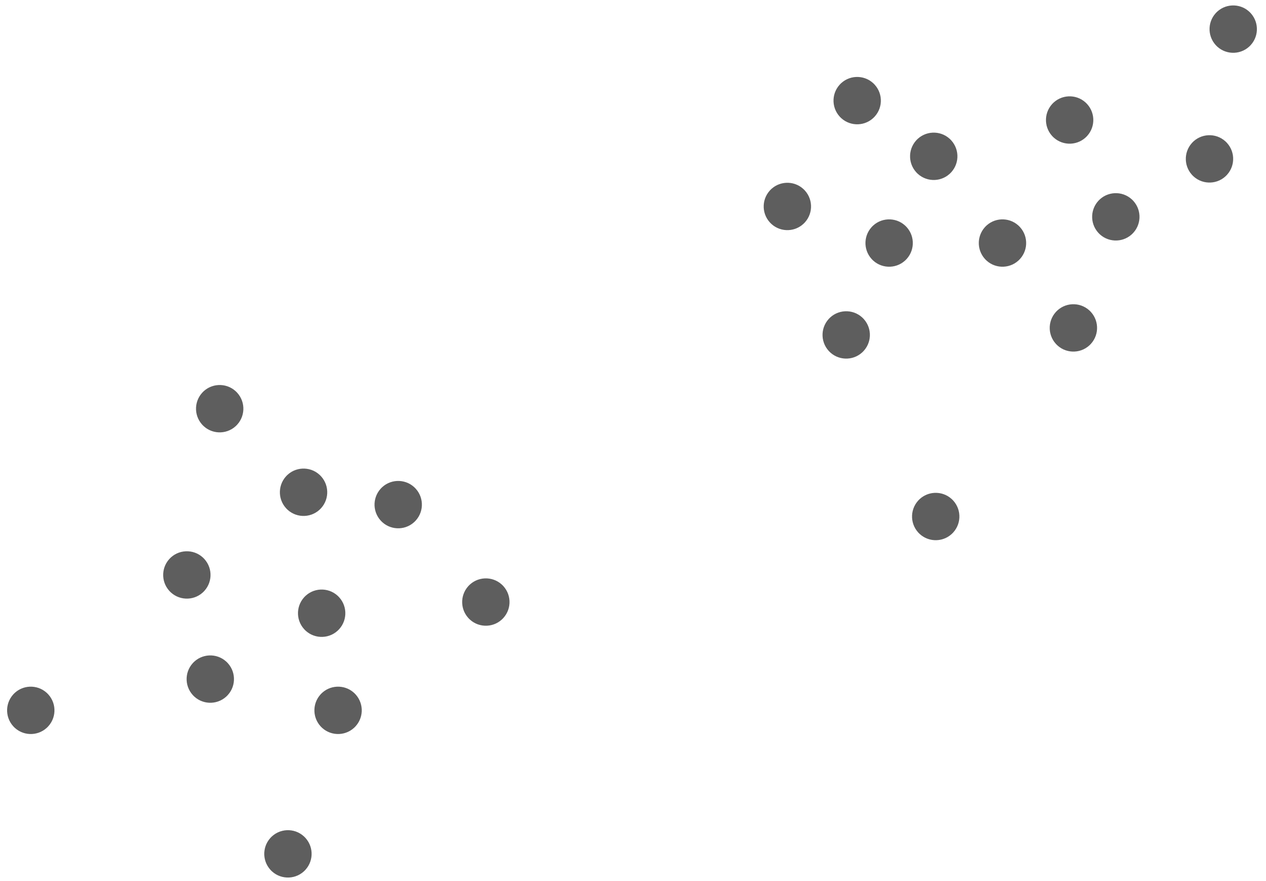
Clustering with an ε-net

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

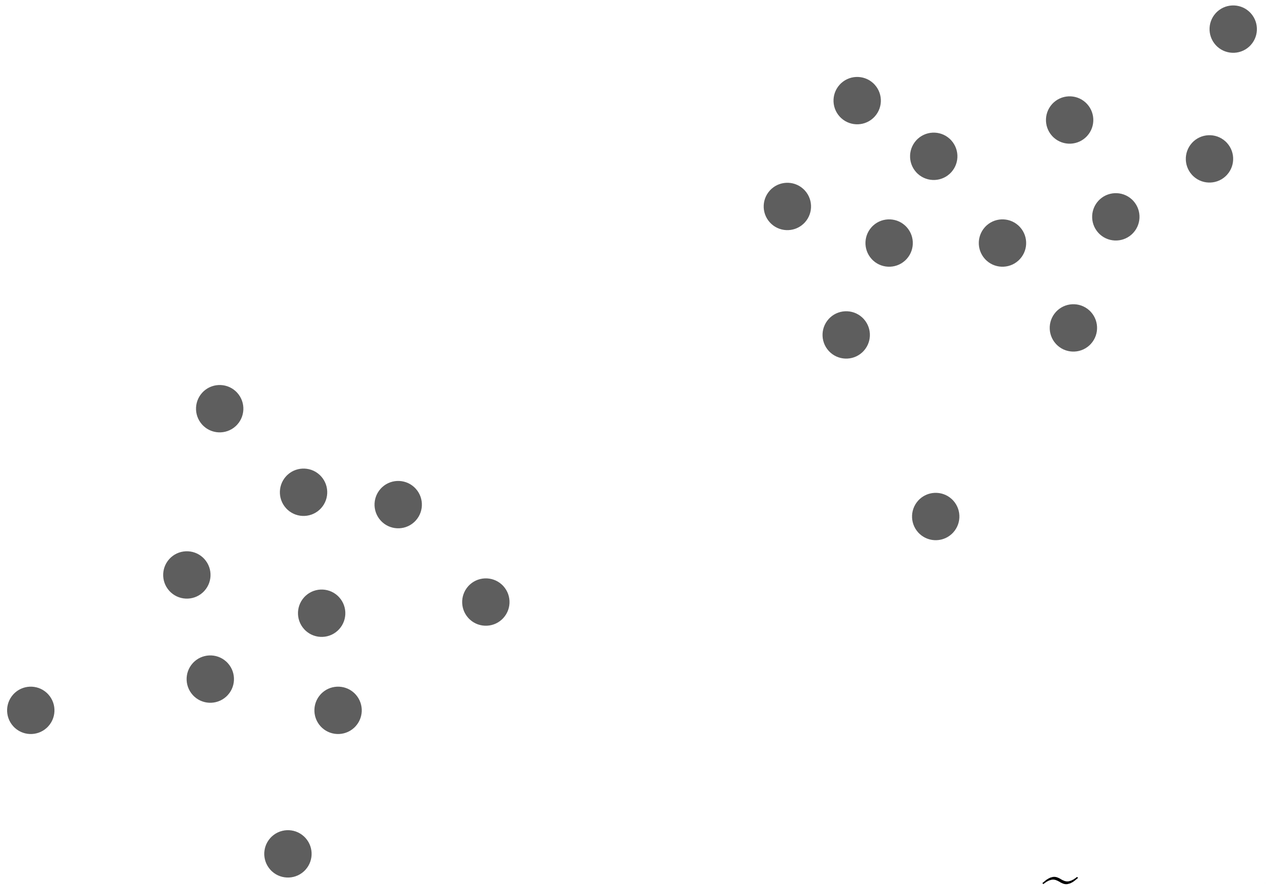
Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)



Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

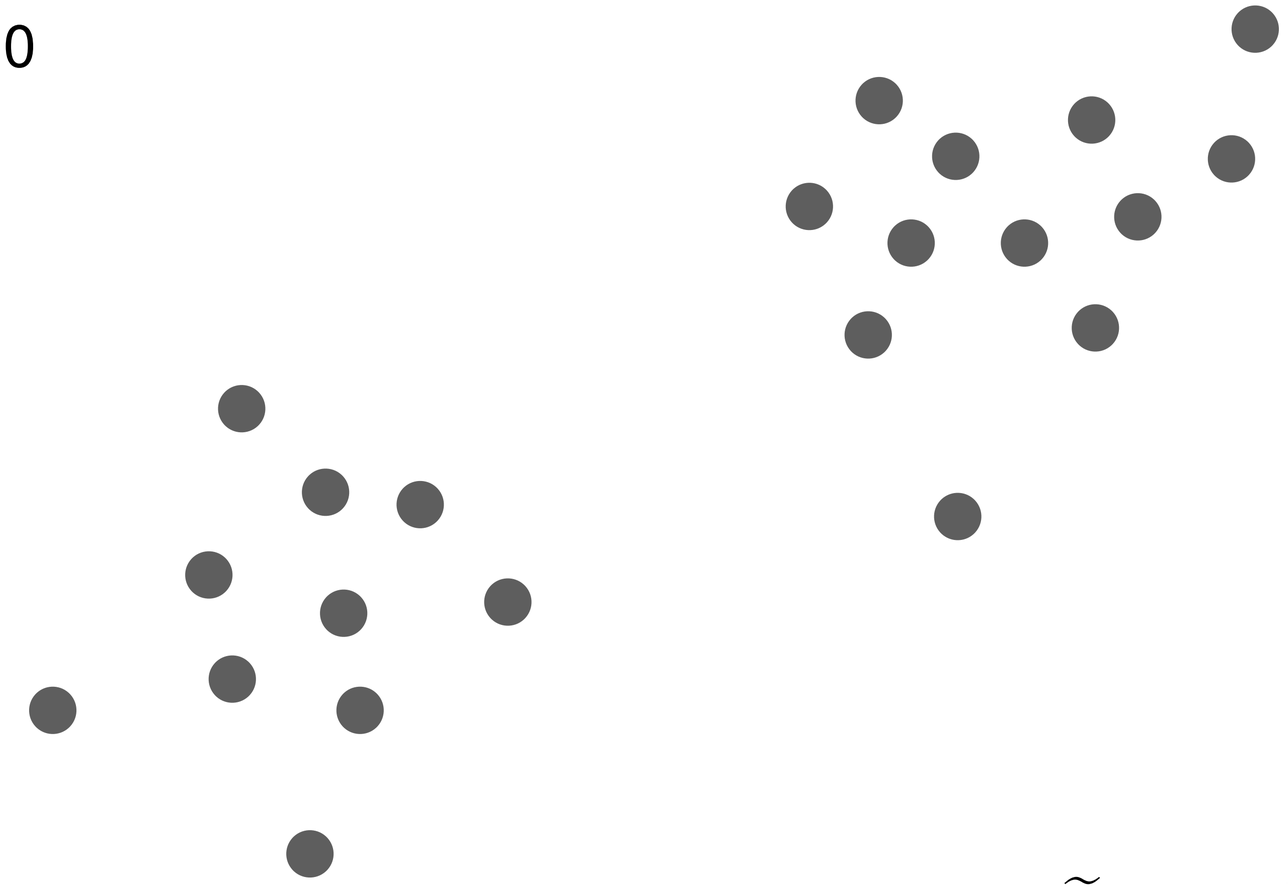


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



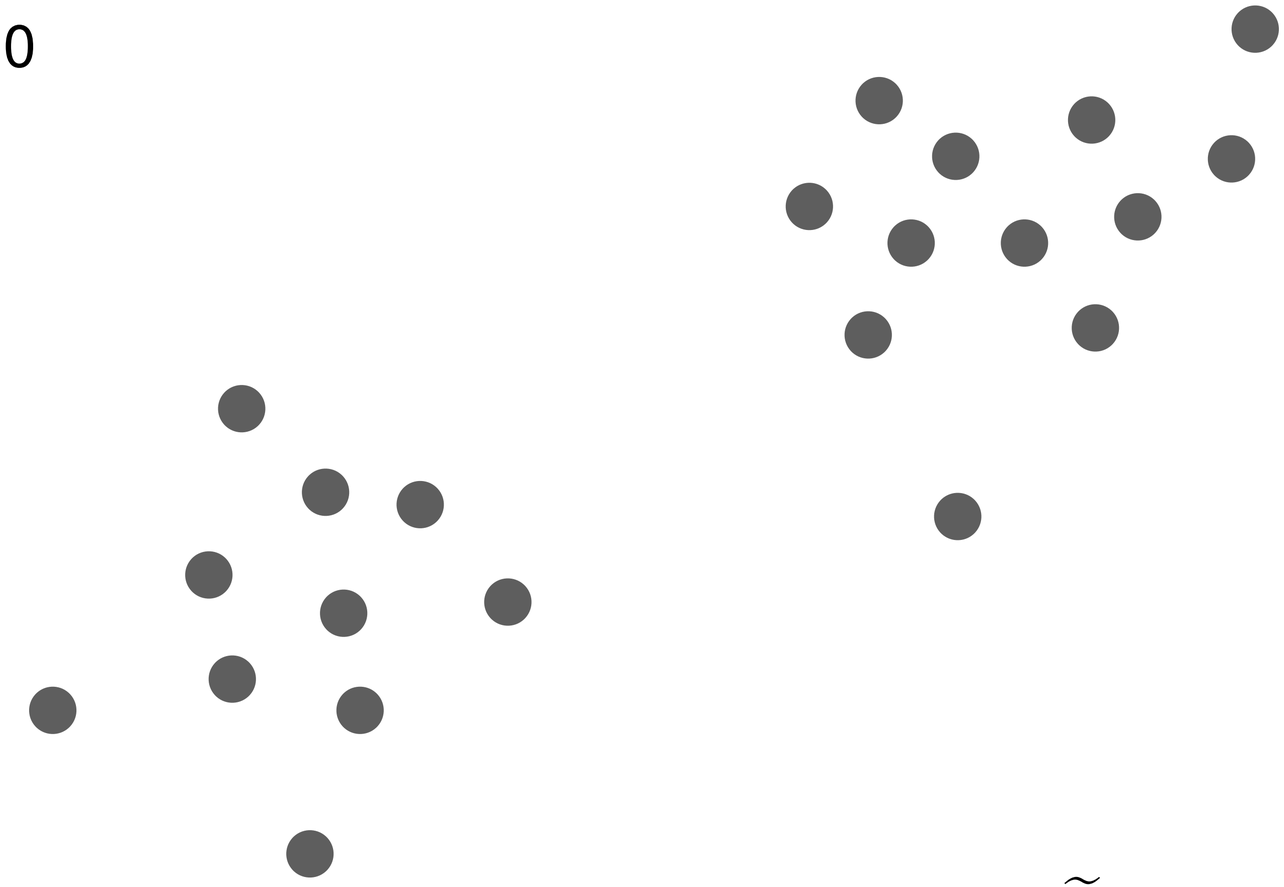
These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

1. Pick a point without cluster assignment



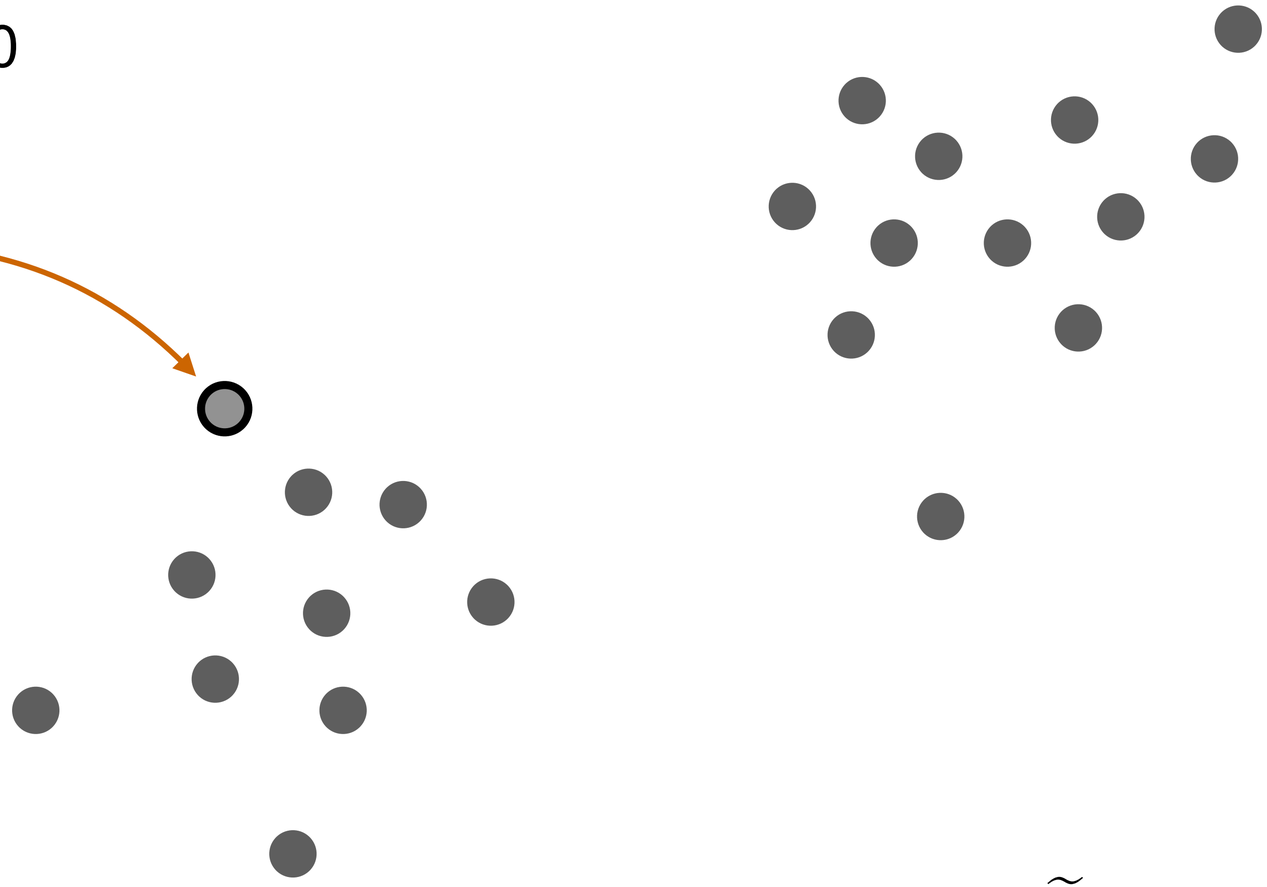
These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

1. Pick a point without cluster assignment



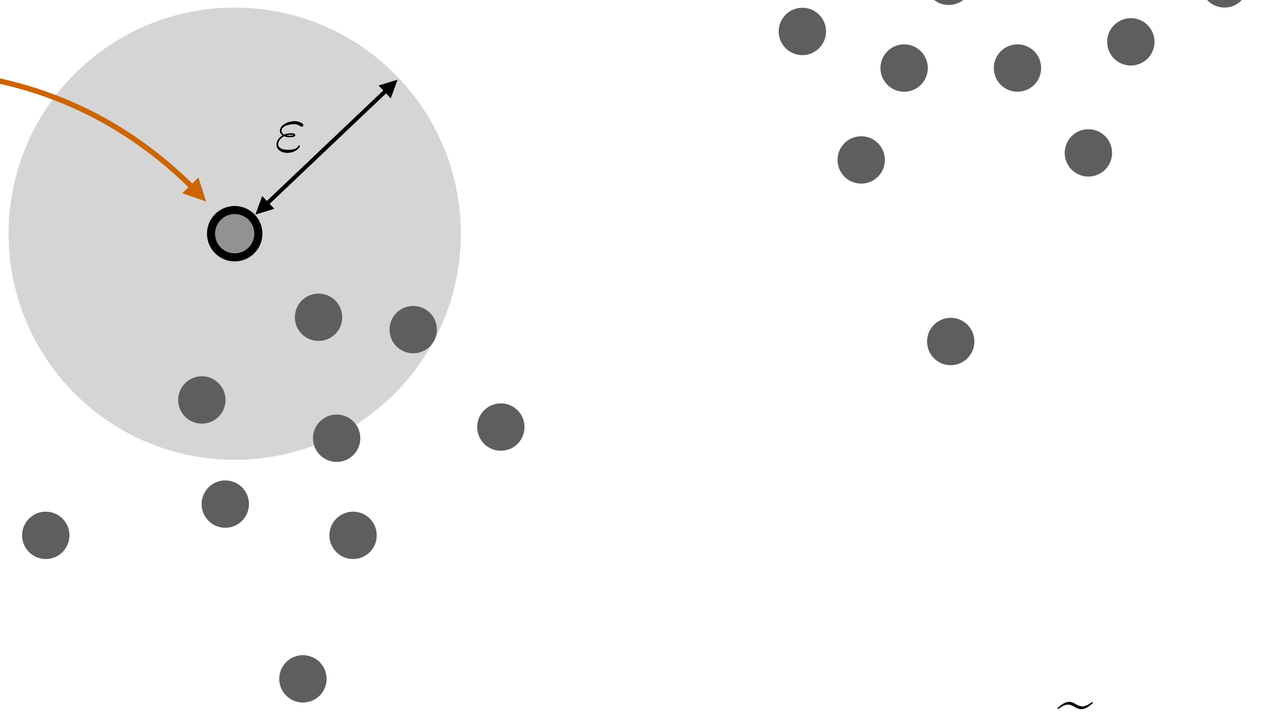
These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

1. Pick a point without cluster assignment



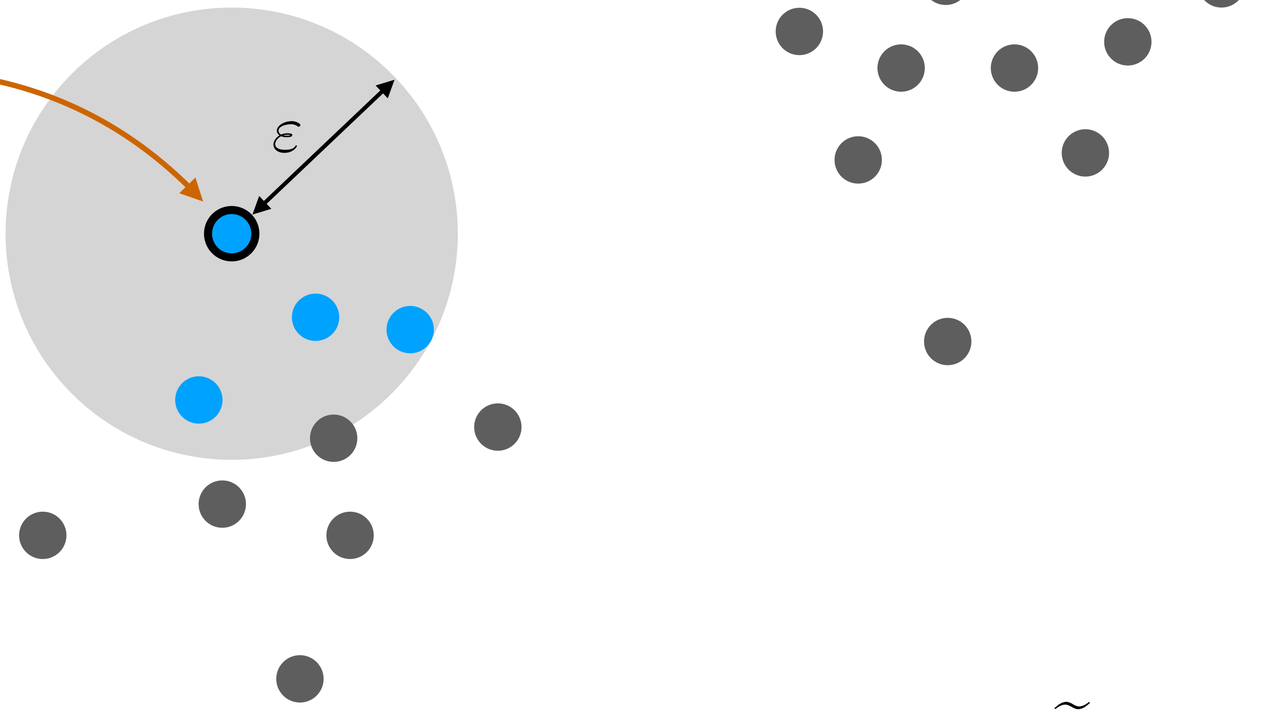
These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

1. Pick a point without cluster assignment
2. Assign points in shaded ball not already clustered to new cluster

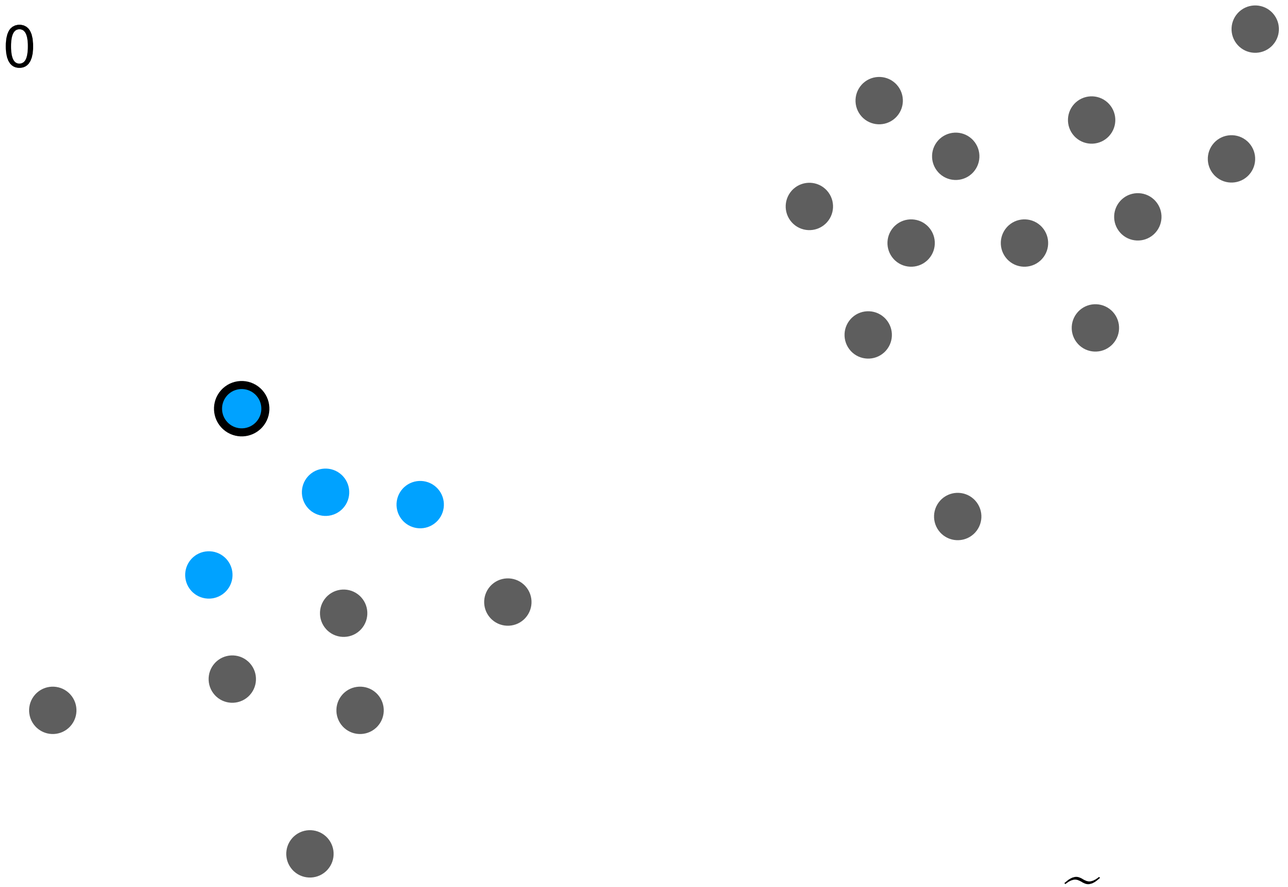


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

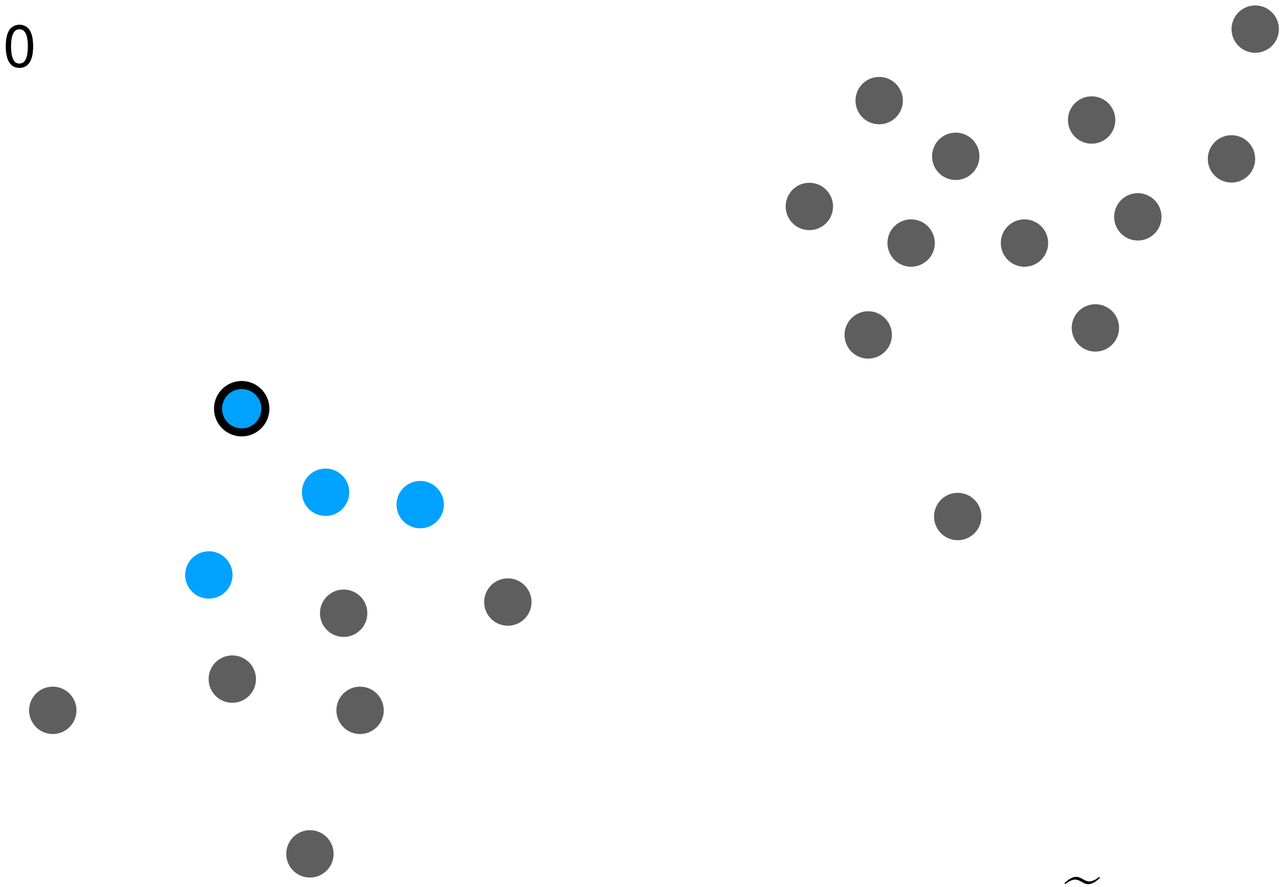


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



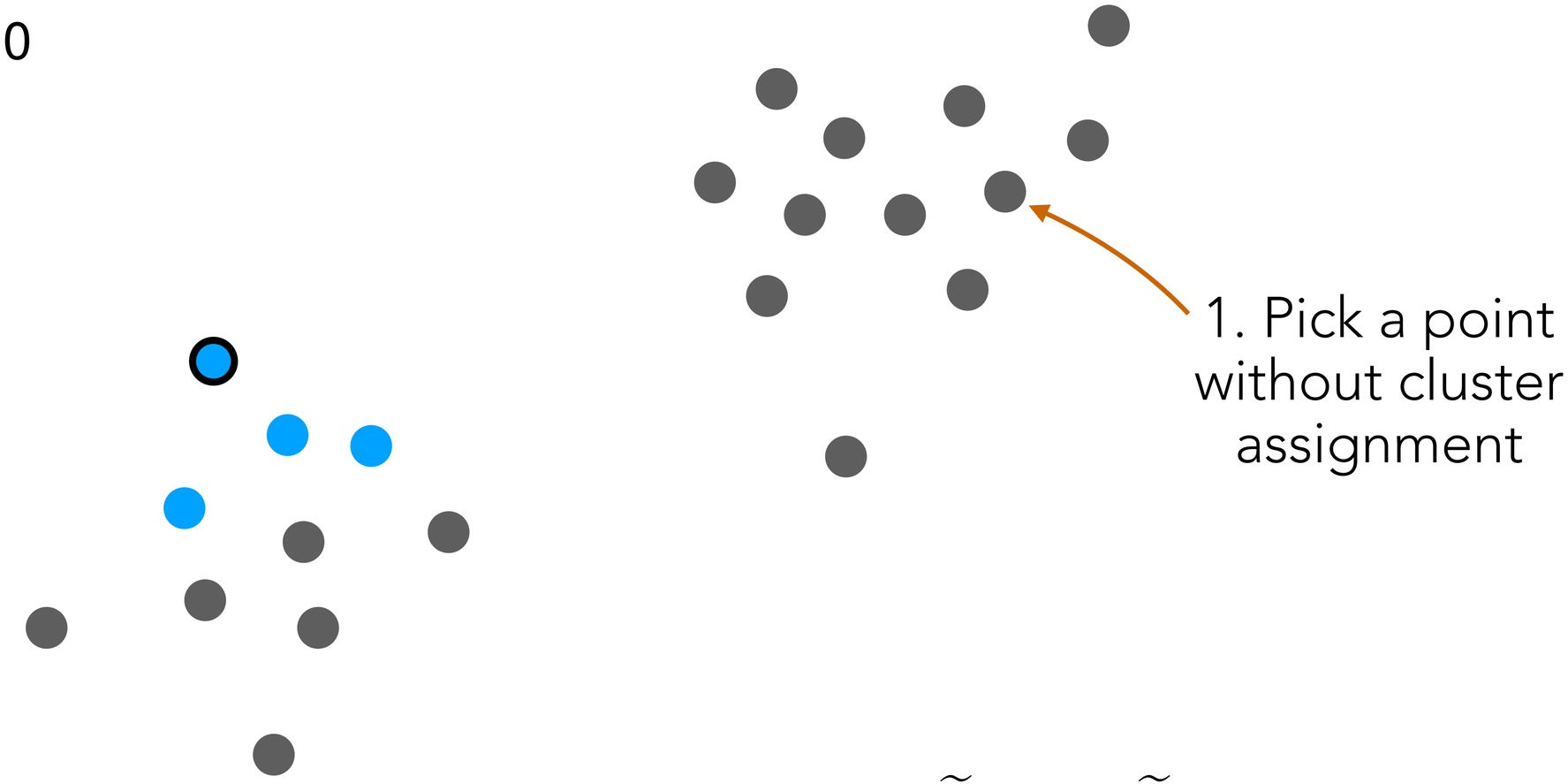
1. Pick a point without cluster assignment

These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

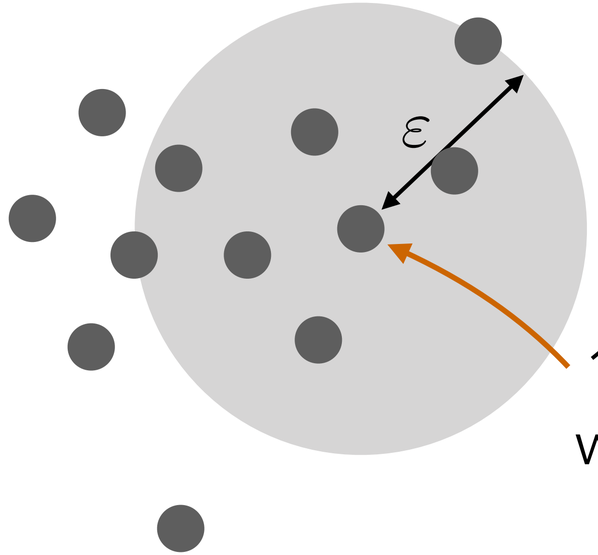
Parameter: $\varepsilon > 0$



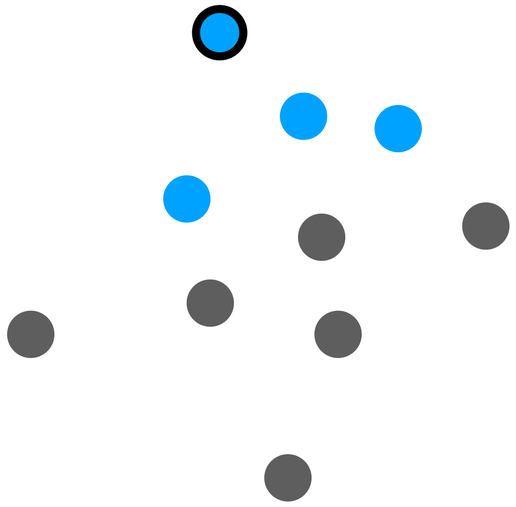
Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



1. Pick a point without cluster assignment

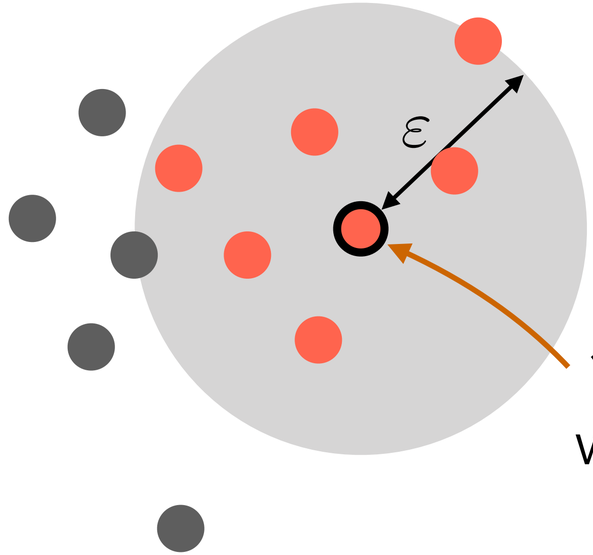


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

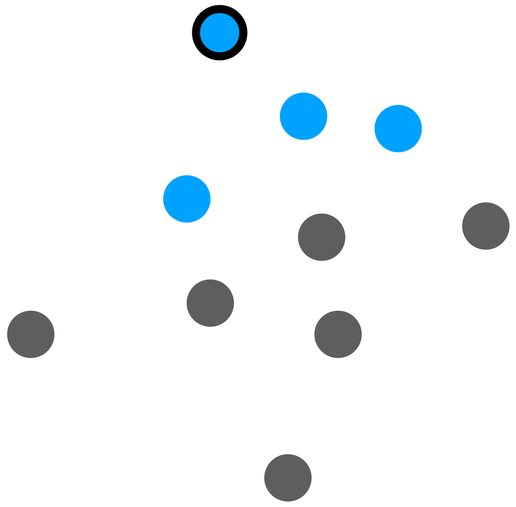
Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



1. Pick a point without cluster assignment
2. Assign points in shaded ball not already clustered to new cluster

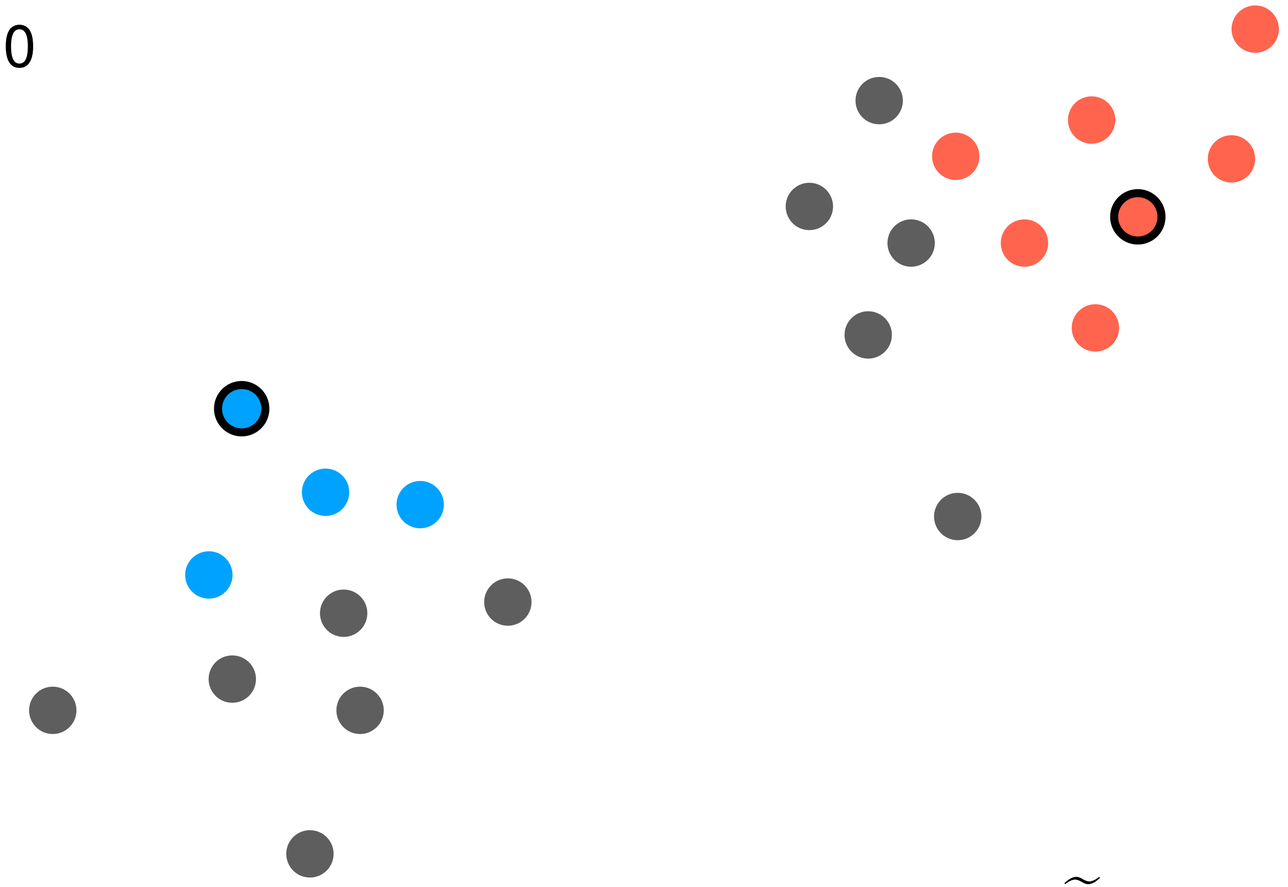


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

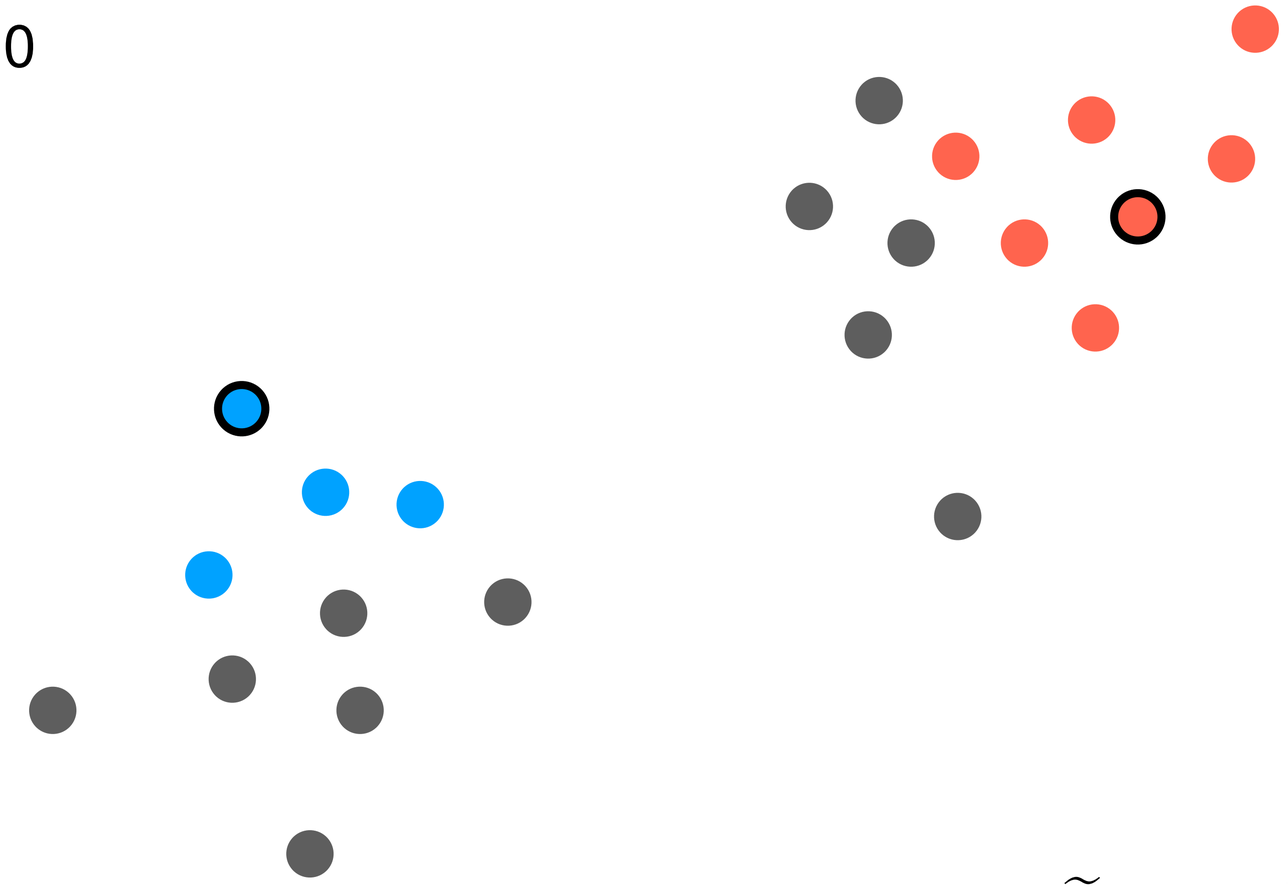


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



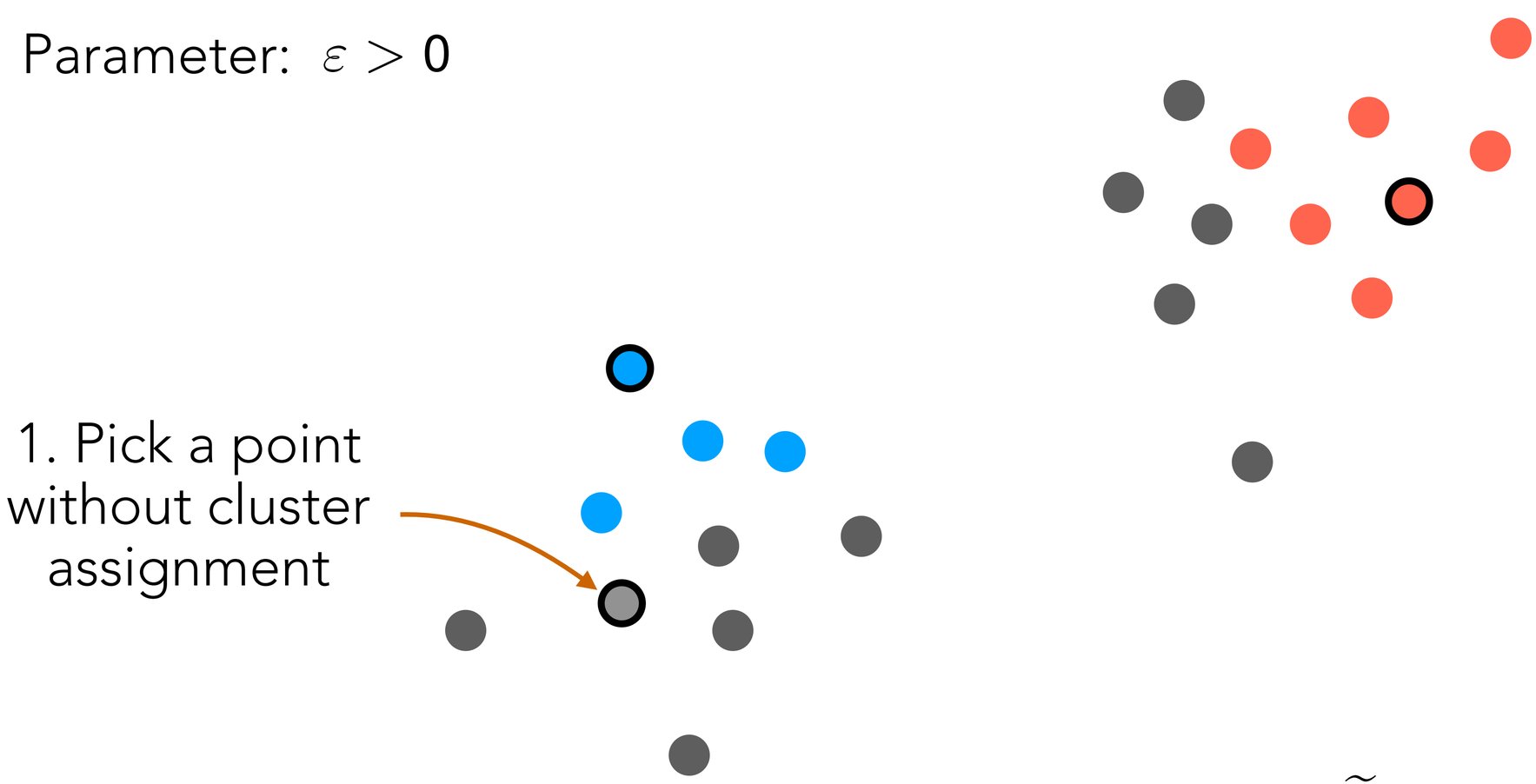
1. Pick a point without cluster assignment

These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

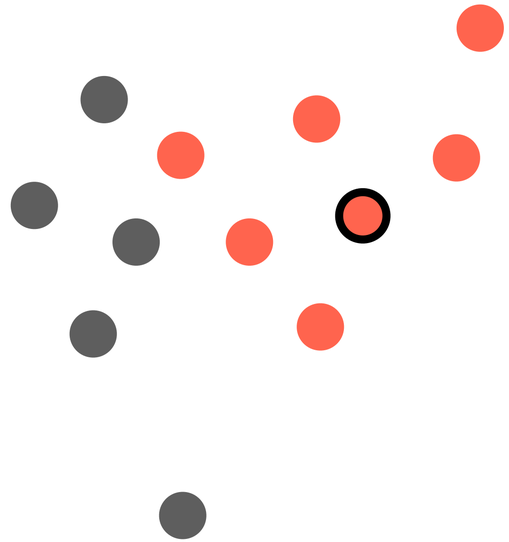


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

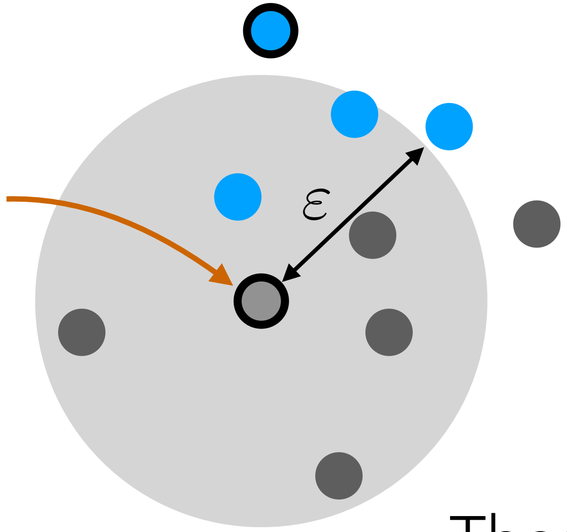
Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



1. Pick a point without cluster assignment

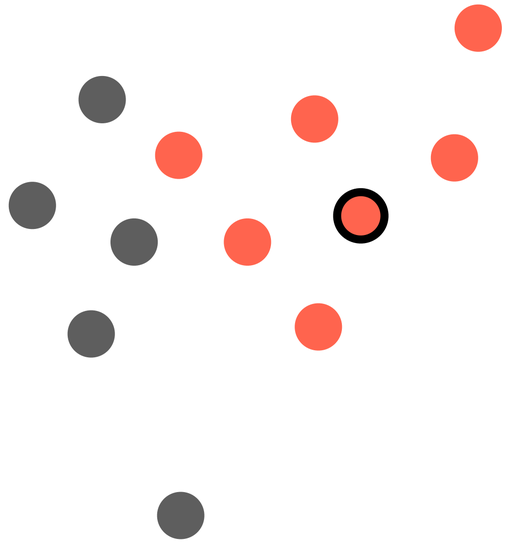


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

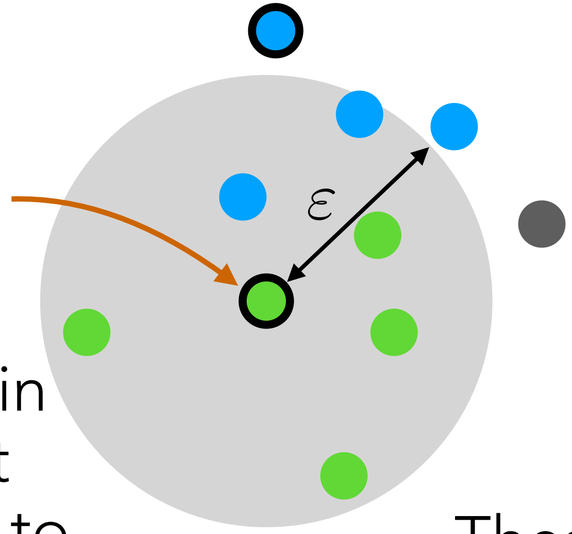
Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



1. Pick a point without cluster assignment



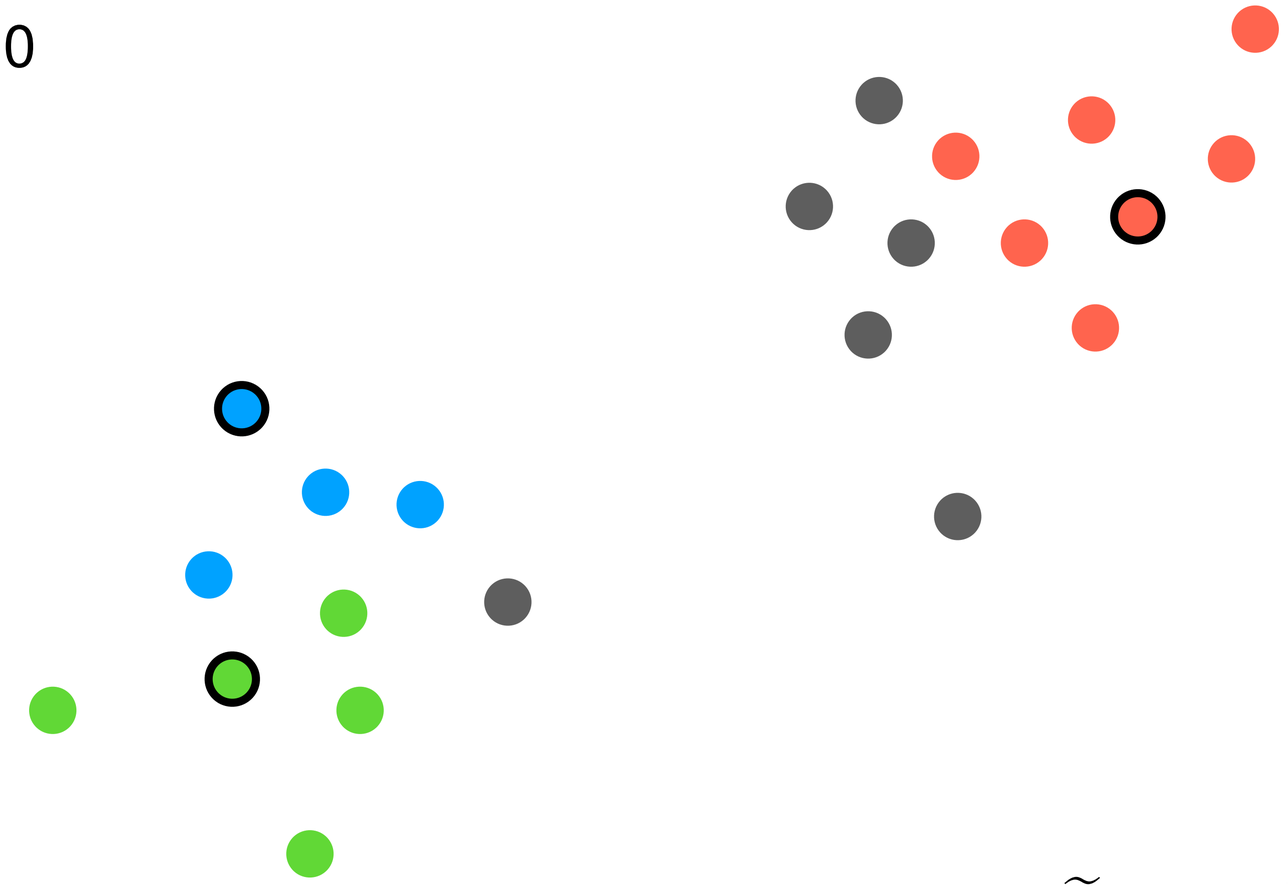
2. Assign points in shaded ball not already clustered to new cluster

These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

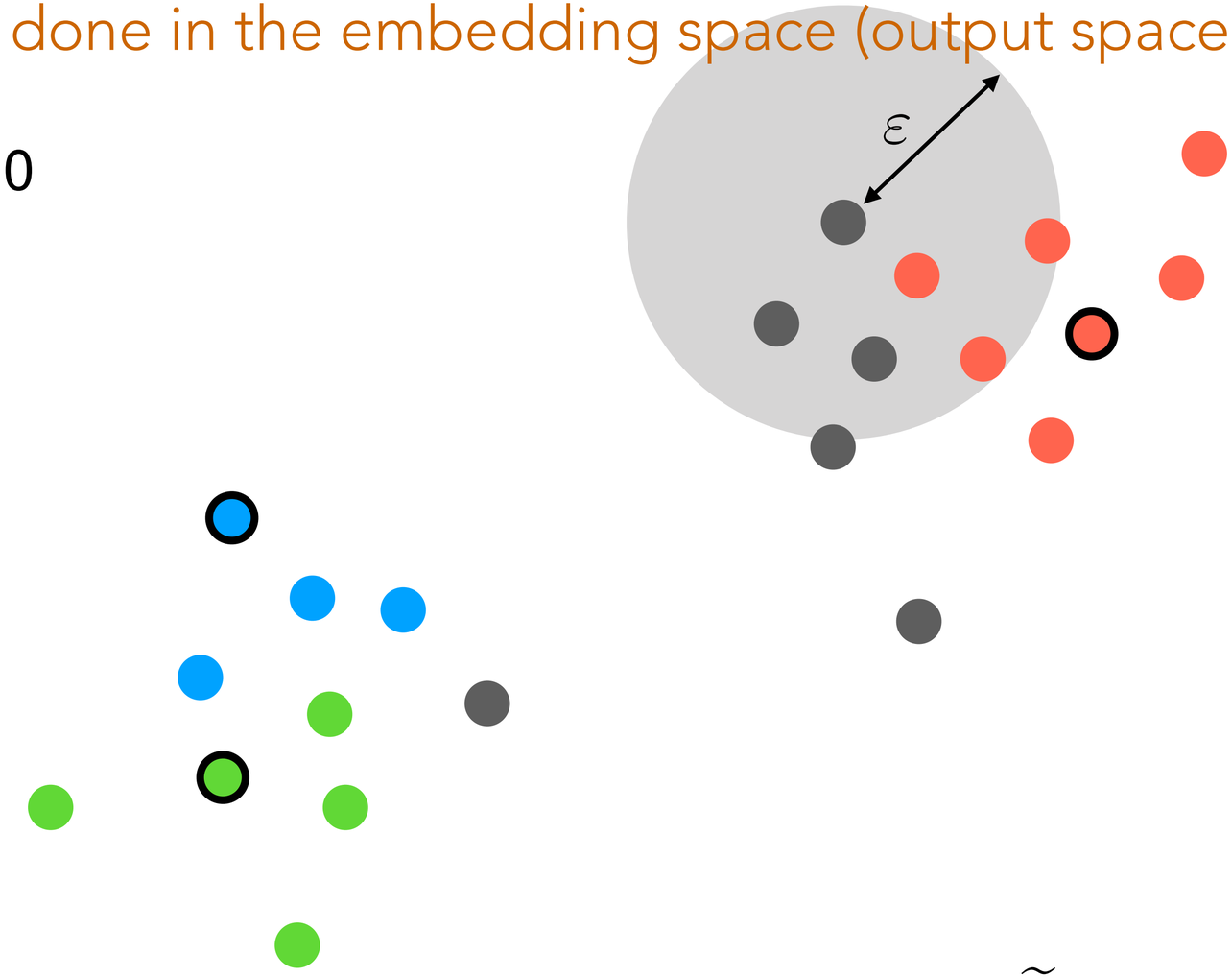


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

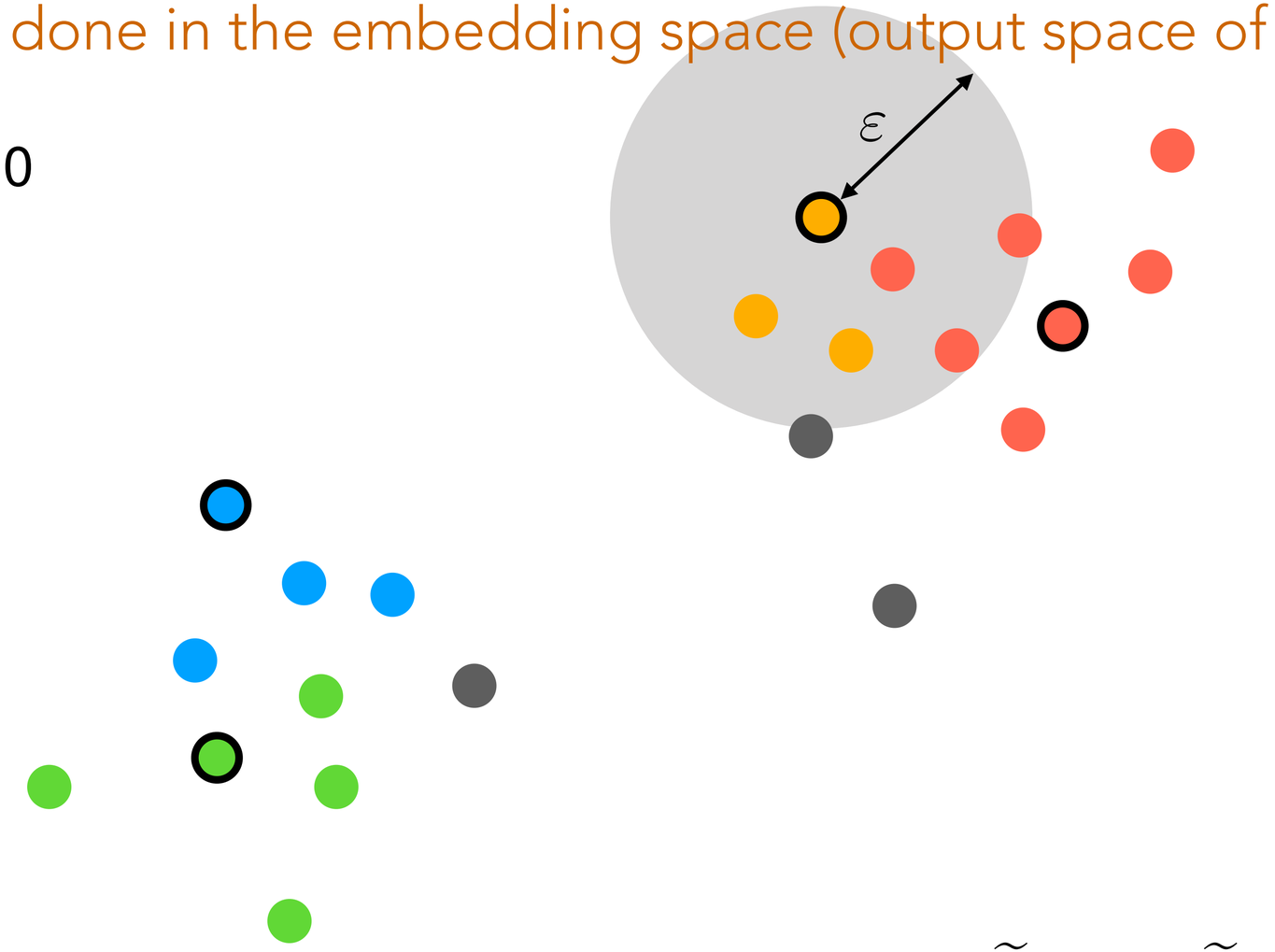


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

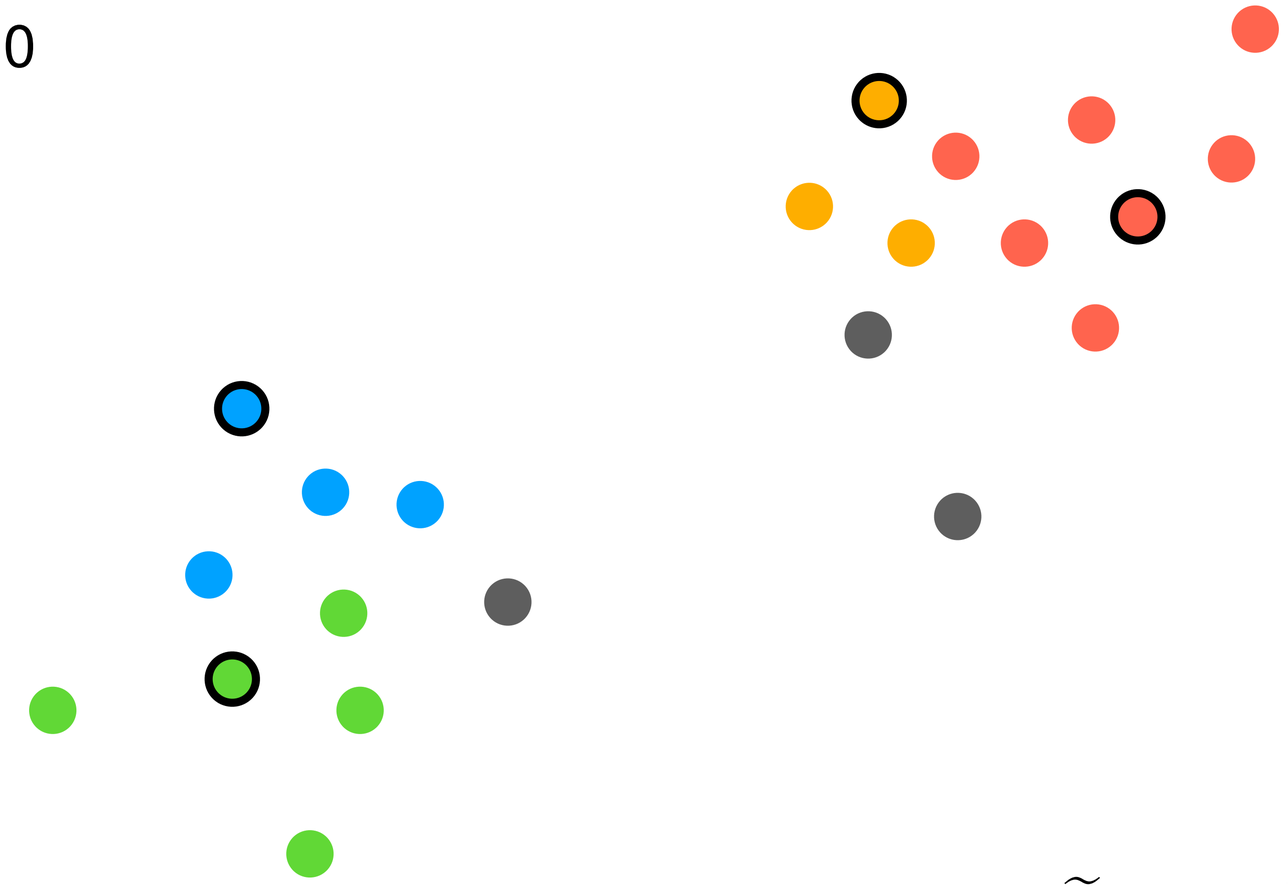


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

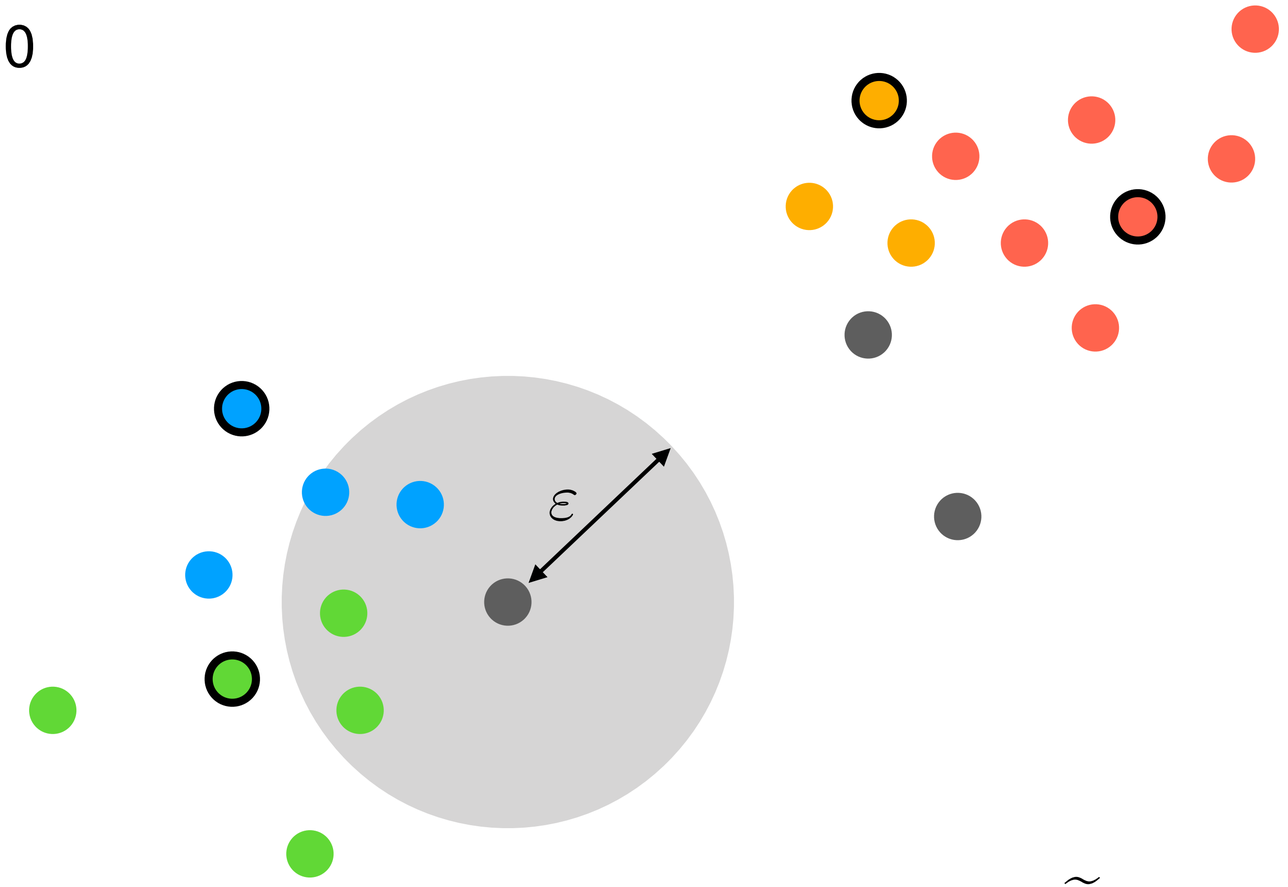


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

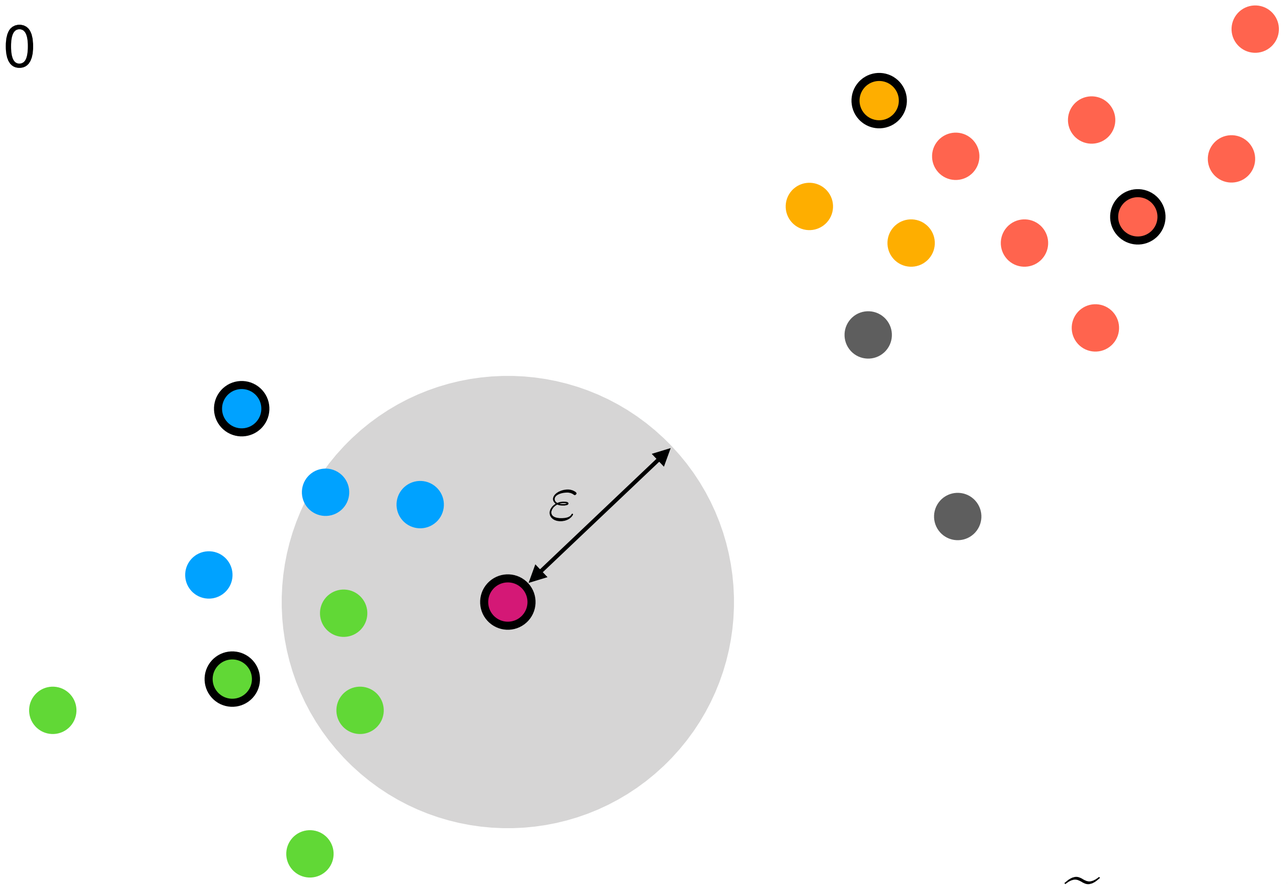


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

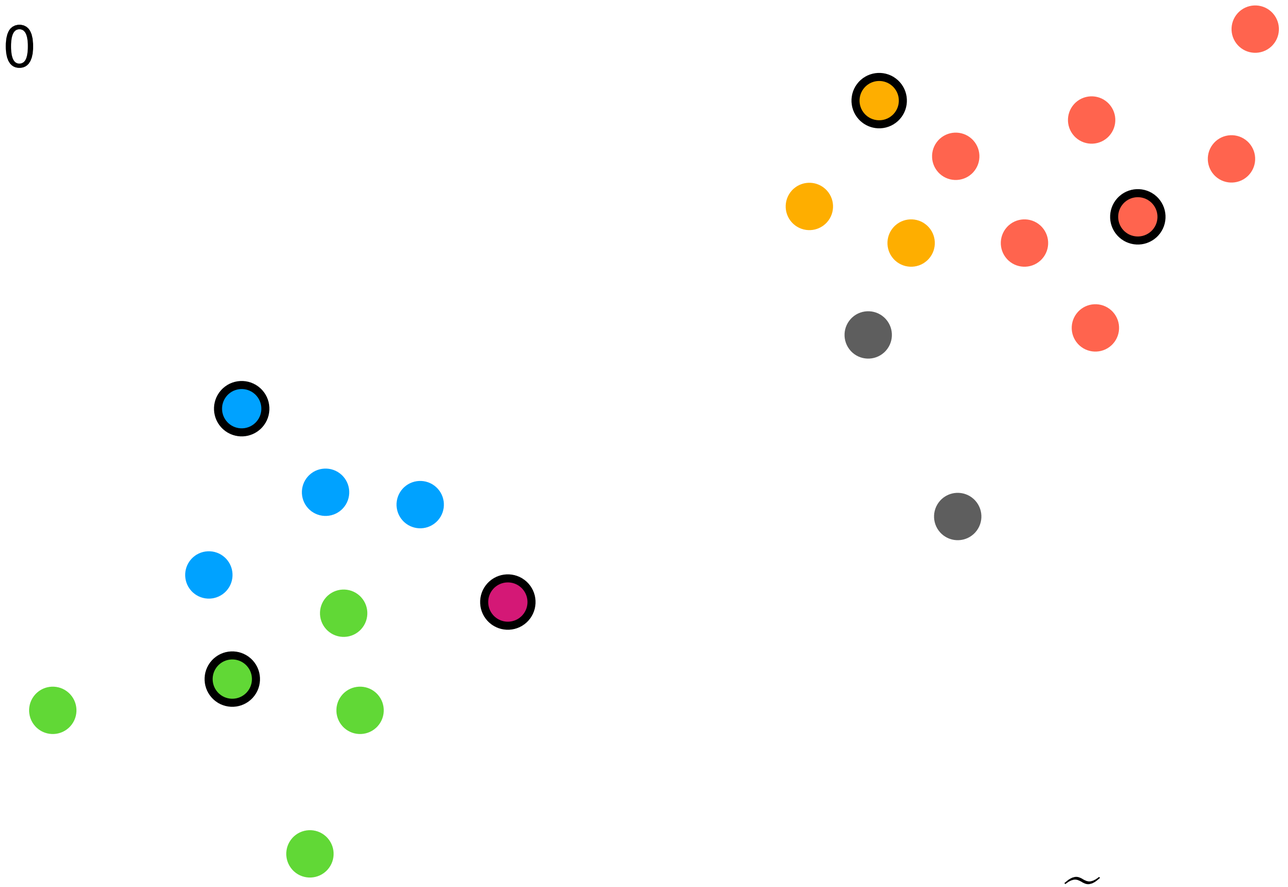


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

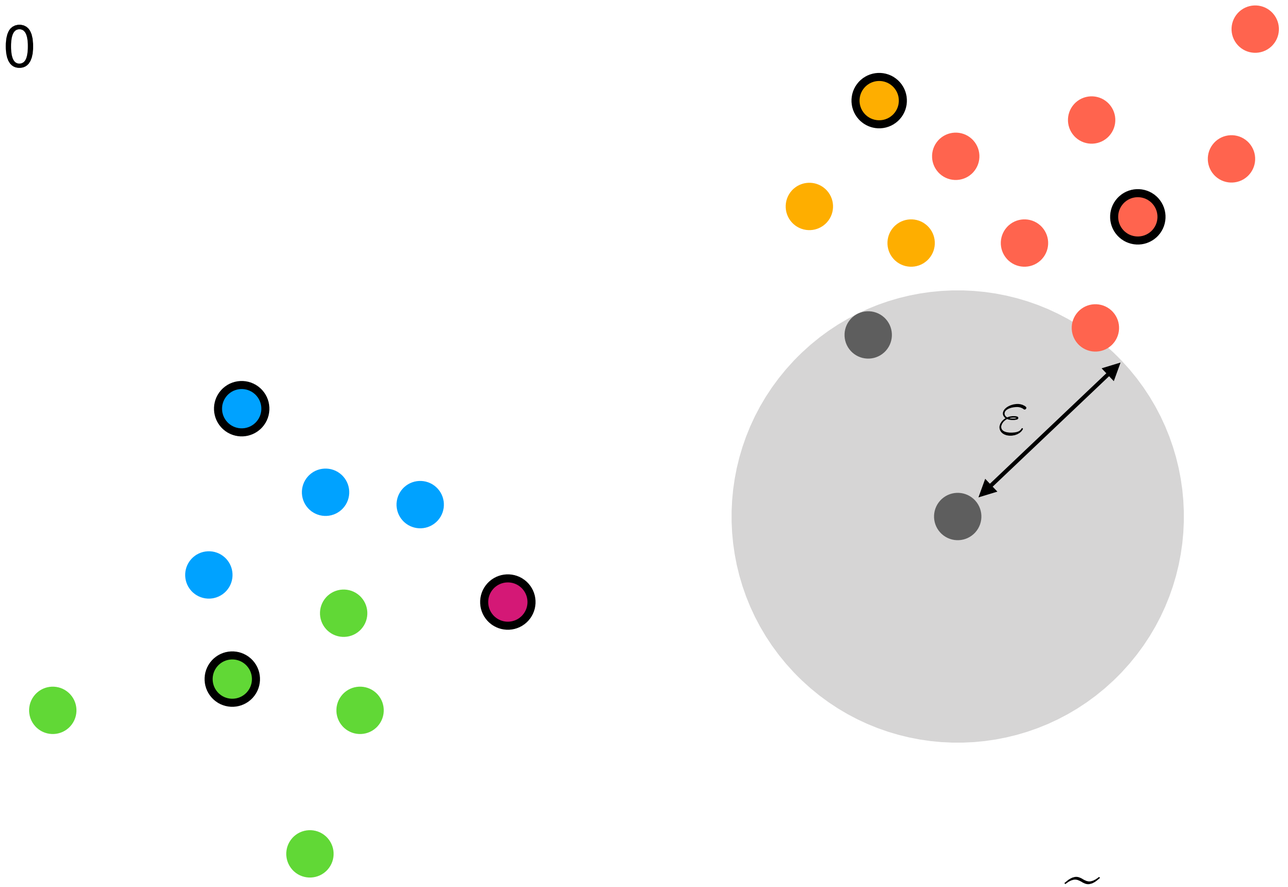


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

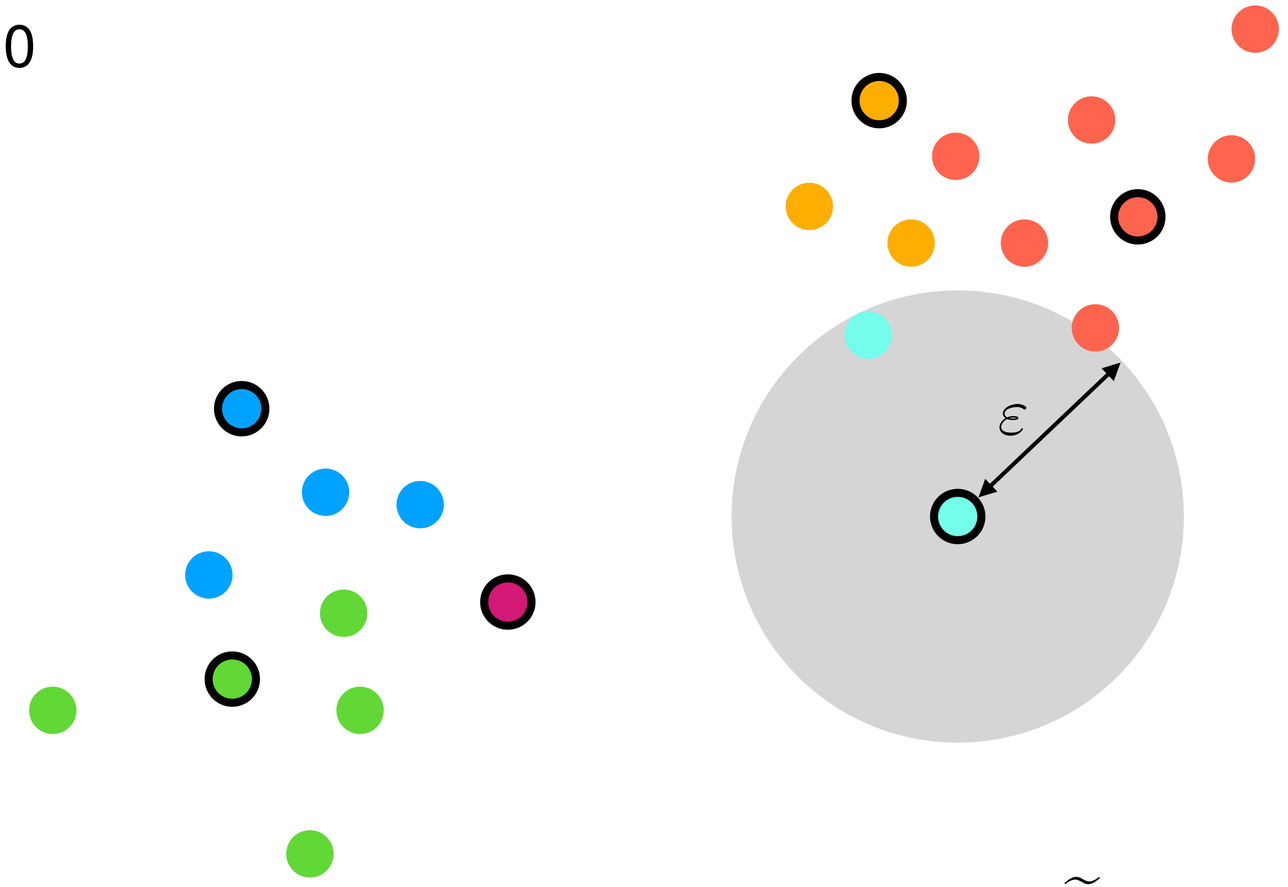


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$

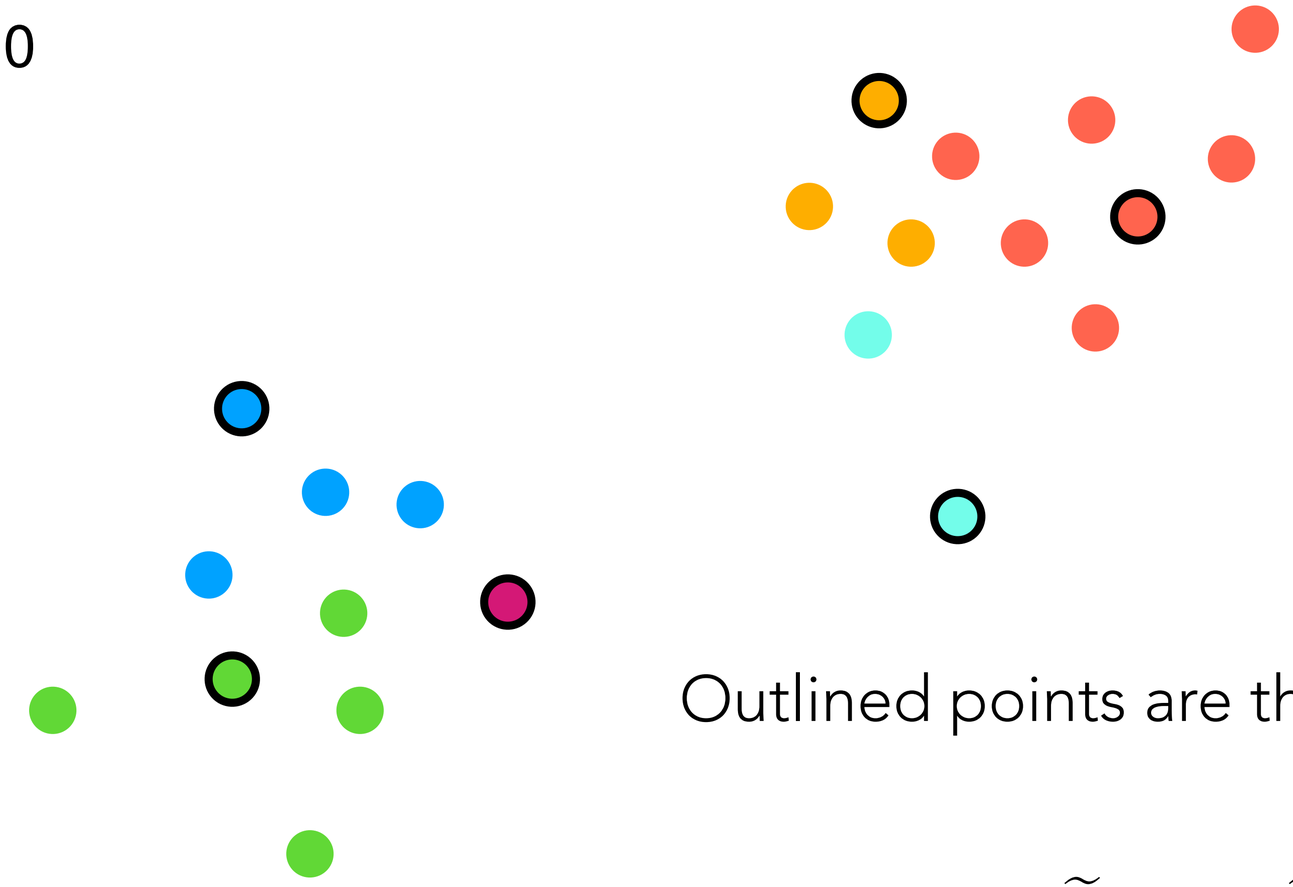


These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$

Clustering with an ε-net

Everything is done in the embedding space (output space of neural net)

Parameter: $\varepsilon > 0$



Outlined points are the *exemplars* $\mathcal{Q}$

These gray points are $\widetilde{X}_1, \ldots, \widetilde{X}_n$