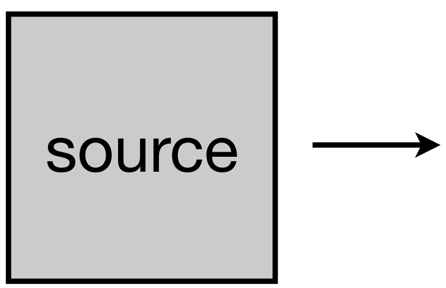
Discriminative Training of Translation Models

MT Marathon - September 7, 2012

Noisy Channels Again

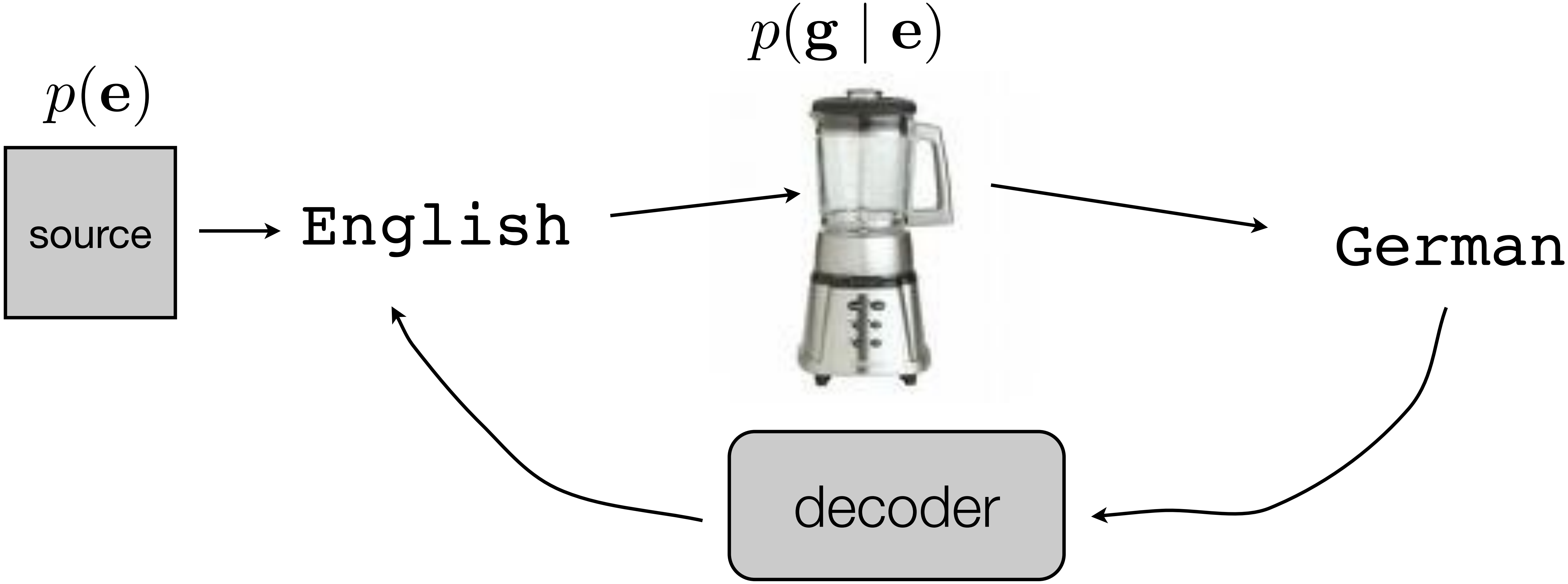
$p(\mathbf{e})$



`English`

Noisy Channels Again

Noisy Channels Again



$$
\begin{aligned}
\mathbf{e}^* &= \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{g}) \\
&= \arg\max_{\mathbf{e}} \frac{p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})}{p(\mathbf{g})} \\
&= \arg\max_{\mathbf{e}} p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})
\end{aligned}
$$

Noisy Channels Again

$$
\begin{aligned}
\mathbf{e}^* &= \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{g}) \\
&= \arg\max_{\mathbf{e}} \frac{p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})}{p(\mathbf{g})} \\
&= \arg\max_{\mathbf{e}} p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})
\end{aligned}
$$

Noisy Channels Again

$$\begin{aligned}
\mathbf{e}^* &= \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{g}) \\
&= \arg\max_{\mathbf{e}} \frac{p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})}{p(\mathbf{g})} \\
&= \arg\max_{\mathbf{e}} p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e}) \\
&= \arg\max_{\mathbf{e}} \log p(\mathbf{g} \mid \mathbf{e}) + \log p(\mathbf{e})
\end{aligned}$$

Noisy Channels Again

$$
\begin{aligned}
\mathbf{e}^* &= \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{g}) \\
&= \arg\max_{\mathbf{e}} \frac{p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})}{p(\mathbf{g})} \\
&= \arg\max_{\mathbf{e}} p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e}) \\
&= \arg\max_{\mathbf{e}} \log p(\mathbf{g} \mid \mathbf{e}) + \log p(\mathbf{e}) \\
&= \arg\max_{\mathbf{e}} \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top}_{\mathbf{w}^\top} \underbrace{\begin{bmatrix} \log p(\mathbf{g} \mid \mathbf{e}) \\ \log p(\mathbf{e}) \end{bmatrix}}_{\mathbf{h}(\mathbf{g},\mathbf{e})}
\end{aligned}
$$

Noisy Channels Again

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} p(\mathbf{e} \mid \mathbf{g})$$

$$= \arg\max_{\mathbf{e}} \frac{p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})}{p(\mathbf{g})}$$

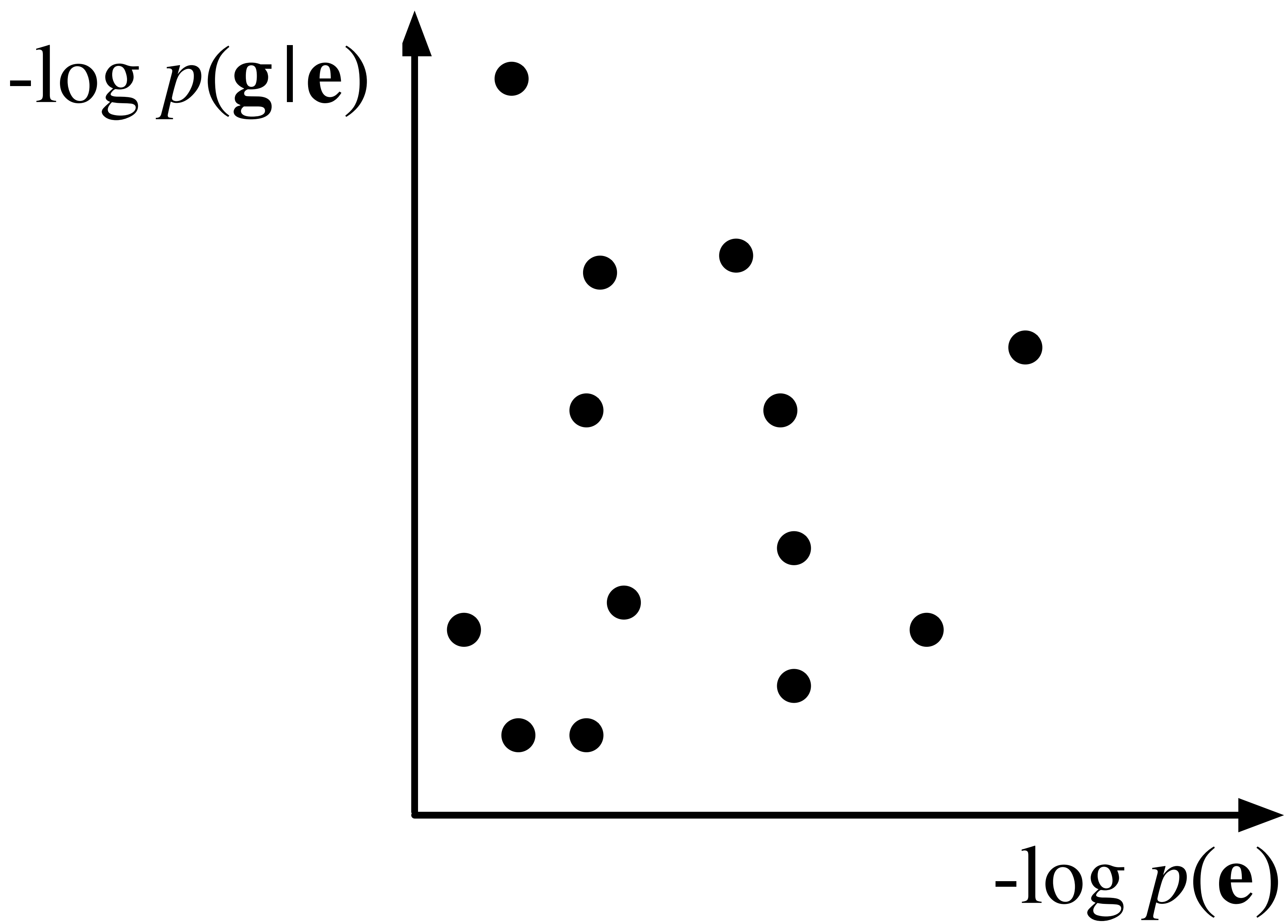
$$= \arg\max_{\mathbf{e}} p(\mathbf{g} \mid \mathbf{e}) \times p(\mathbf{e})$$

This is a linear combinaion

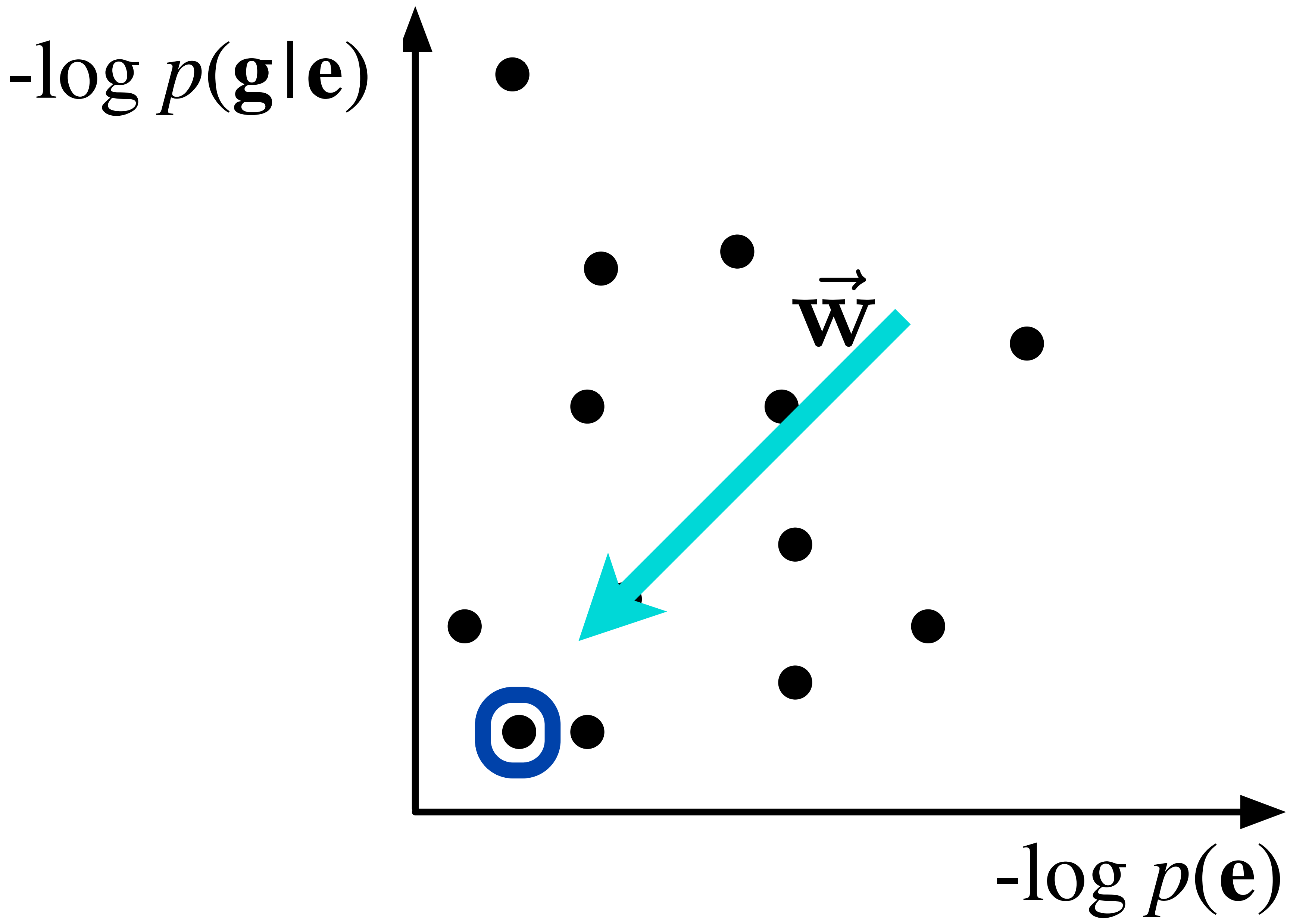
$$= \arg\max_{\mathbf{e}} \log p(\mathbf{g} \mid \mathbf{e}) + \log p(\mathbf{e})$$

$$= \arg\max_{\mathbf{e}} \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}^{\top}}_{\mathbf{w}^{\top}} \underbrace{\begin{bmatrix} \log p(\mathbf{g} \mid \mathbf{e}) \\ \log p(\mathbf{e}) \end{bmatrix}}_{\mathbf{h}(\mathbf{g},\mathbf{e})}$$

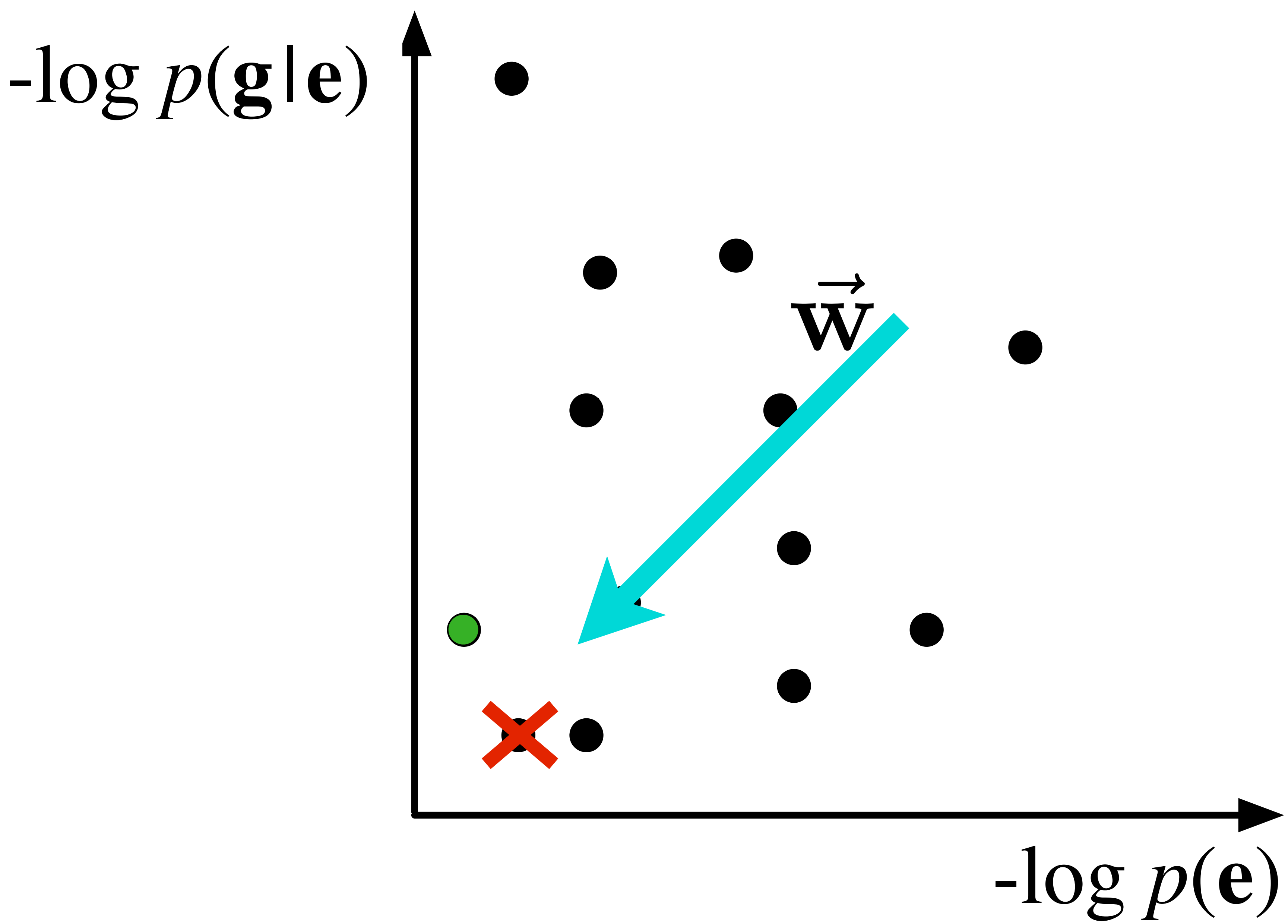
The Noisy Channel



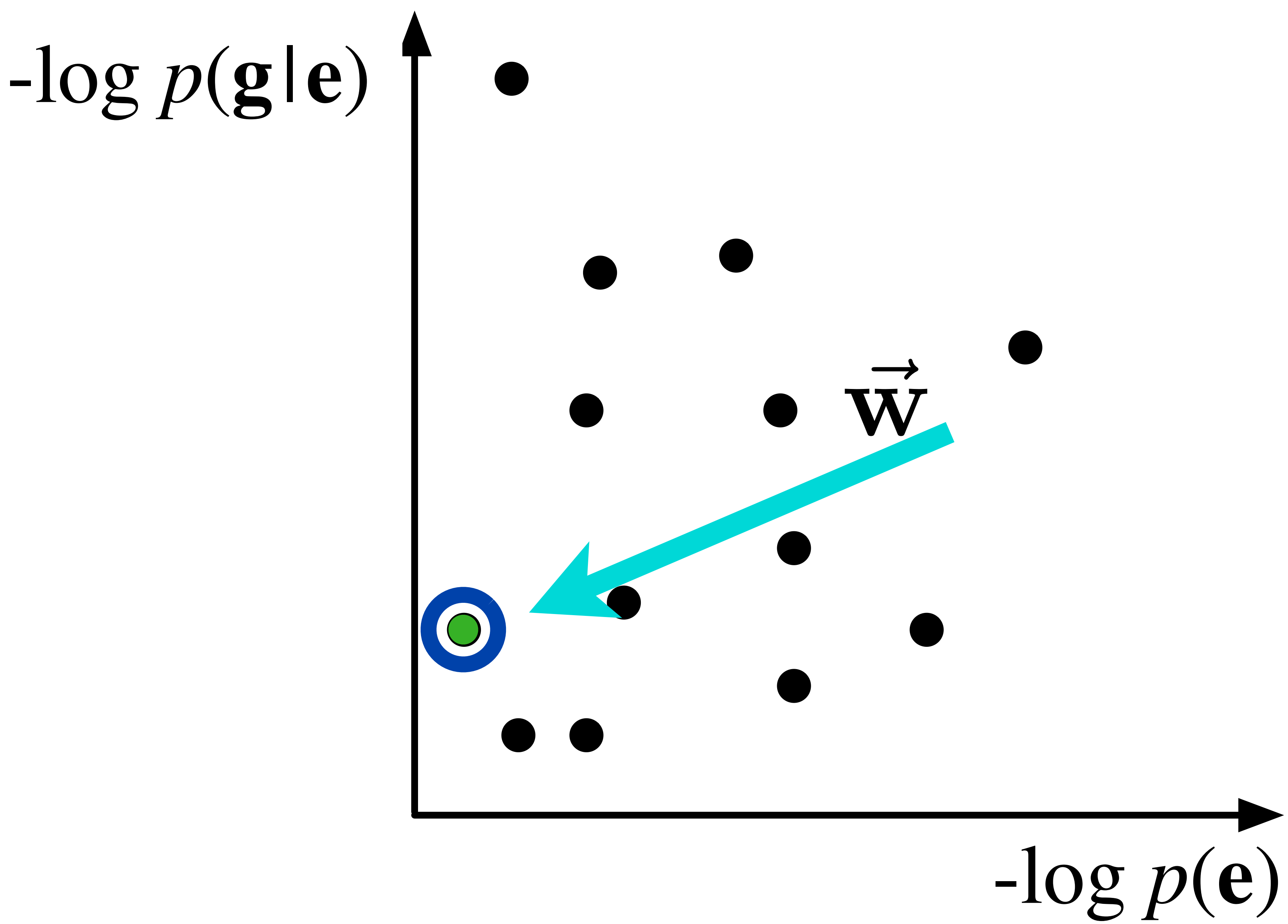
As a Linear Model



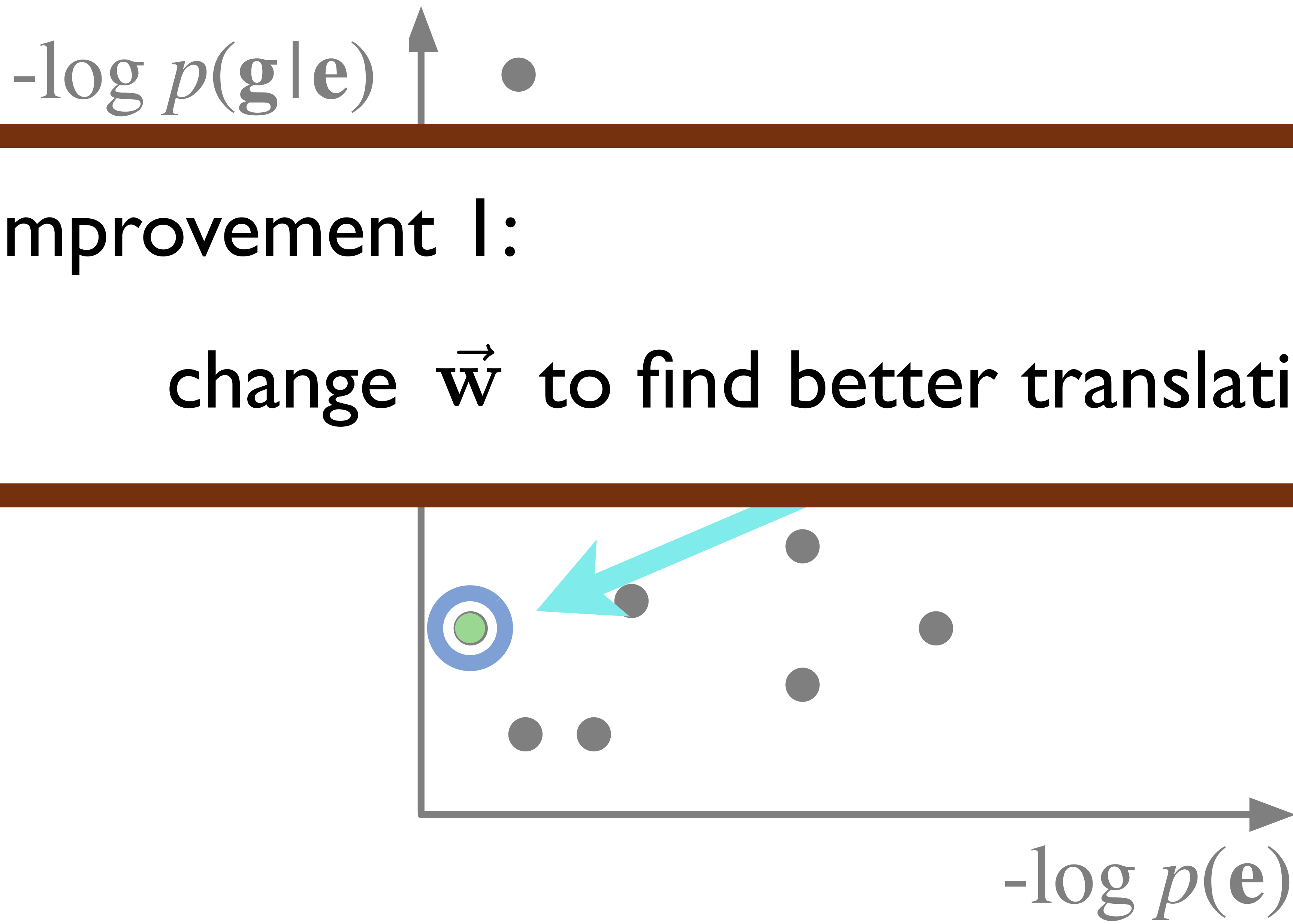
As a Linear Model



As a Linear Model



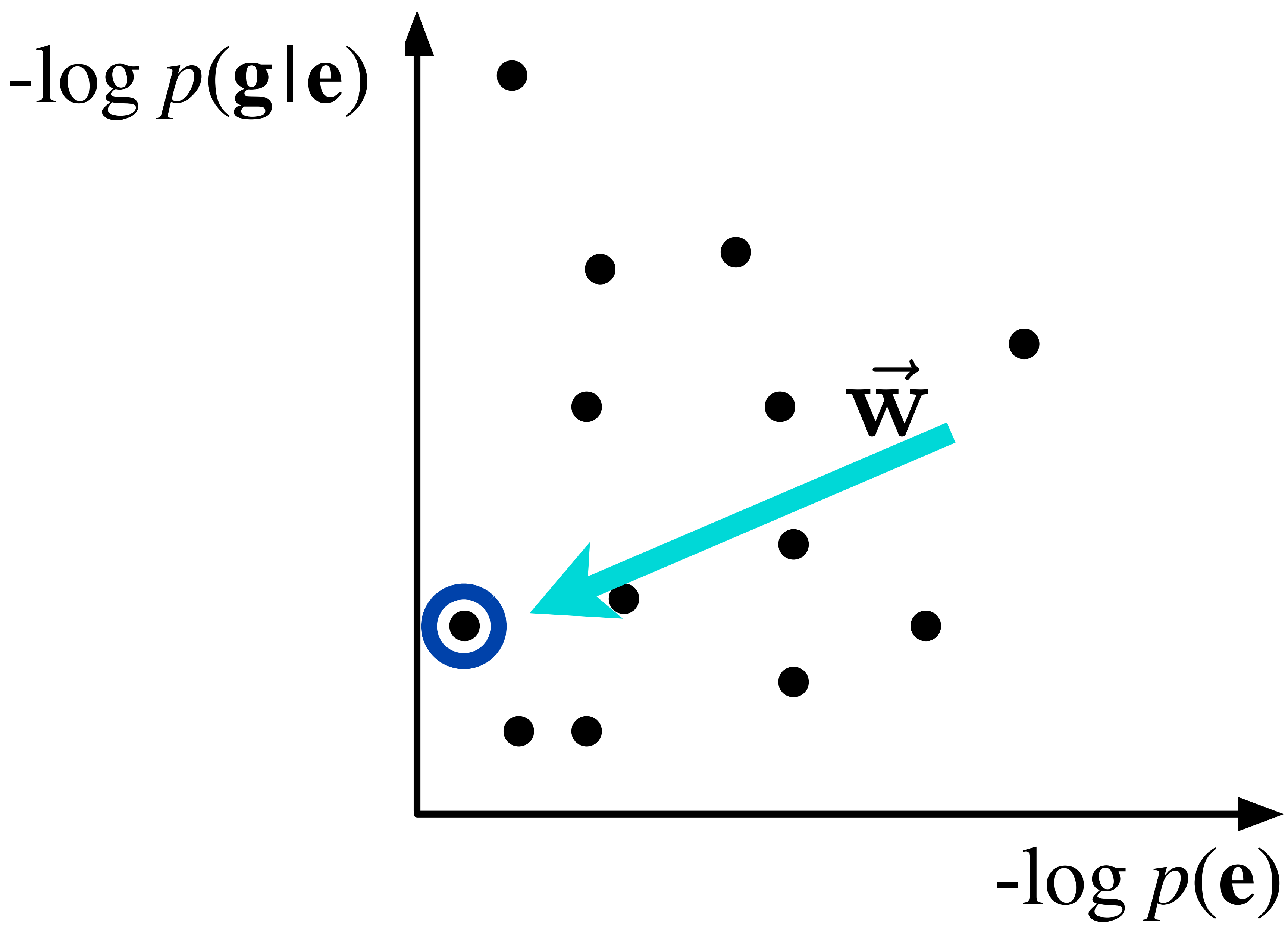
As a Linear Model



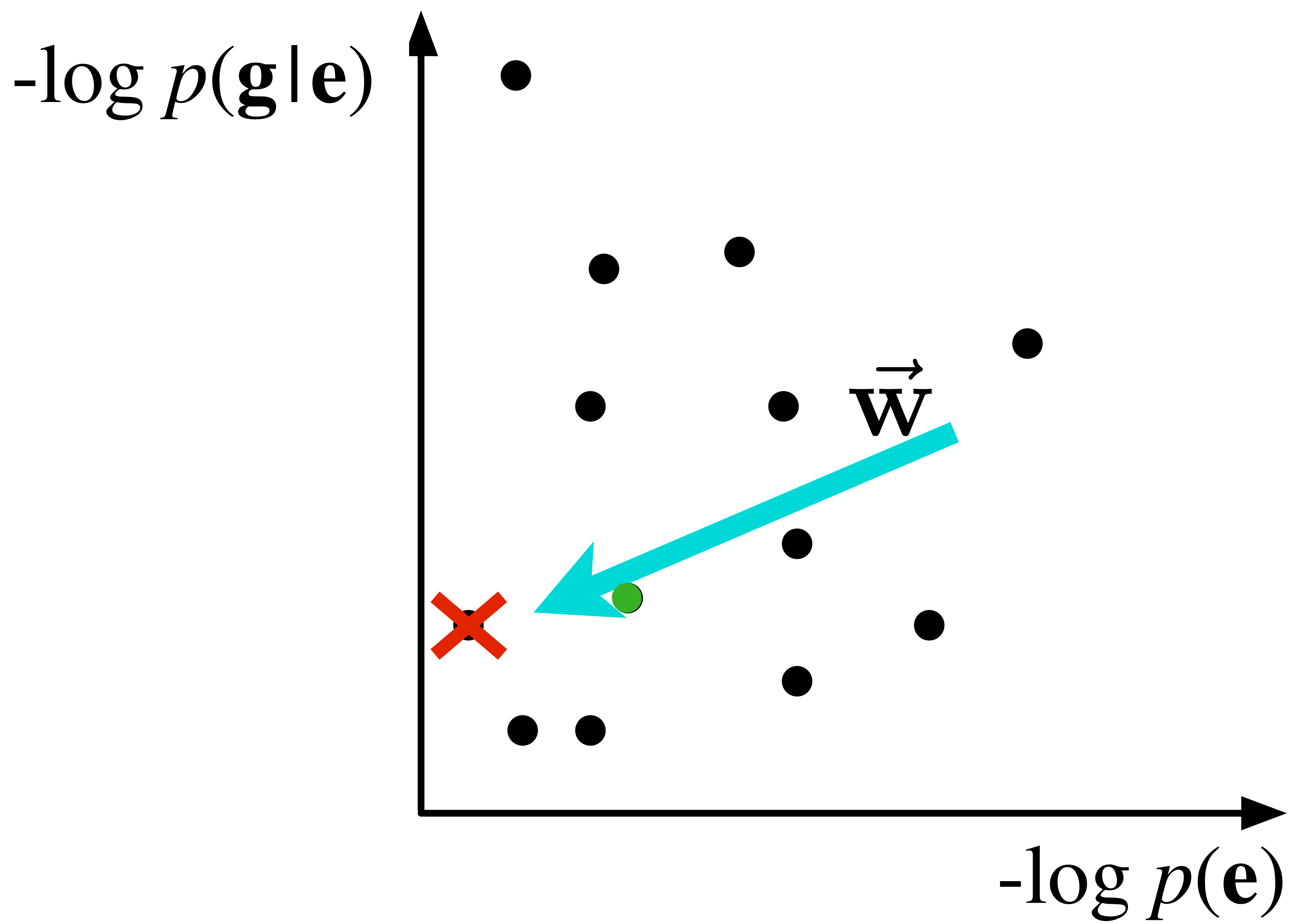
Improvement 1:

change $\vec{w}$ to find better translations

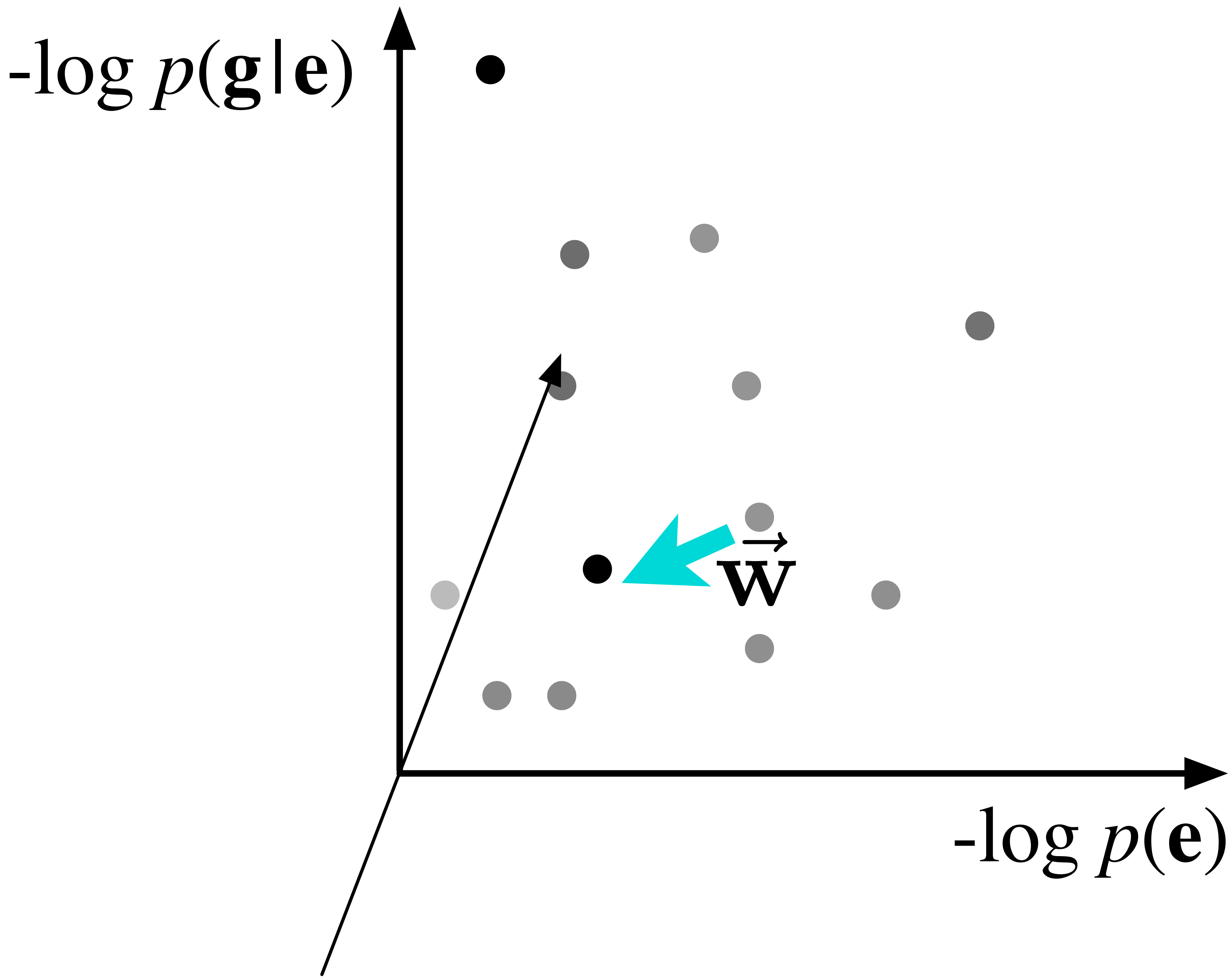
As a Linear Model



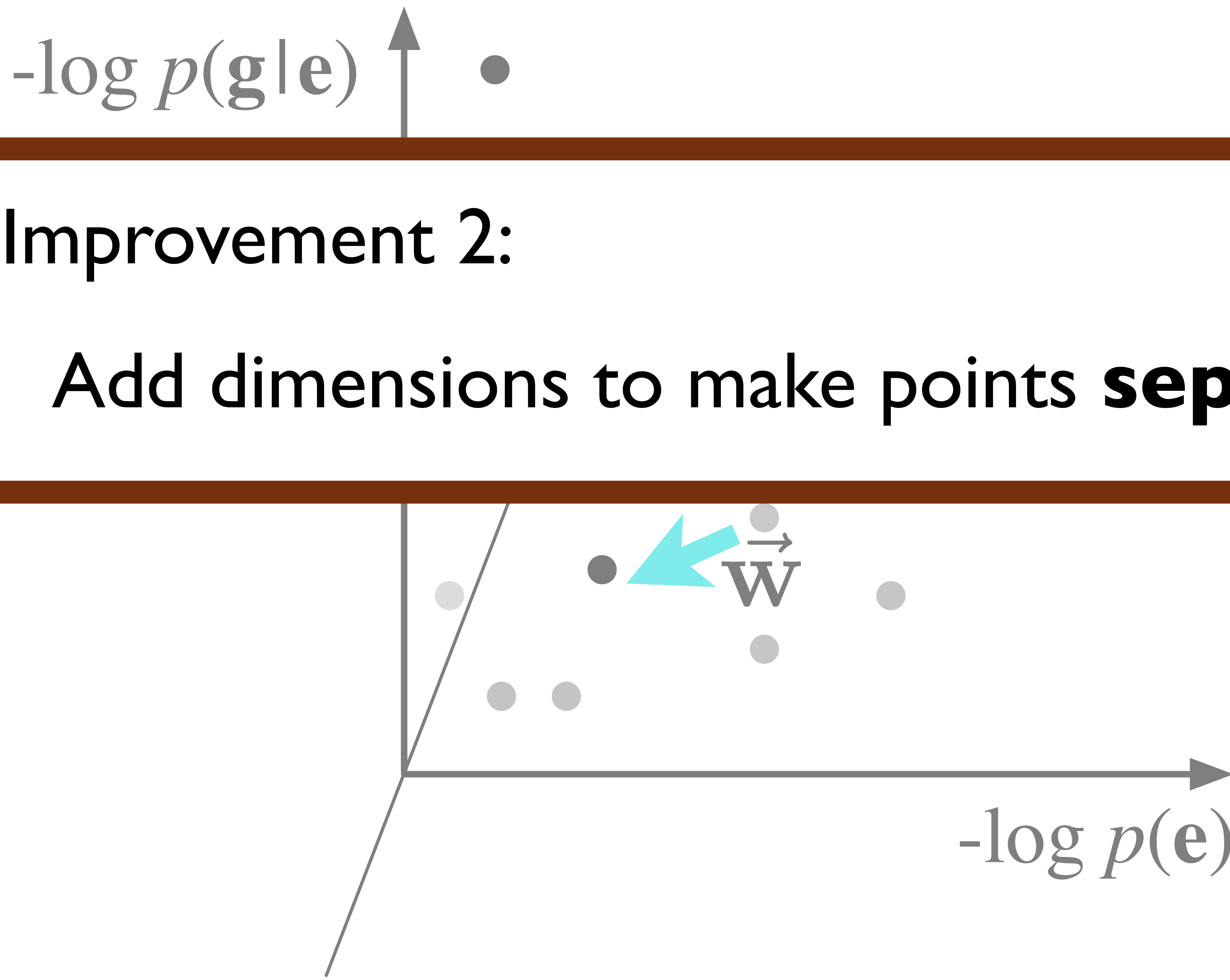
As a Linear Model



As a Linear Model



As a Linear Model



Improvement 2:

Add dimensions to make points **separable**

Linear Models

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e})$$

- Improve the modeling capacity of the noisy channel in two ways
  - Reorient the weight vector
  - Add new dimensions (***new features***)
- Questions
  - What features? $\mathbf{h}(\mathbf{g}, \mathbf{e})$
  - How do we set the weights? $\mathbf{w}$

Mann beißt Hund

Mann

beißt

x BITES y

Hund

Mann beißt Hund

x BITES y

Mann man	beißt bites	Hund cat	Mann man	beißt chase	Hund dog
Mann man	beißt bite	Hund cat	Mann man	beißt bite	Hund dog
Mann dog	beißt bites	Hund man	Mann man	beißt bites	Hund dog

Mann beißt Hund

x BITES y

Mann man	beißt bites	Hund cat	Mann man	beißt chase	Hund dog
Mann man	beißt bite	Hund cat	Mann man	beißt bite	Hund dog
Mann dog	beißt bites	Hund man	Mann man	beißt bites	Hund dog

Mann beißt Hund

x BITES *y*

Mann	beißt	Hund
man	bites	cat
man	bite	cat
dog	bites	man

Mann	beißt	Hund
man	chase	dog
man	bite	dog
man	bites	dog

Mann beißt Hund

x BITES y

Mann man	beißt bites	Hund cat		Mann man	beißt chase	Hund dog
Mann man	beißt bite	Hund cat		Mann man	beißt bite	Hund dog
Mann dog	beißt bites	Hund man		Mann man	beißt bites	Hund dog

Mann beißt Hund

x BITES y

Mann	beißt	Hund
man	bites	cat
man	bite	cat
dog	bites	man

Mann	beißt	Hund
man	chase	dog
man	bite	dog
man	bites	dog

Feature Classes

Lexical

Are lexical choices appropriate?

bank = "River bank" vs. "Financial institution"

Feature Classes

Lexical

Are lexical choices appropriate?

bank = "River bank" vs. "Financial institution"

Configurational

Are semantic/syntactic relations preserved?

"Dog bites man" vs. "Man bites dog"

Feature Classes

Lexical

Are lexical choices appropriate?

bank = "River bank" vs. "Financial institution"

Configurational

Are semantic/syntactic relations preserved?

"Dog bites man" vs. "Man bites dog"

Grammatical

Is the output fluent / well-formed?

"Man *bites* dog" vs. "Man *bite* dog"

What do lexical features look like?

Mann	beißt	Hund
man	bites	cat

What do lexical features look like?

Mann	beißt	Hund
man	bites	cat

What do lexical features look like?

Mann	beißt	Hund
man	bites	cat

First attempt:

$$score(\mathbf{g}, \mathbf{e}) = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e})$$

$$h_{15,342}(\mathbf{g}, \mathbf{e}) = \begin{cases} 1, & \exists i, j : g_i = Hund, e_j = cat \\ 0, & \text{otherwise} \end{cases}$$

What do lexical features look like?

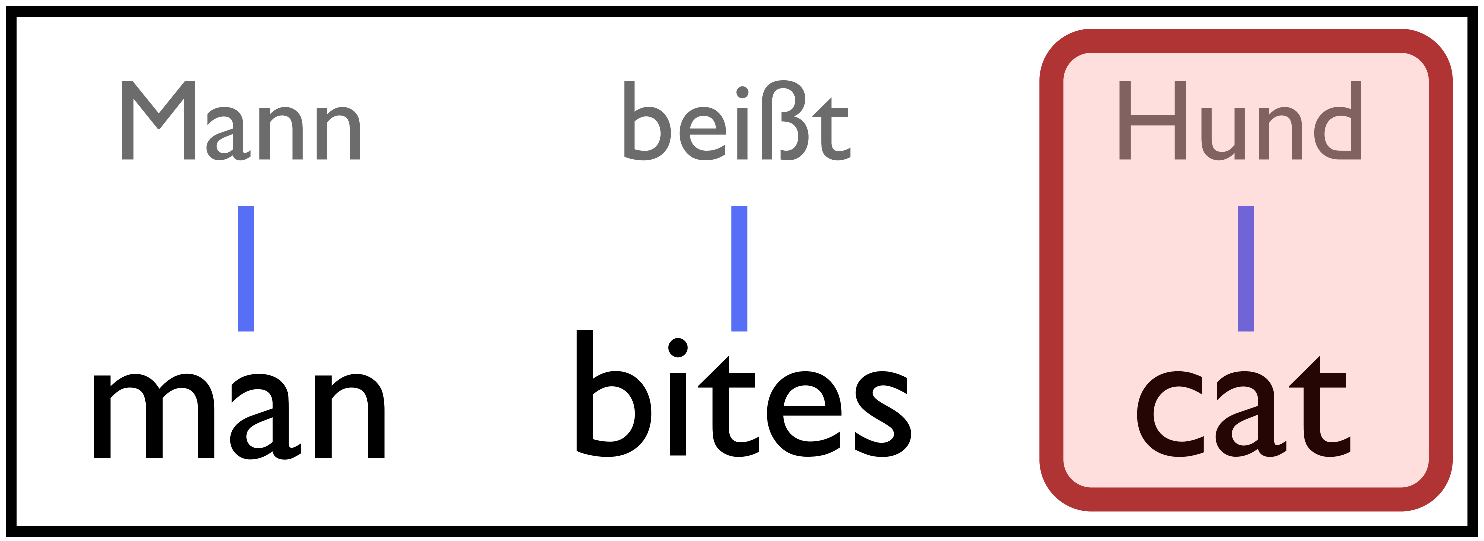
First attempt:

$$score(\mathbf{g}, \mathbf{e}) = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e})$$

$$h_{15,342}(\mathbf{g}, \mathbf{e}) = \begin{cases} 1, & \exists i, j : g_i = Hund, e_j = cat \\ 0, & \text{otherwise} \end{cases}$$

*But what if a **cat** is being chased by a **Hund**?*

What do lexical features look like?



Latent variables enable more precise features:

$$score(\mathbf{g}, \mathbf{e}, \mathbf{a}) = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$$

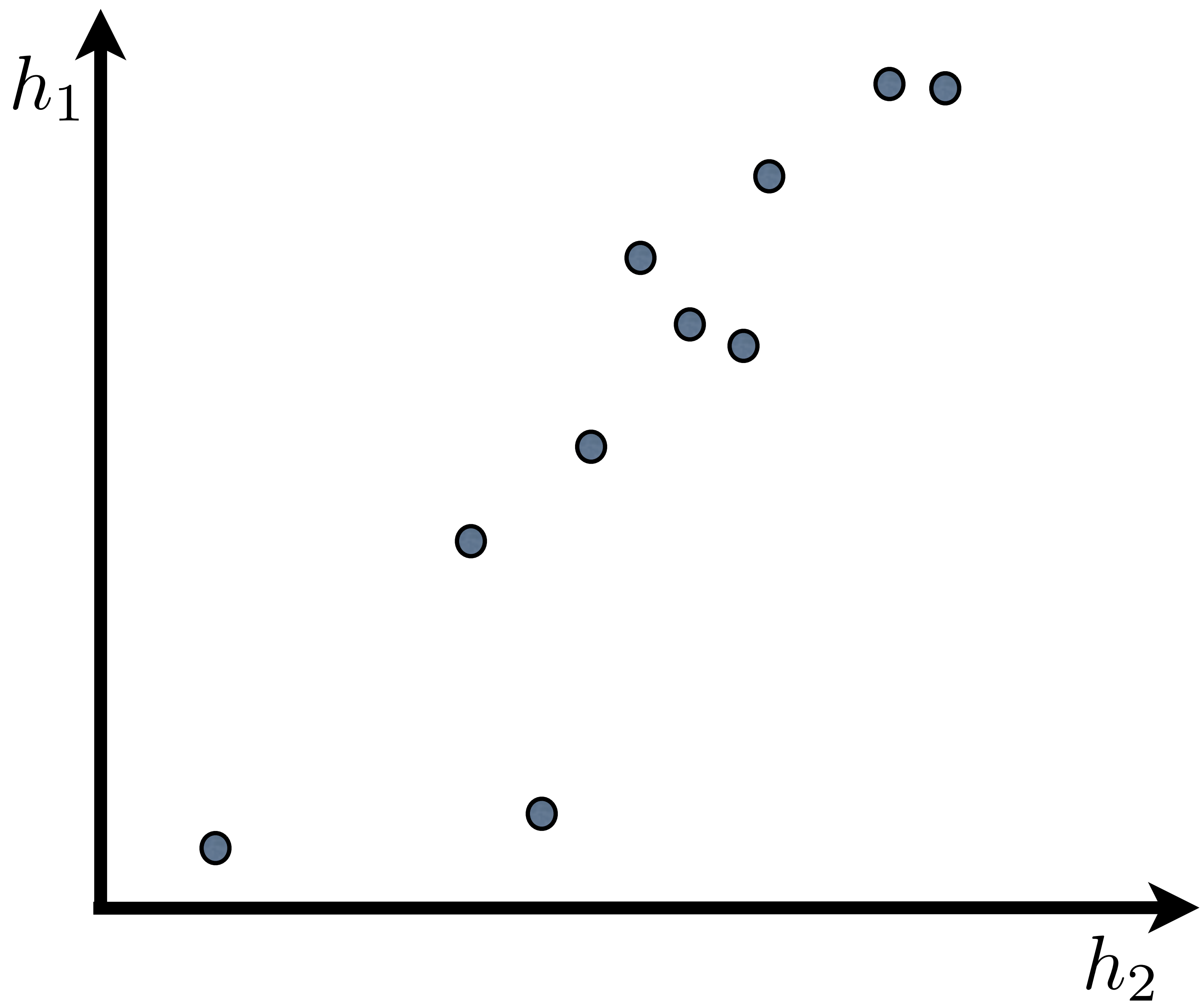
$$h_{15,342}(\mathbf{g}, \mathbf{e}, \mathbf{a}) = \sum_{(i,j) \in \mathbf{a}} \begin{cases} 1, & \text{if } g_i = Hund, e_j = cat \\ 0, & \text{otherwise} \end{cases}$$

Standard Features

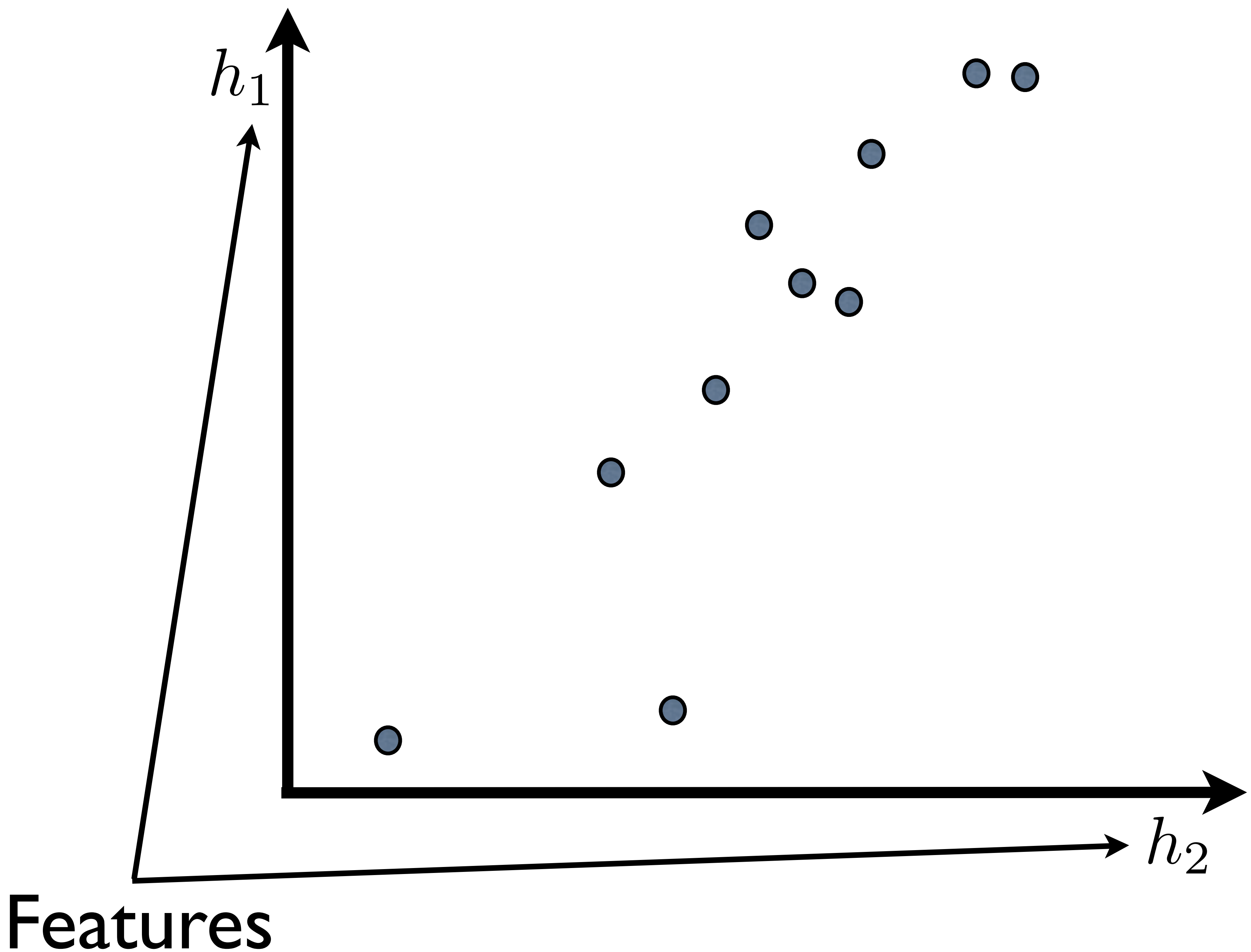
- **Target side features**
  - log p(e) [*n*-gram language model]
  - Number of words in hypothesis
- **Source + target features**
  - log relative frequency e|*f* of each rule [log #(e,f) - log #(f)]
  - log relative frequency *f*|e of each rule [log #(e,f) - log #(e)]
  - "lexical translation" log probability e|*f* of each rule [$\approx \log p_{model1}(e|f)$]
  - "lexical translation" log probability *f*|e of each rule [$\approx \log p_{model1}(f|e)$]
- **Other features**
  - Count of rules/phrases used
  - Reordering pattern probabilities

Parameter Learning

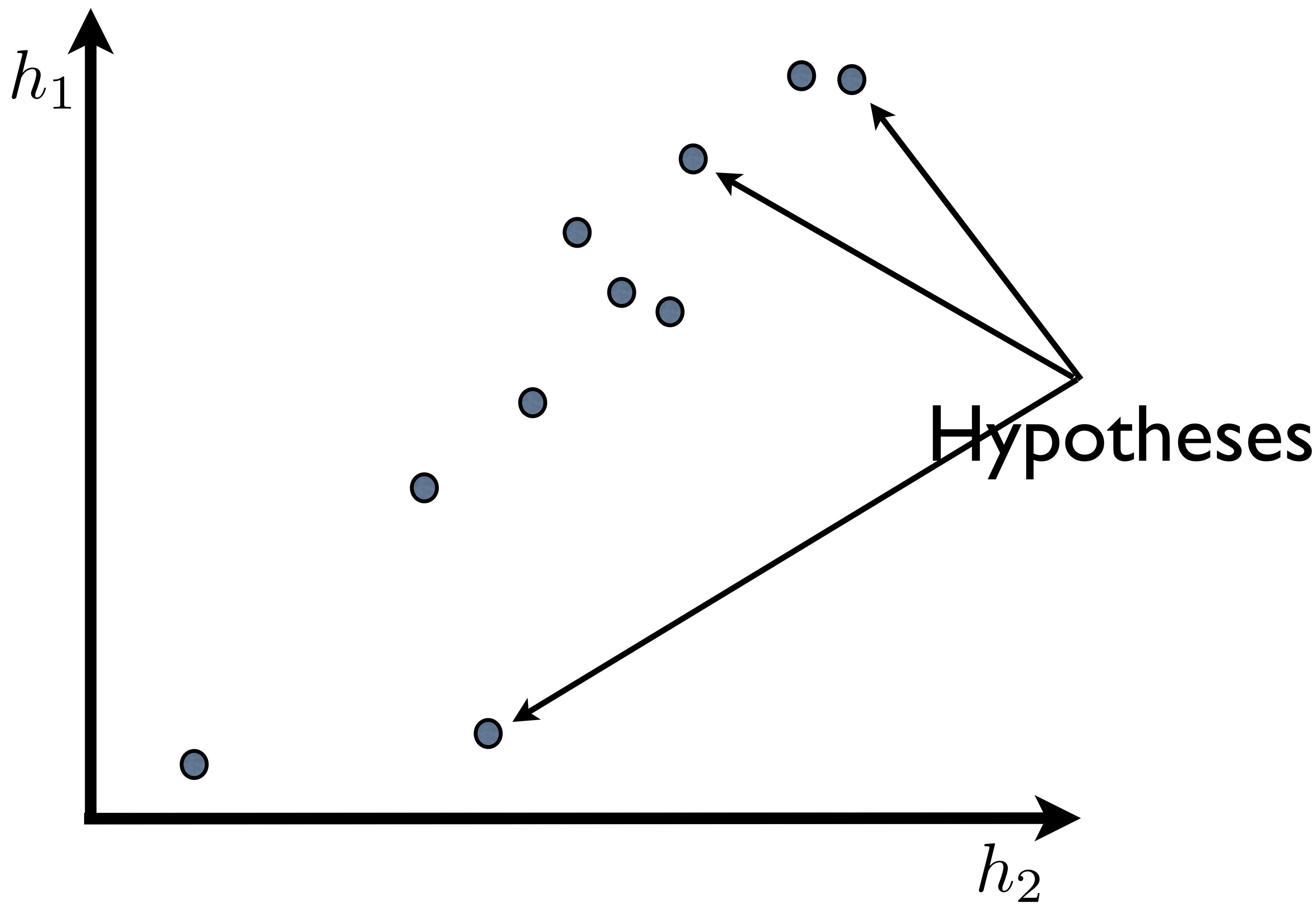
Hypothesis Space



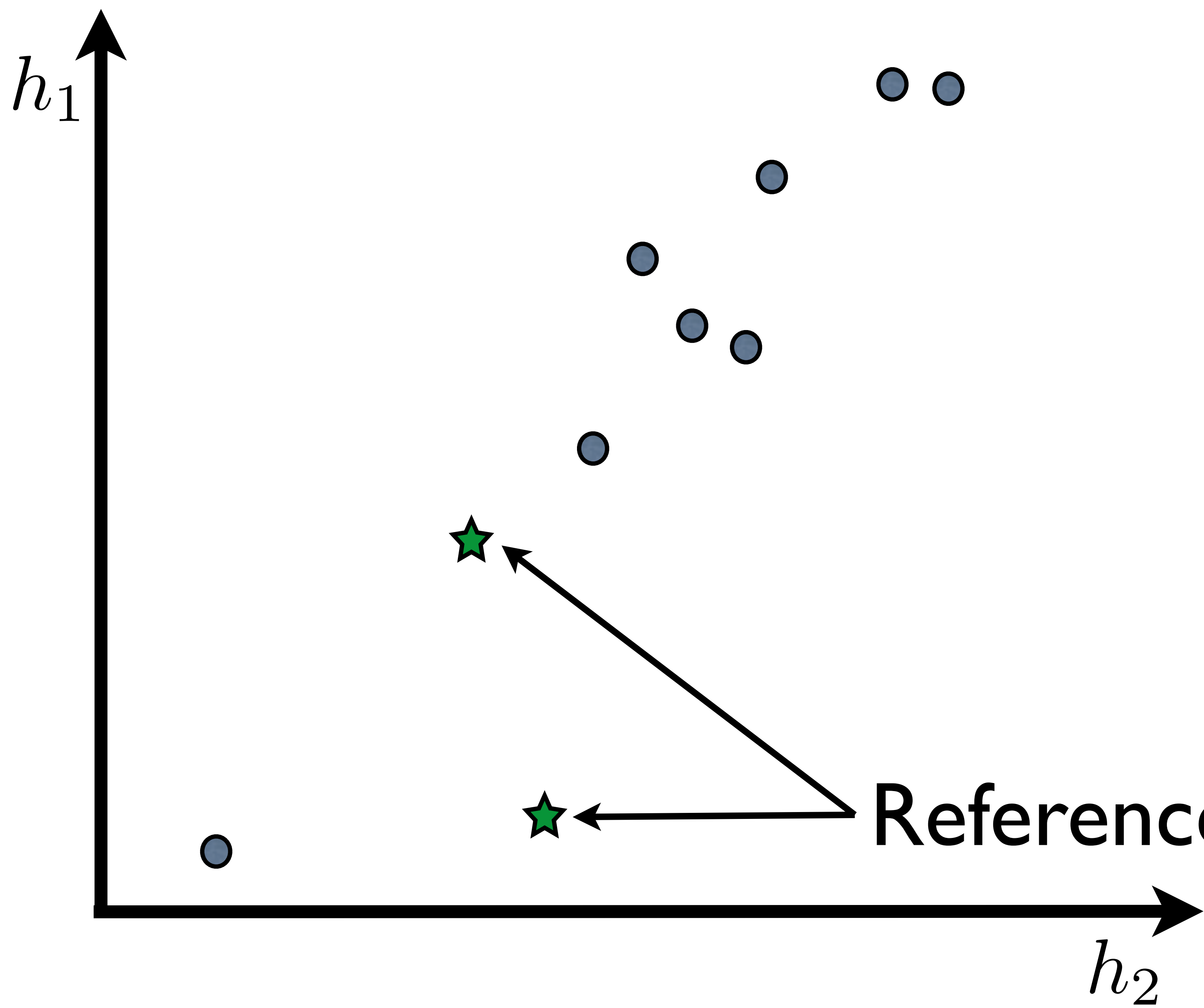
Hypothesis Space



Hypothesis Space



Hypothesis Space



Preliminaries

We assume a **decoder** that computes:

$$\langle \mathbf{e}^*, \mathbf{a}^* \rangle = \arg\max_{\langle \mathbf{e}, \mathbf{a} \rangle} \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$$

And ***K*-best lists** of, that is:

$$\{\langle \mathbf{e}_i^*, \mathbf{a}_i^* \rangle\}_{i=1}^{i=K} = \arg i^{\text{th}}\text{-}\max_{\langle \mathbf{e}, \mathbf{a} \rangle} \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$$

Standard, efficient algorithms exist for this.

Learning Weights

- Try to match the reference translation ***exactly***
  - **Conditional random field**
    - Maximize the conditional probability of the reference translations
    - "Average" over the different latent variables

Learning Weights

- Try to match the reference translation ***exactly***
  - **Conditional random field**
    - Maximize the conditional probability of the reference translations
    - "Average" over the different latent variables
  - **Max-margin**
    - Find the weight vector that separates the reference translation from others by the maximal margin
    - Maximal setting of the latent variables

Problems

- These methods give "full credit" when the model ***exactly*** produces the reference and no credit otherwise
- **What is the problem with this?**

Problems

- These methods give "full credit" when the model ***exactly*** produces the reference and no credit otherwise
- **What is the problem with this?**
  - There are many ways to translate a sentence
  - What if we have multiple reference translations?
  - **What about partial credit?**

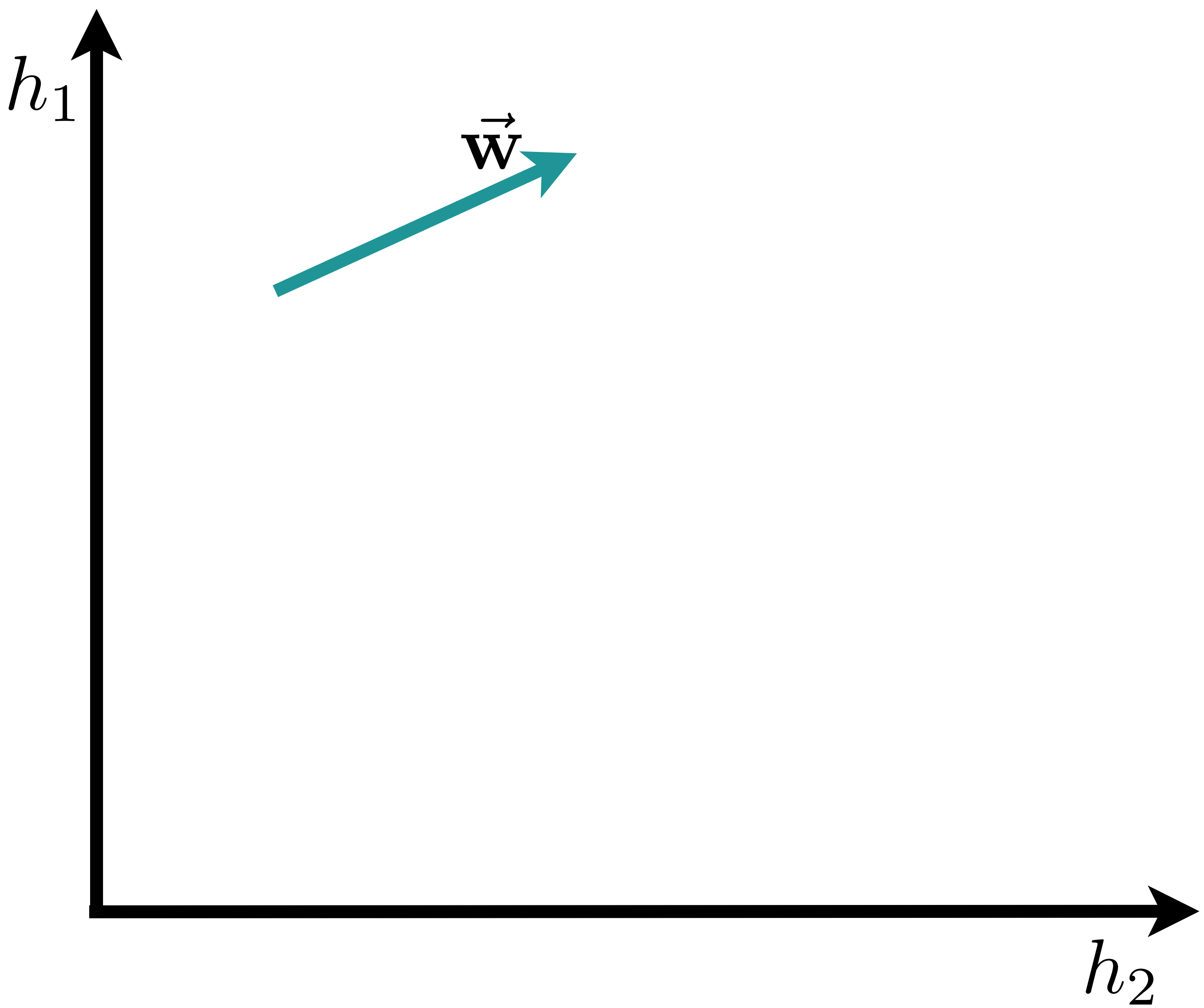
Cost-Sensitive Training

- Assume we have a **cost function** that gives a score for how good/bad a translation is

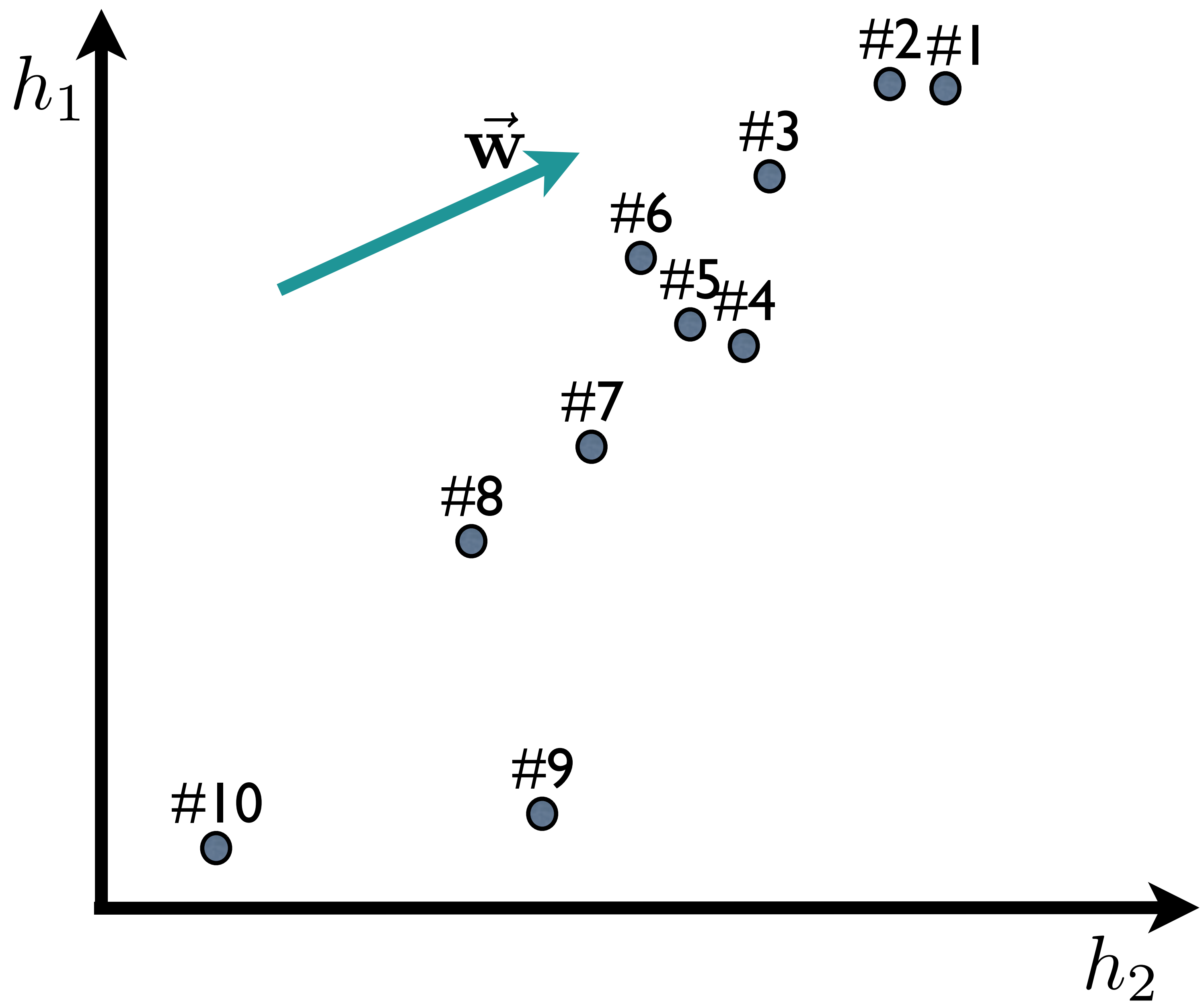
$$\ell(\hat{\mathrm{e}}, \mathcal{E}) \mapsto [0, 1]$$

- Optimize the weight vector by making reference to this function
  - We will talk about two ways to do this

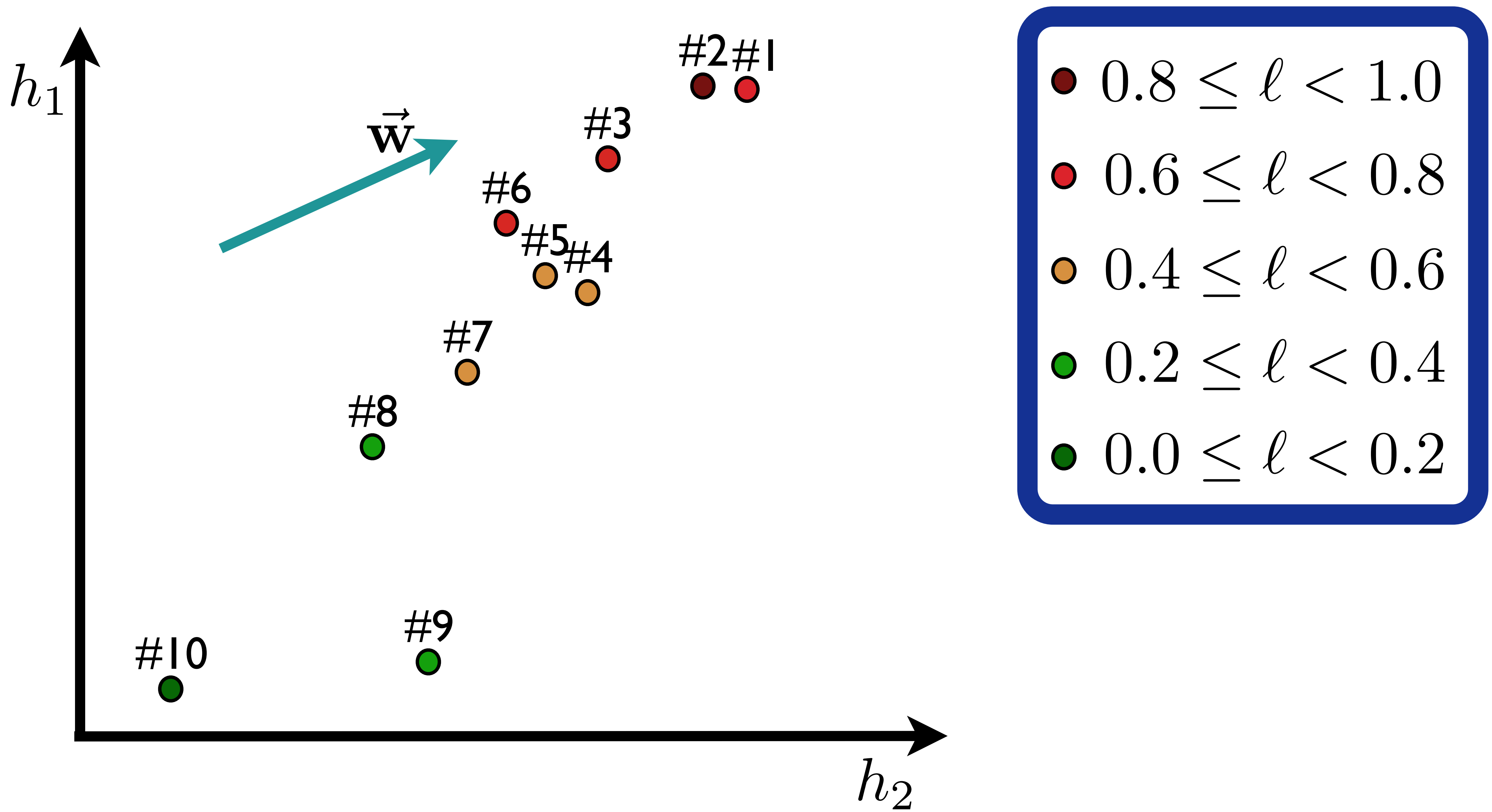
K-Best List Example



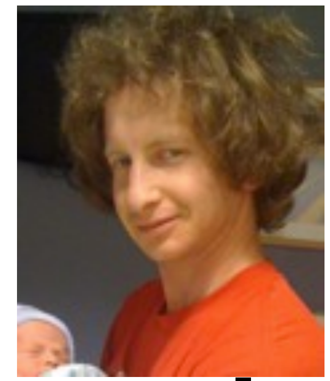
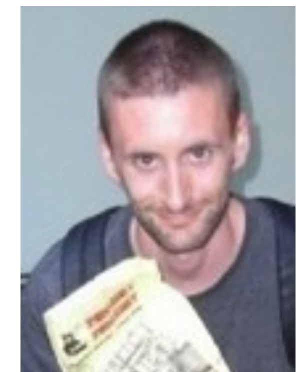
K-Best List Example



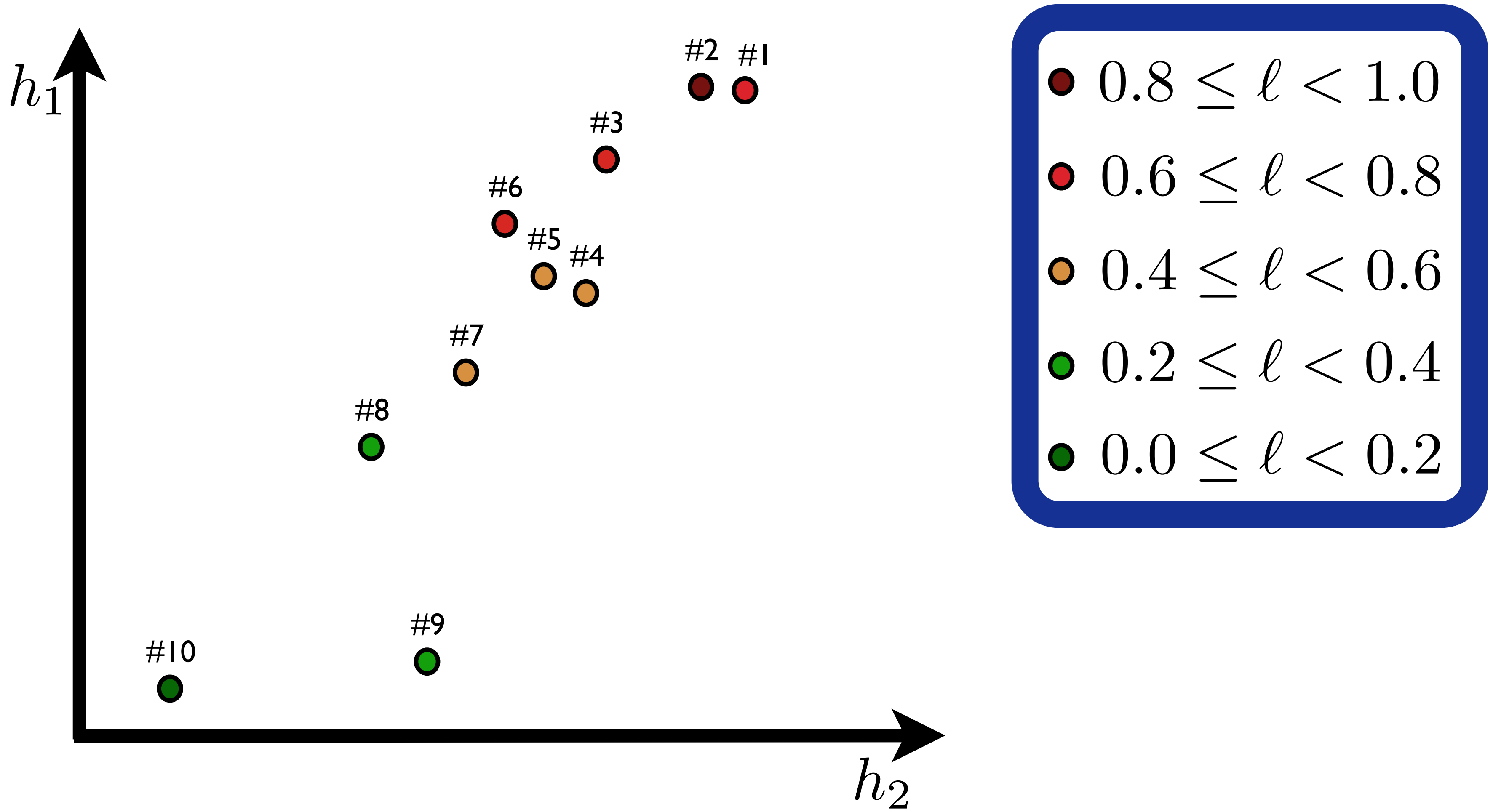
K-Best List Example



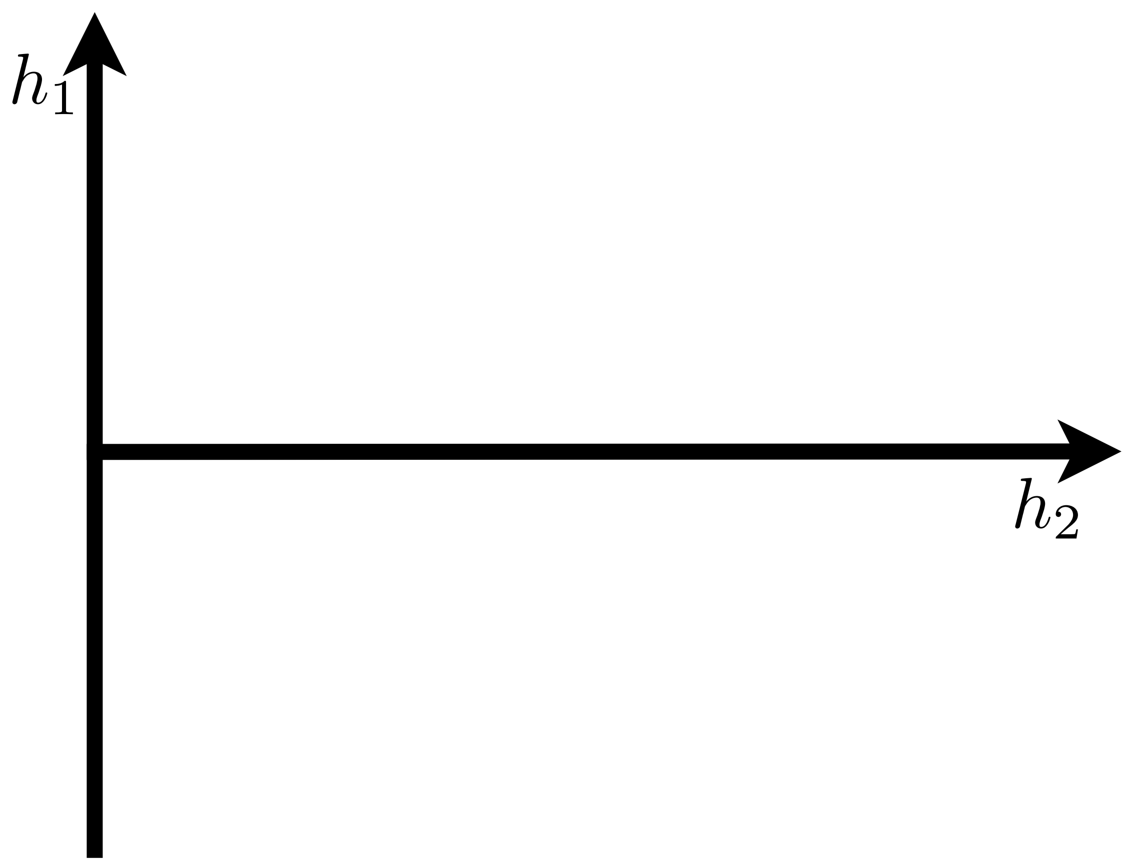
Training as Classification

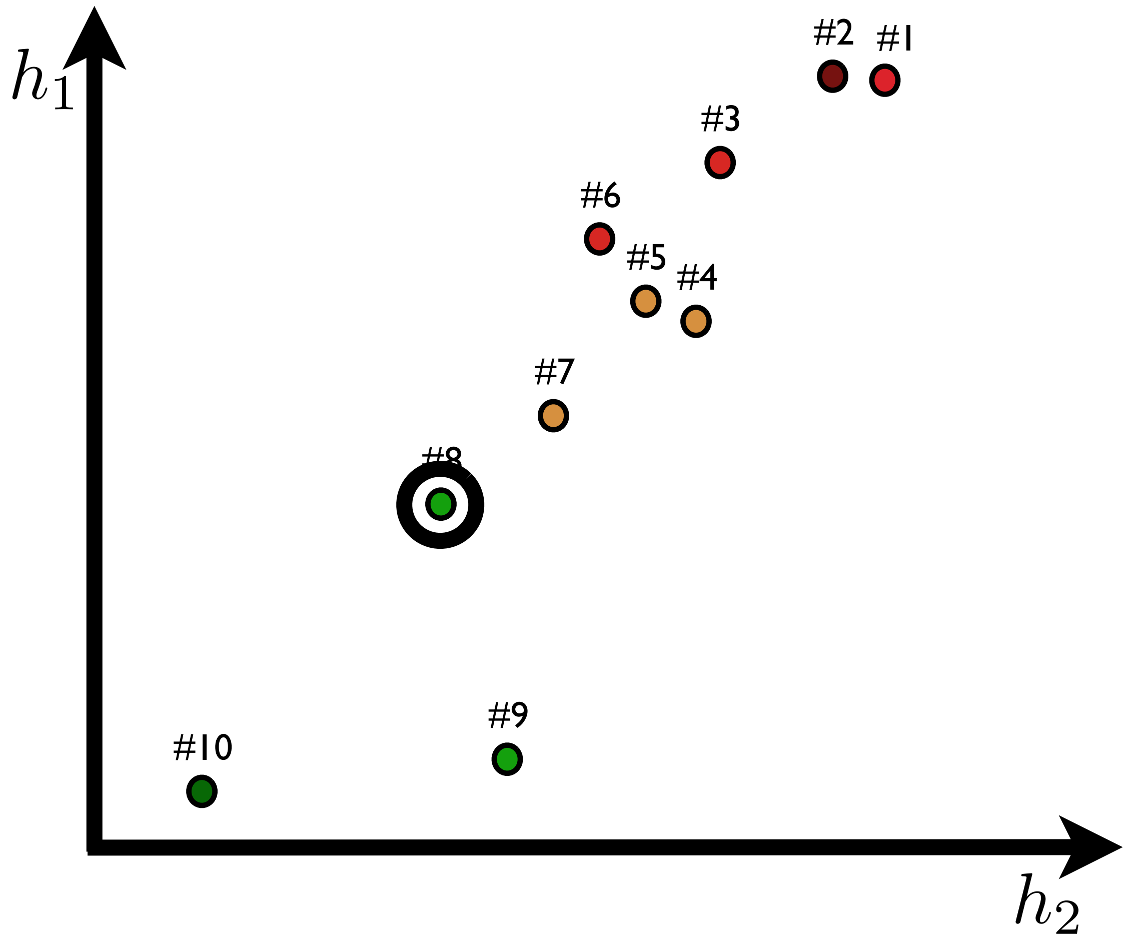
- **P**airwise **R**anking **O**ptimization
  - Reduce training problem to **binary classification** with a **linear model**
- **Algorithm**
  - For *i*=1 to *N*
    - Pick random pair of hypotheses (A,B) from *K*-best list
    - Use cost function to determine if is A or B better
    - Create *i*th training instance
  - Train binary linear classifier



h_1
#2
#1
#3
#6
#5
#4
#7
#8
#9
#10
h_2
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$



h_1
h_2



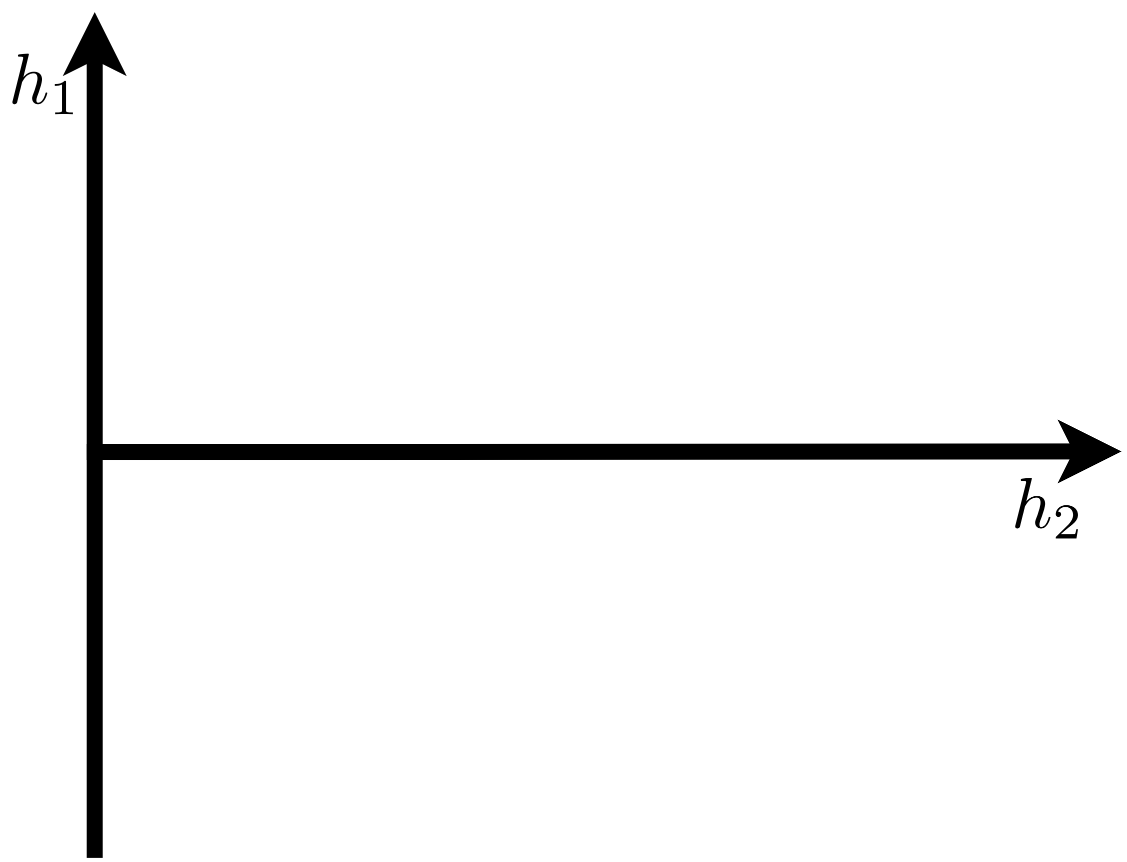
$0.8 \leq \ell < 1.0$

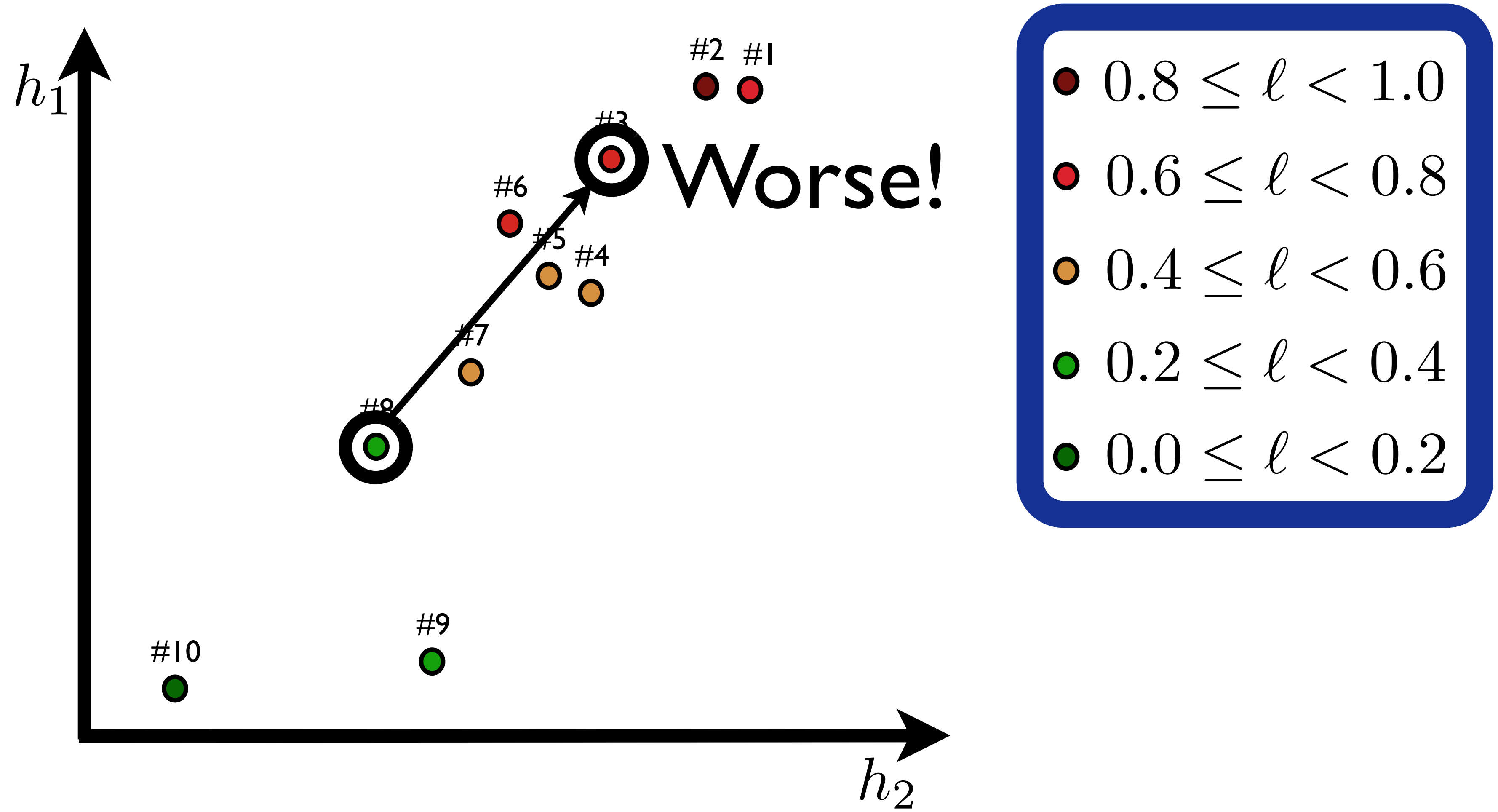
$0.6 \leq \ell < 0.8$

$0.4 \leq \ell < 0.6$

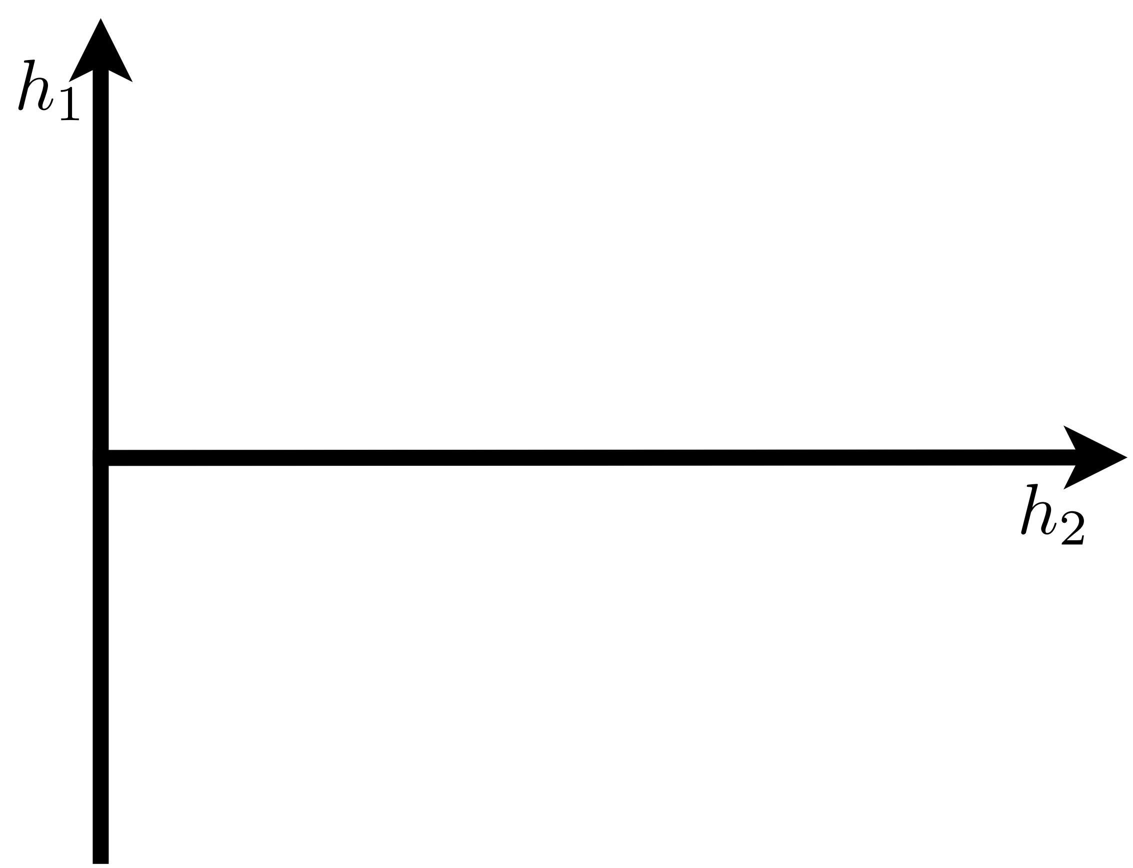
$0.2 \leq \ell < 0.4$

$0.0 \leq \ell < 0.2$

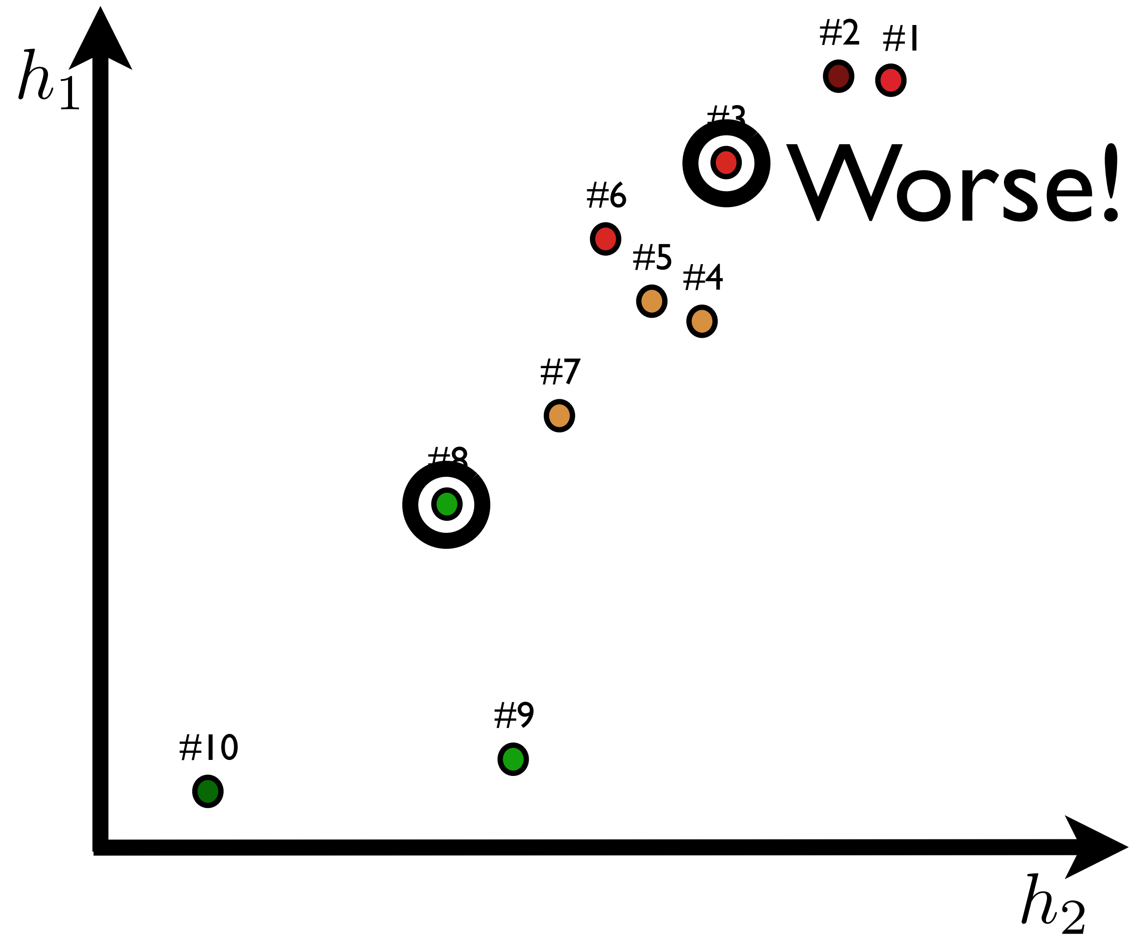




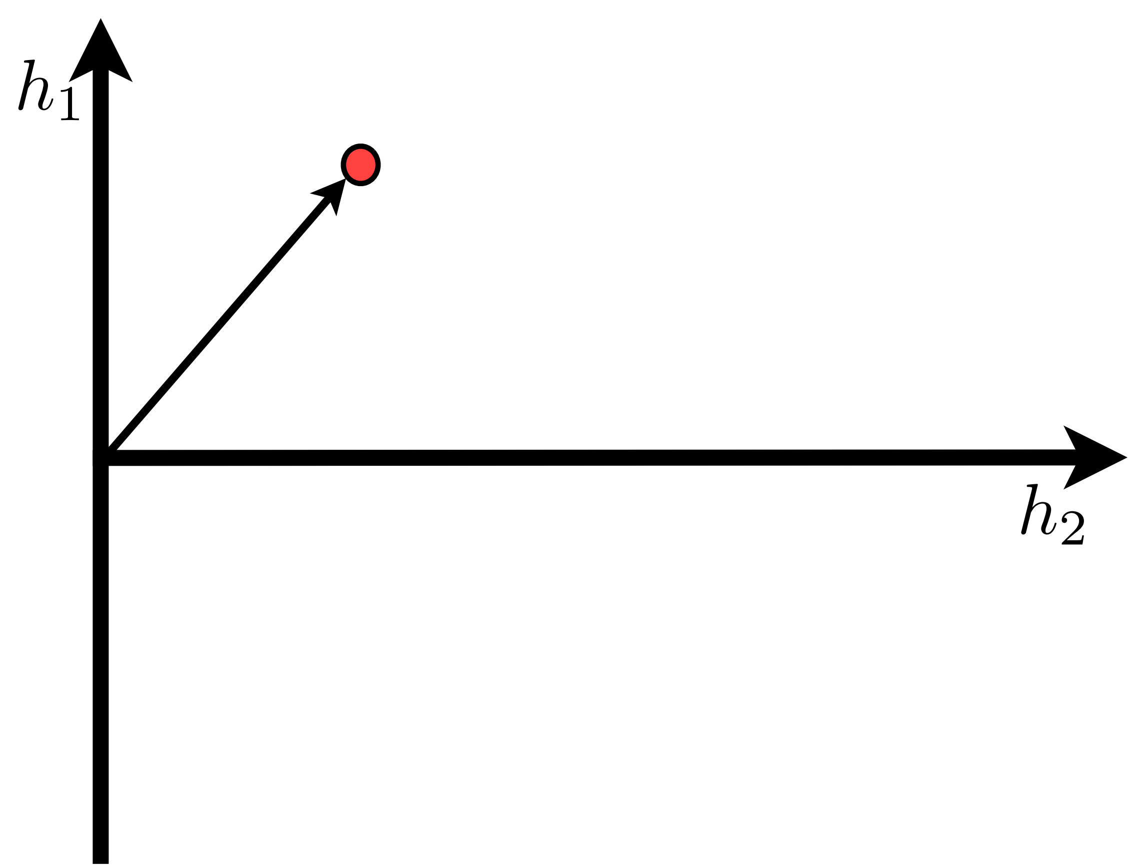
#1
#2
#3
#4
#5
#6
#7
#9
#10
Worse!
h_1
h_2
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$

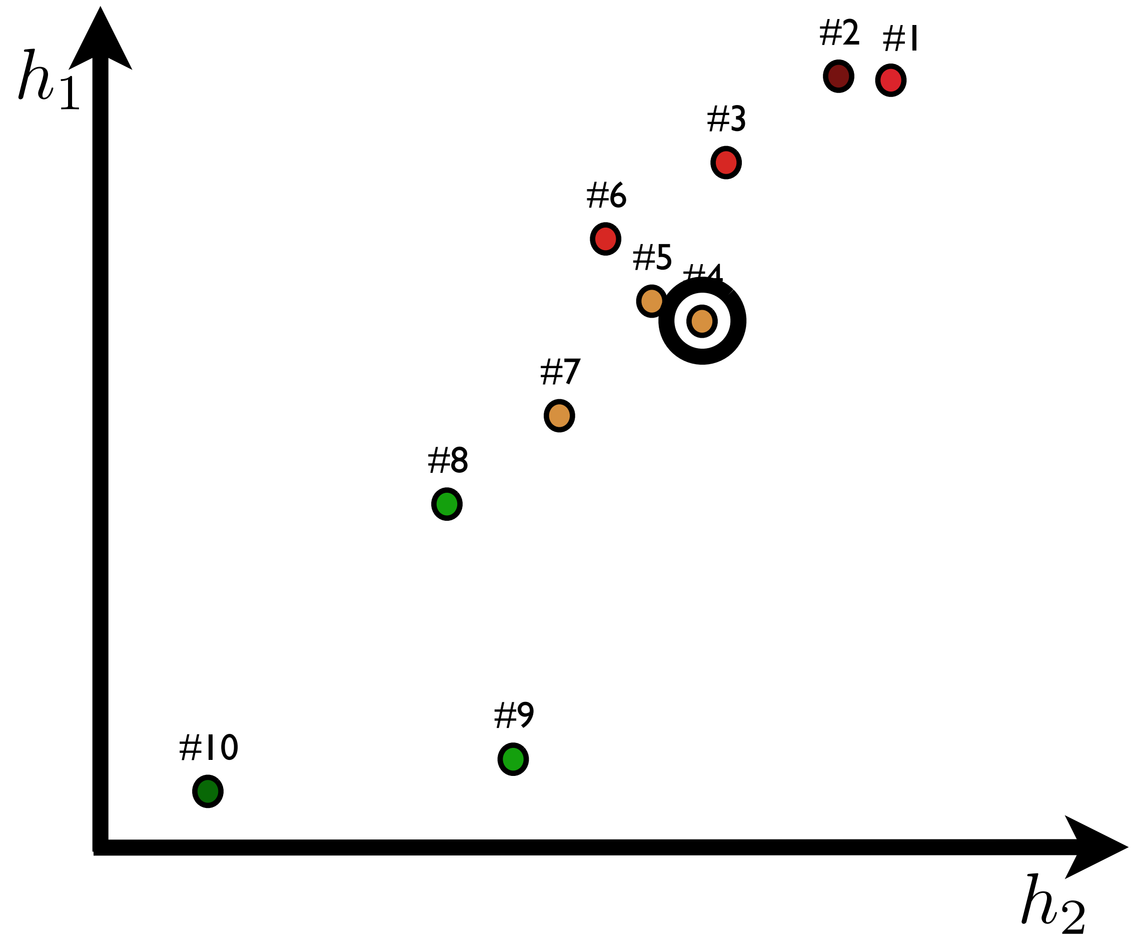


h_1
h_2

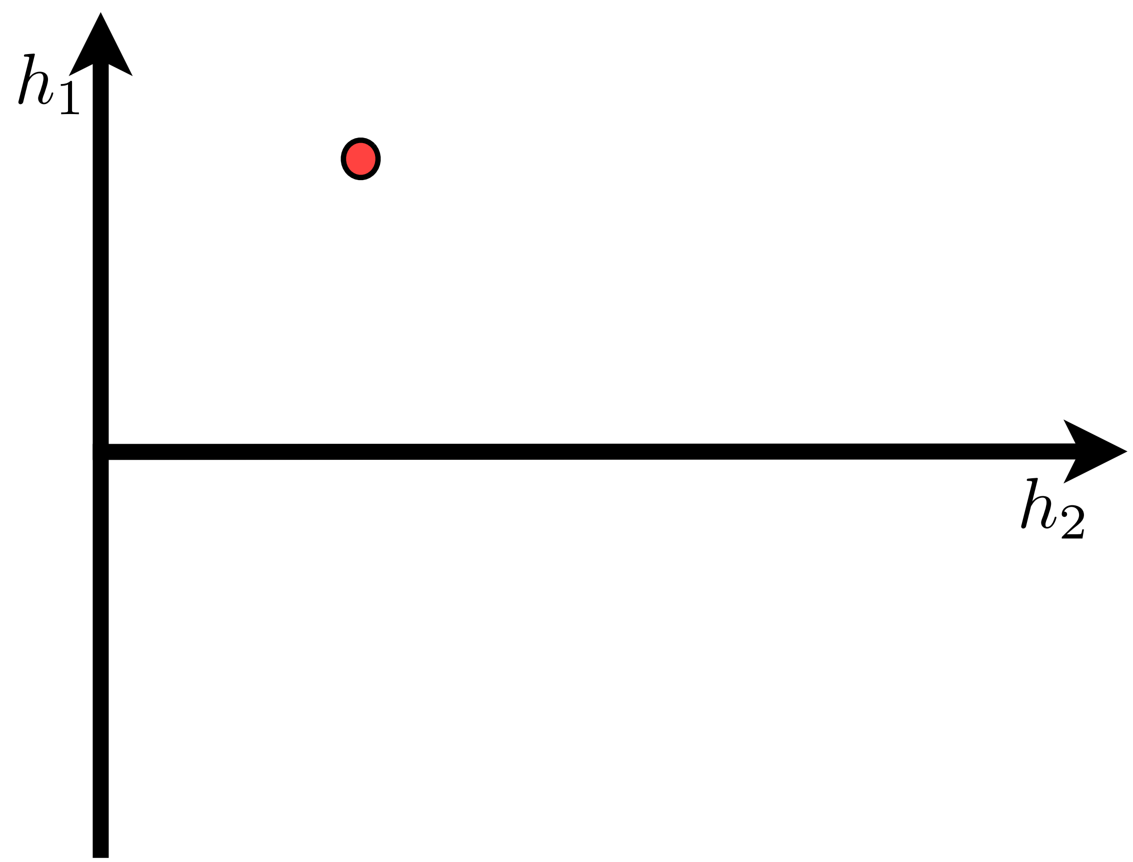


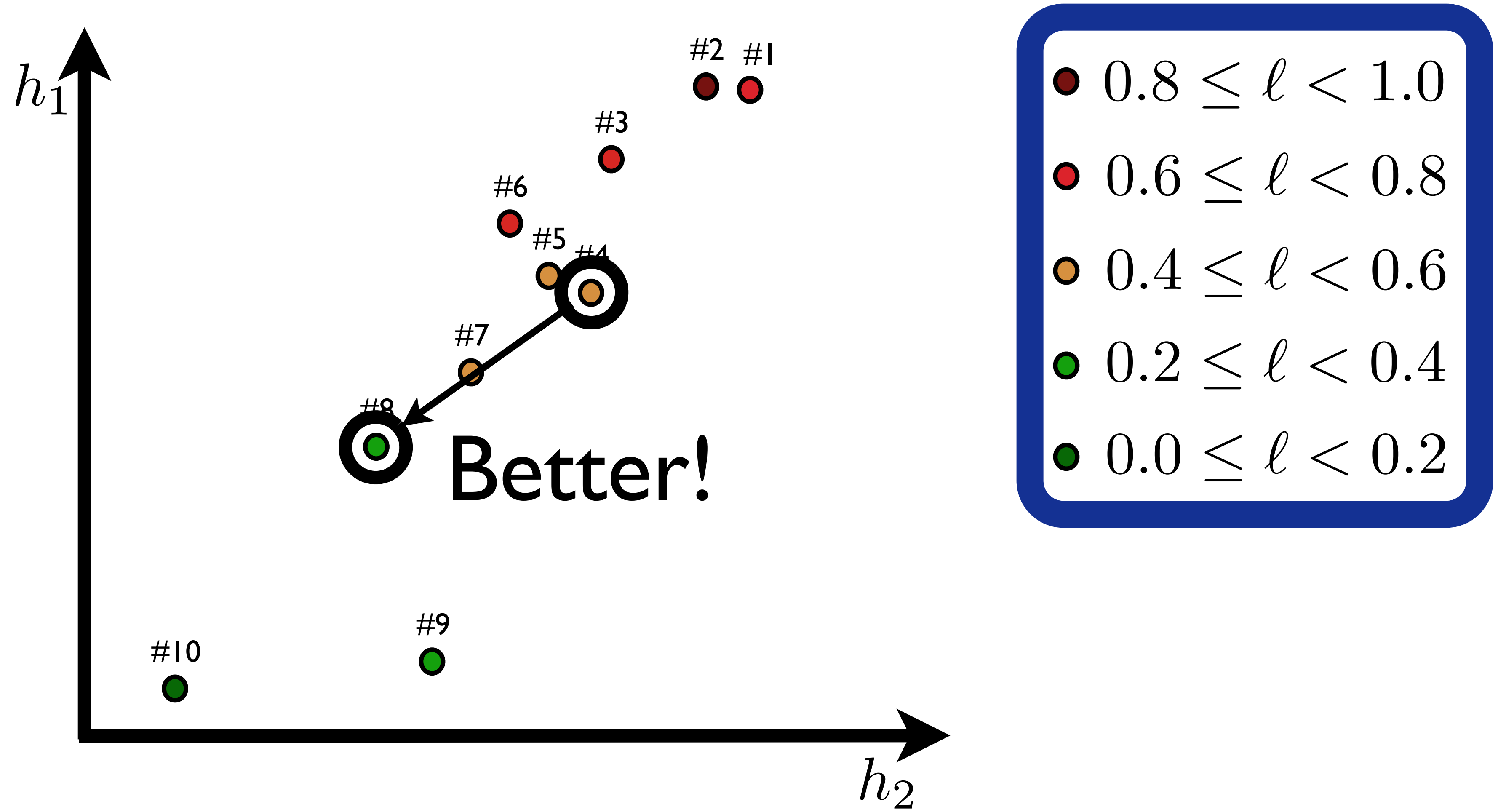
- $0.8 \leq \ell < 1.0$
- $0.6 \leq \ell < 0.8$
- $0.4 \leq \ell < 0.6$
- $0.2 \leq \ell < 0.4$
- $0.0 \leq \ell < 0.2$



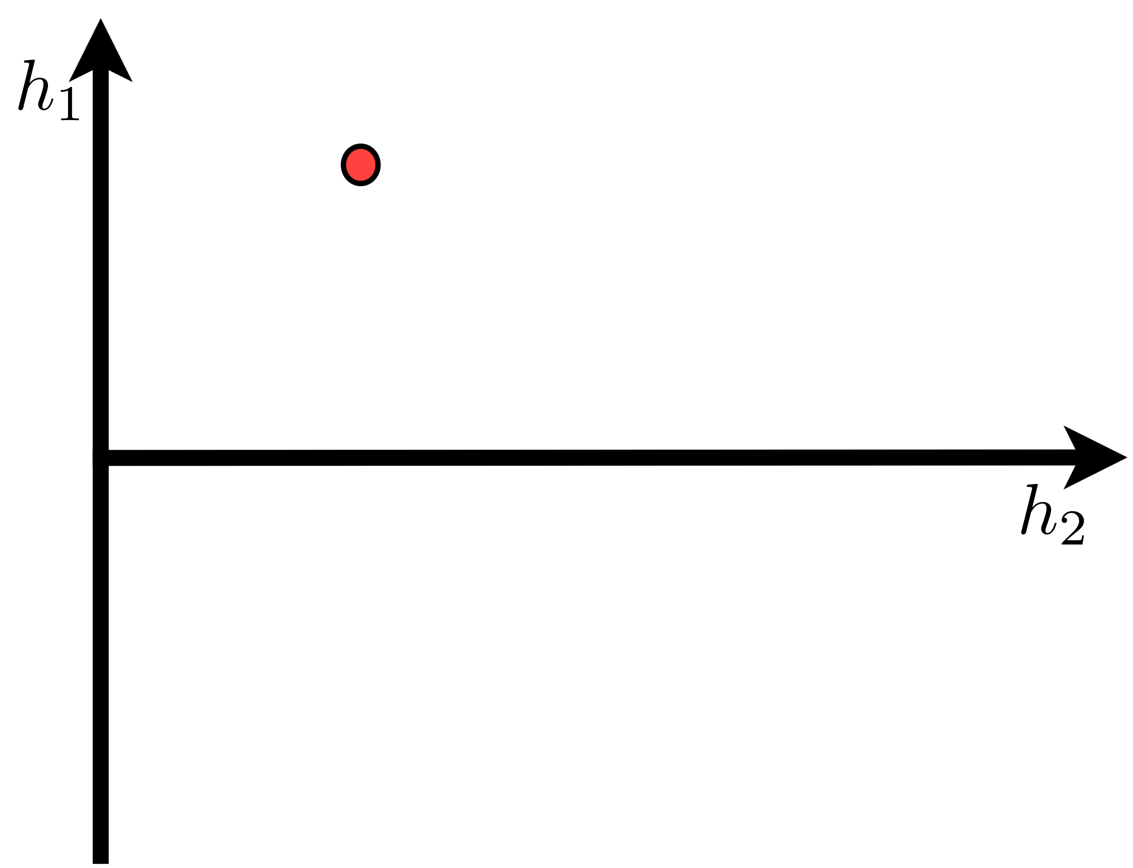


- $0.8 \leq \ell < 1.0$
- $0.6 \leq \ell < 0.8$
- $0.4 \leq \ell < 0.6$
- $0.2 \leq \ell < 0.4$
- $0.0 \leq \ell < 0.2$

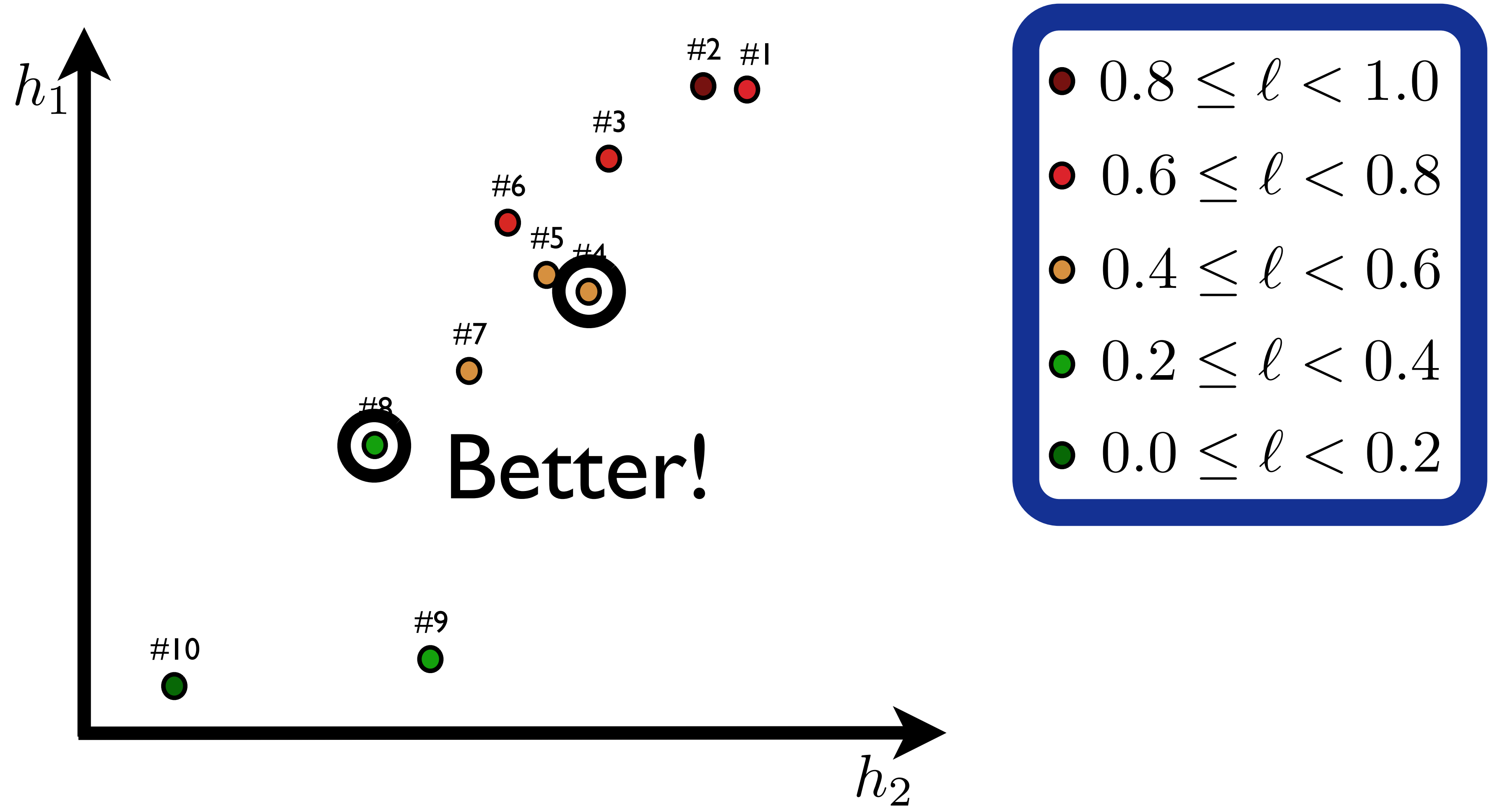


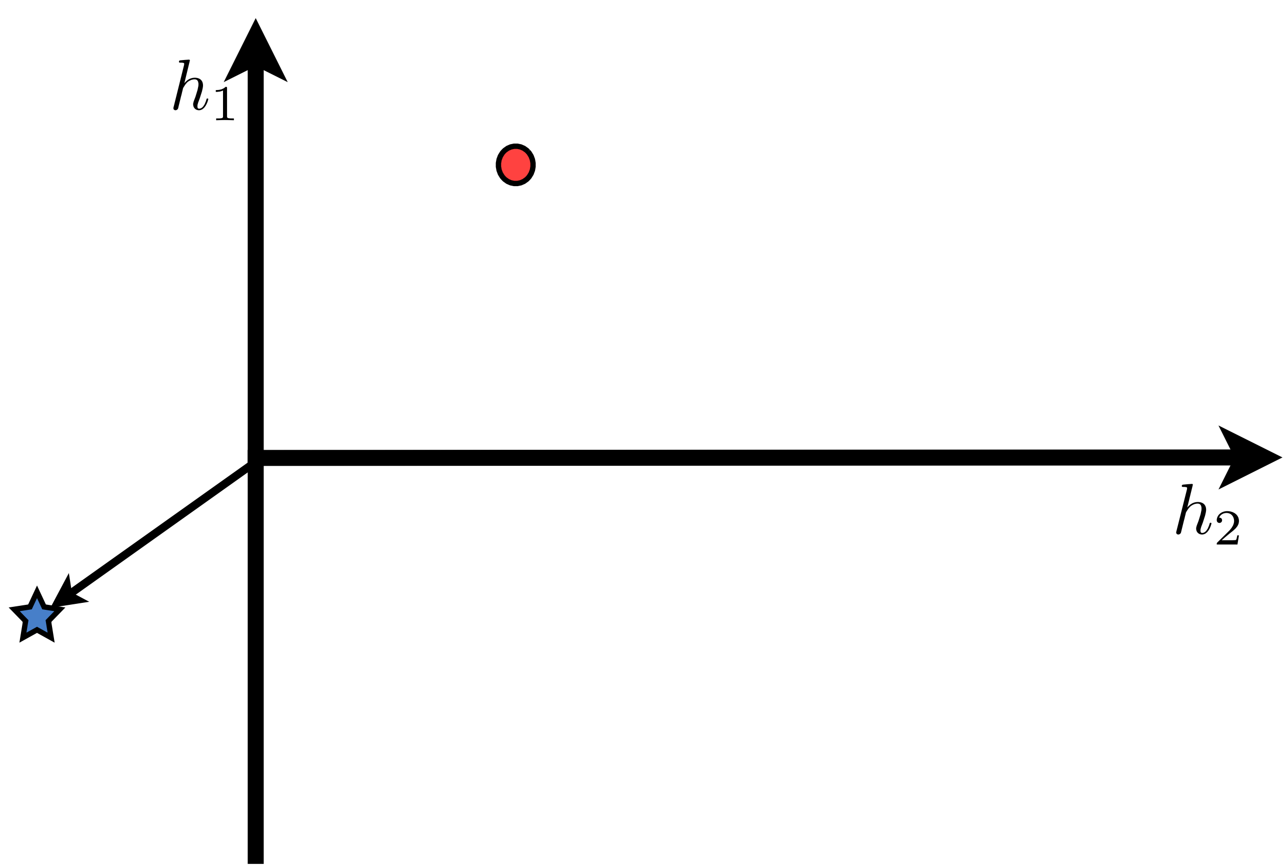


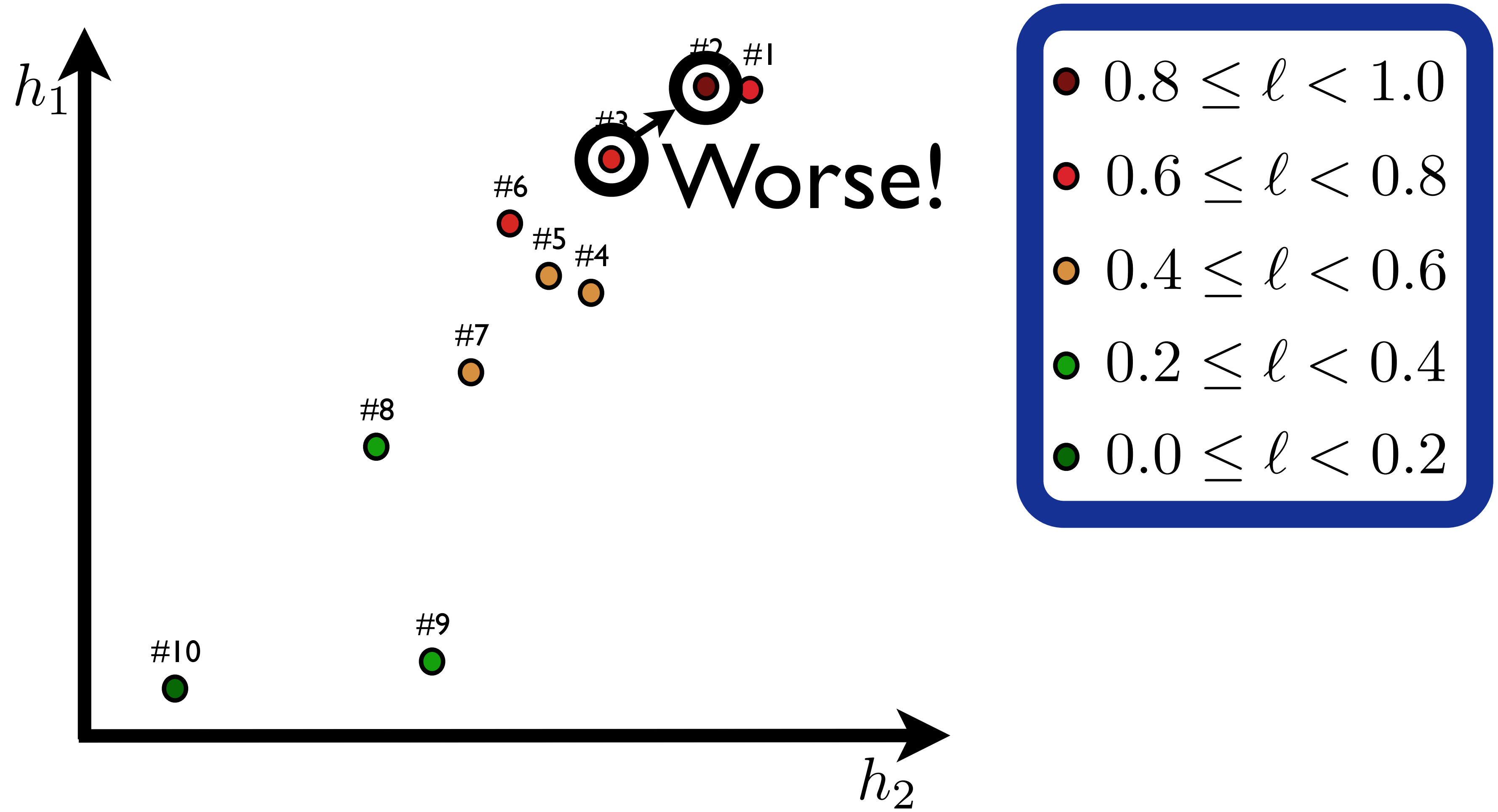
h_1
h_2
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
Better!
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$

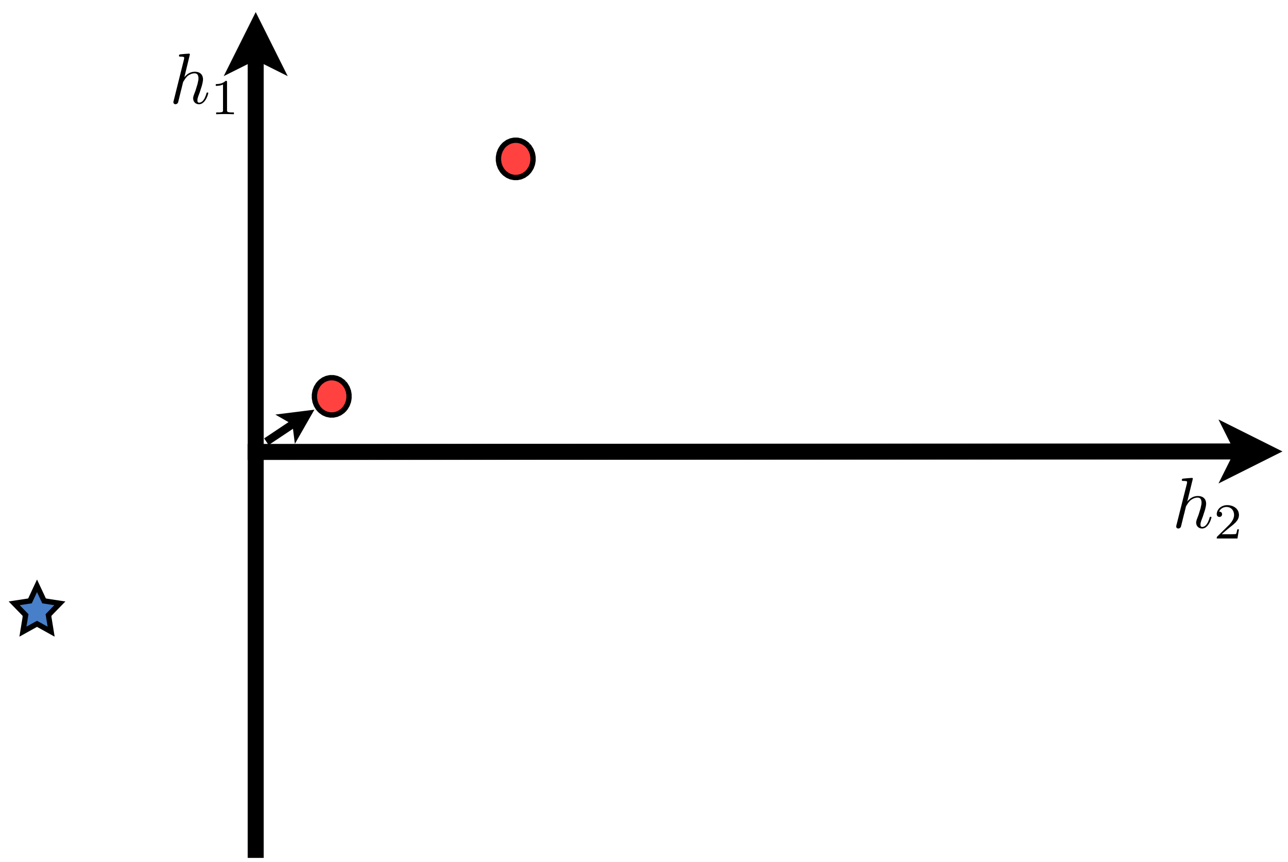


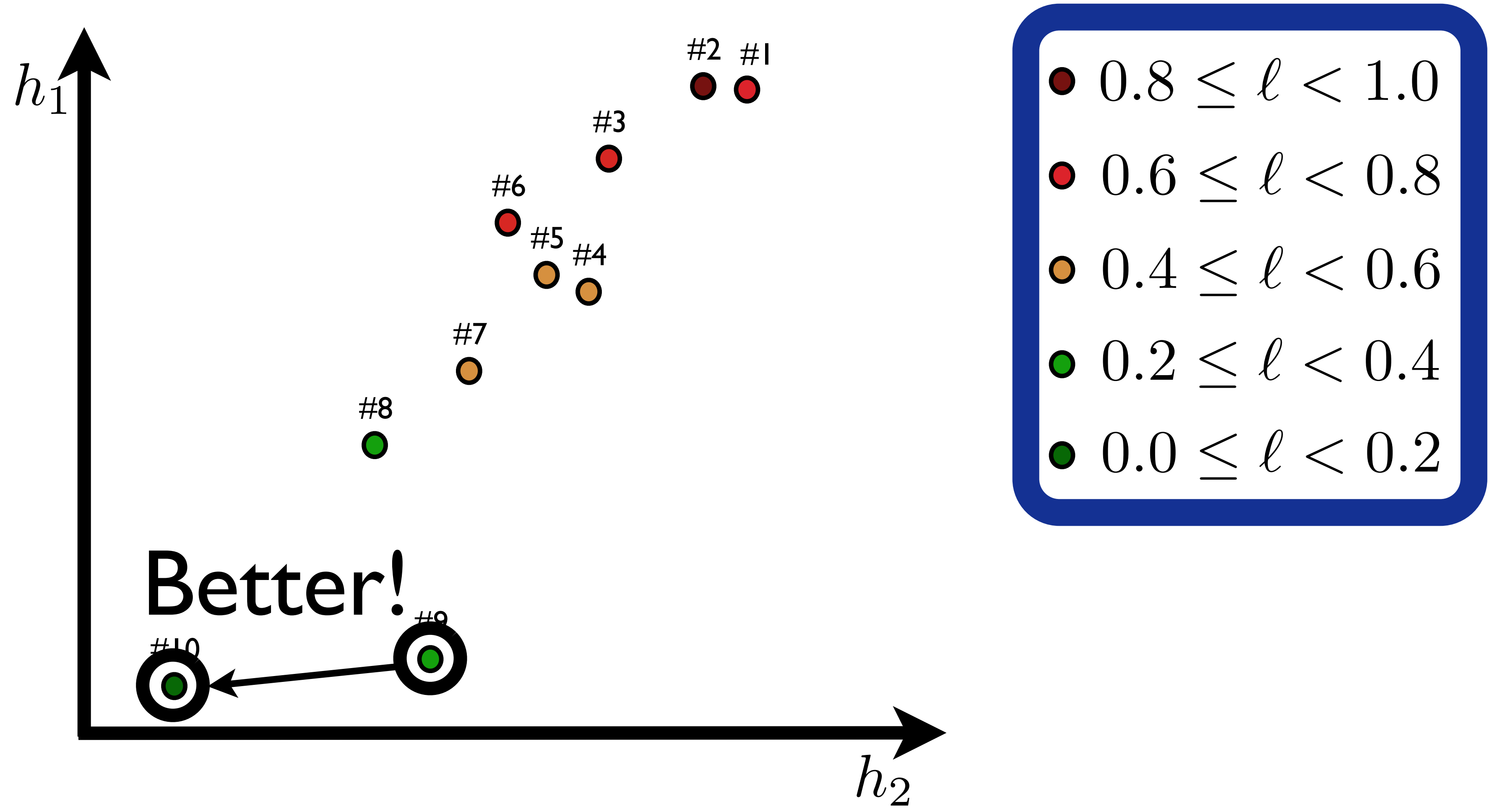
h_1
h_2


h_1
h_2
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
Better!
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$

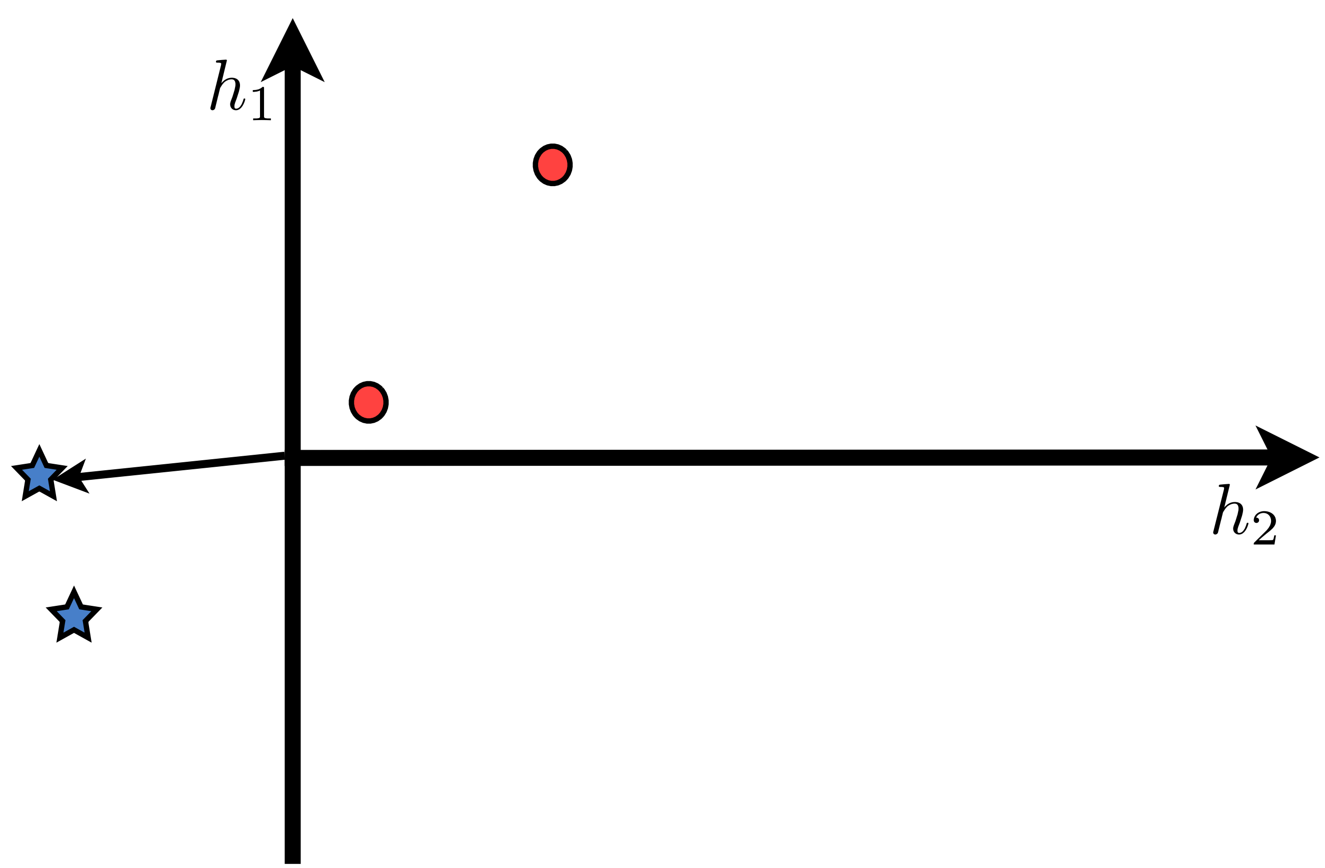

h_1
h_2


h_1
h_2
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
Worse!
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$

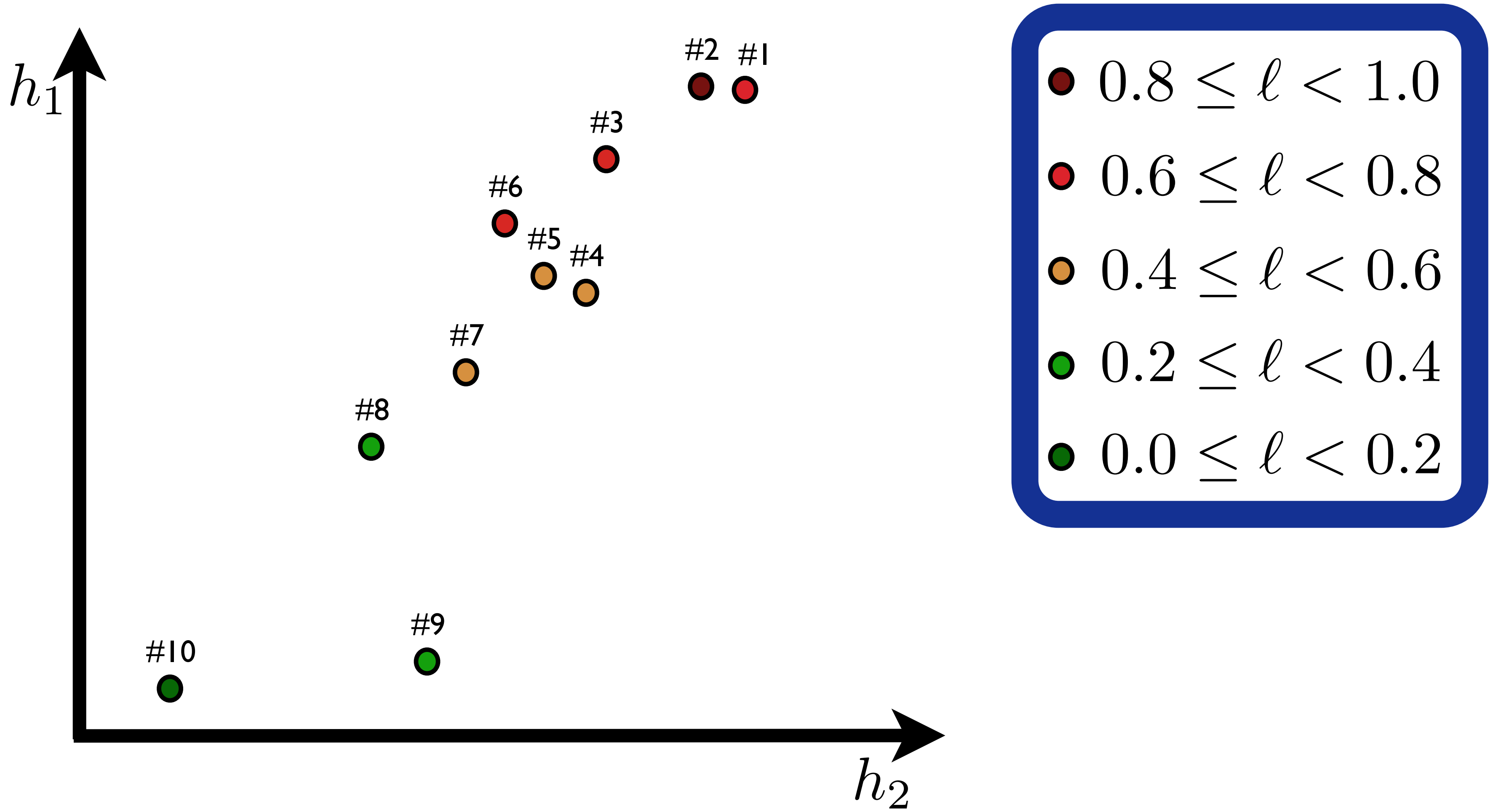

h_1
h_2



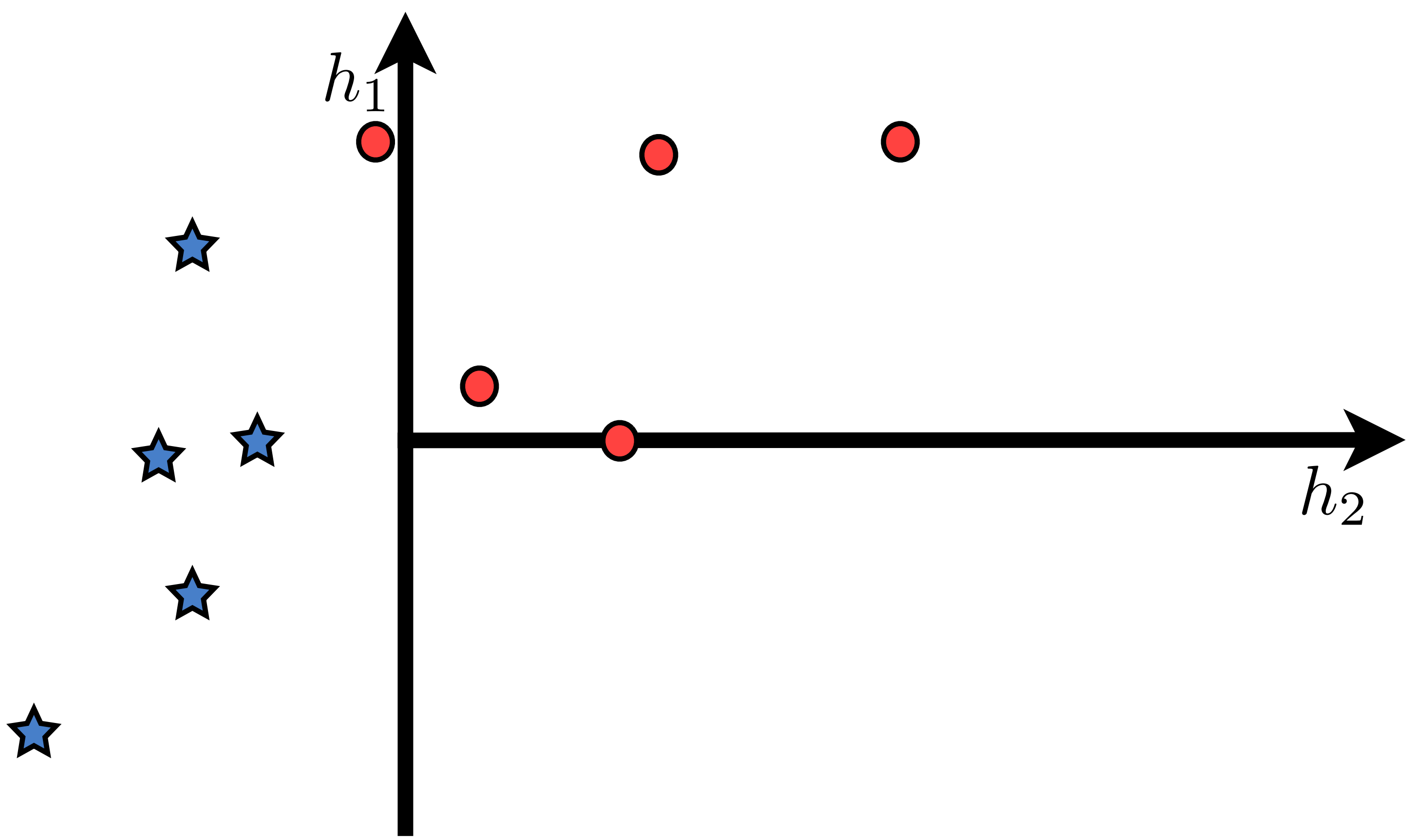
h_1
h_2
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
Better!
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$



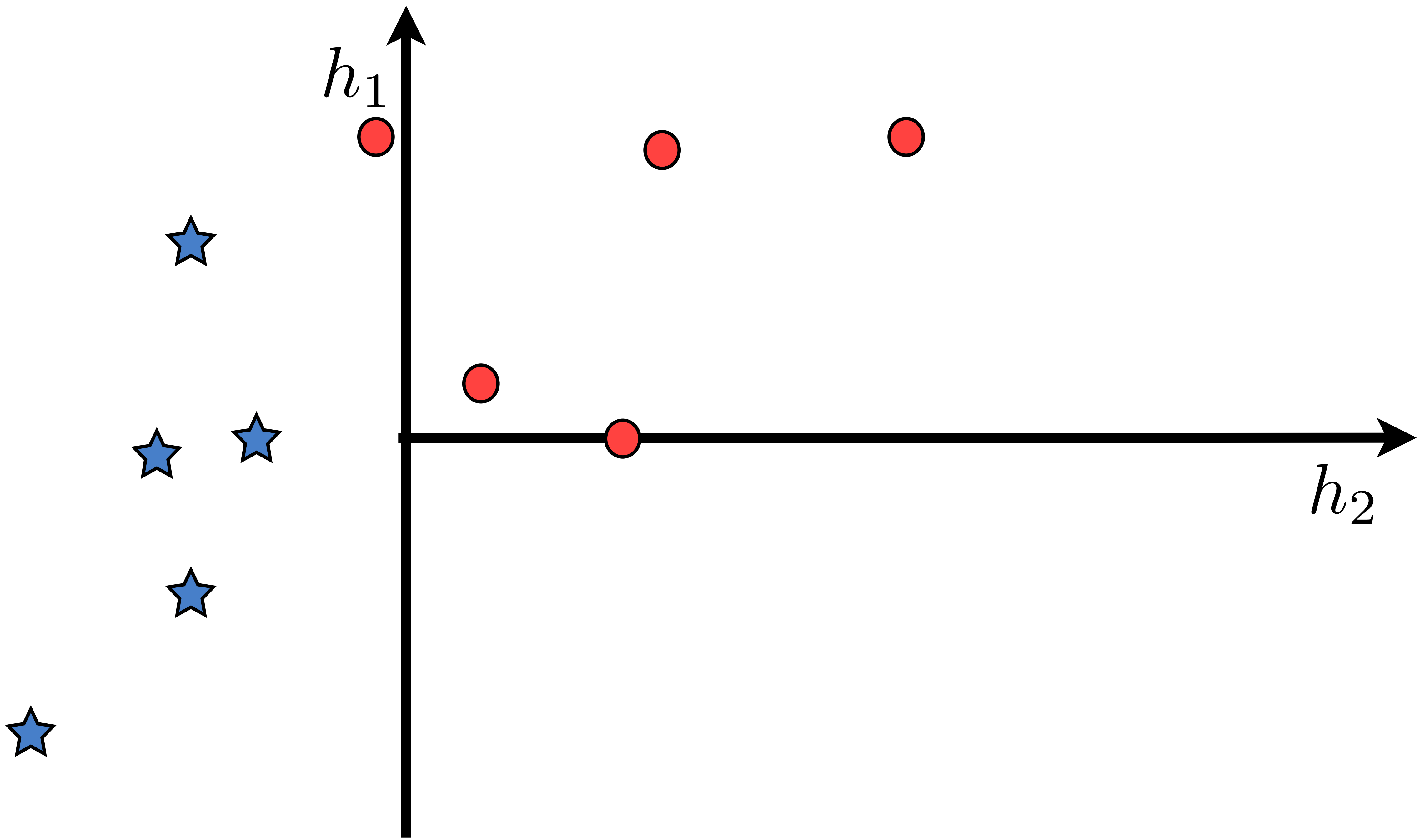
h_1
h_2



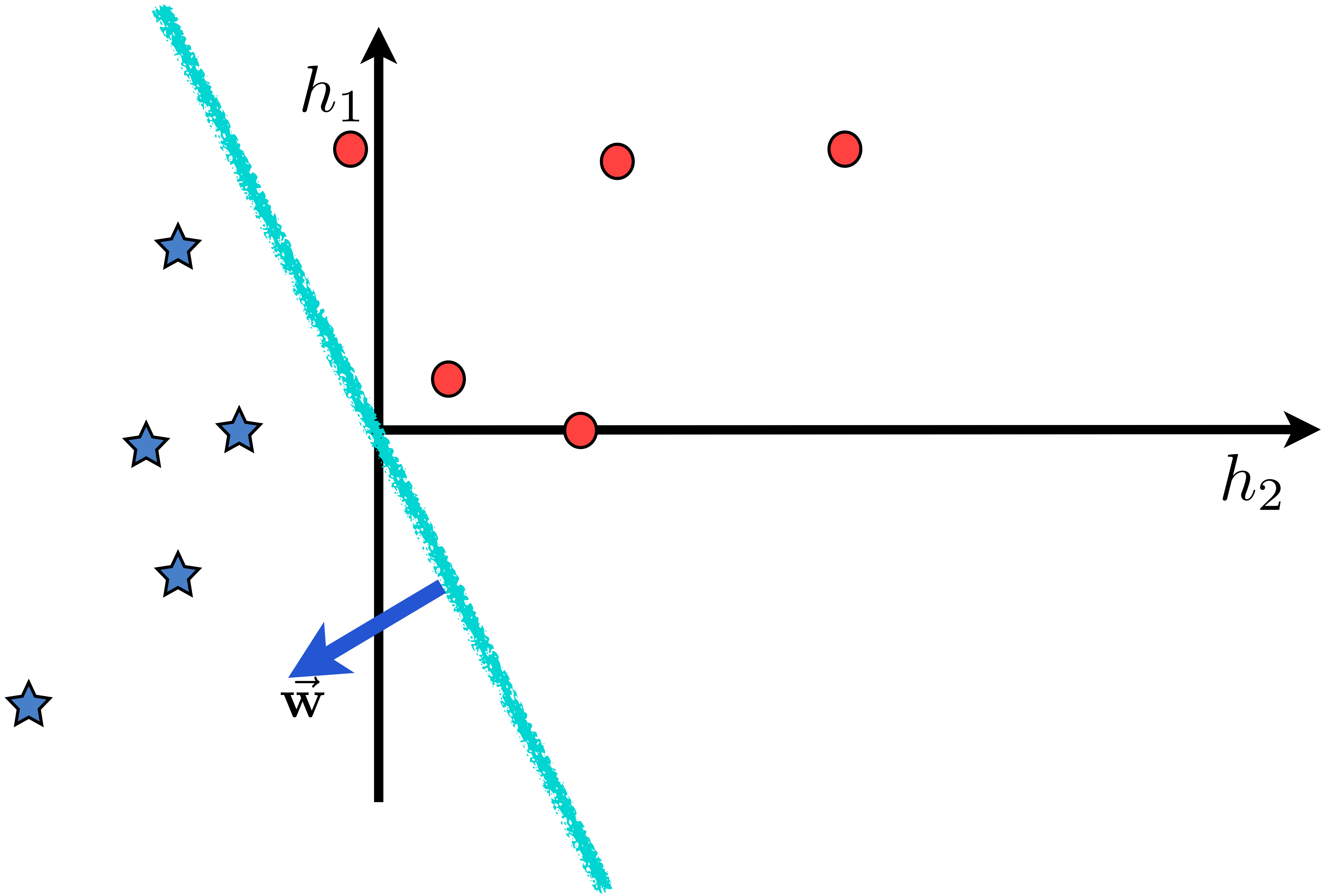
h_1
h_2
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
$0.8 \leq \ell < 1.0$
$0.6 \leq \ell < 0.8$
$0.4 \leq \ell < 0.6$
$0.2 \leq \ell < 0.4$
$0.0 \leq \ell < 0.2$



h_1
h_2

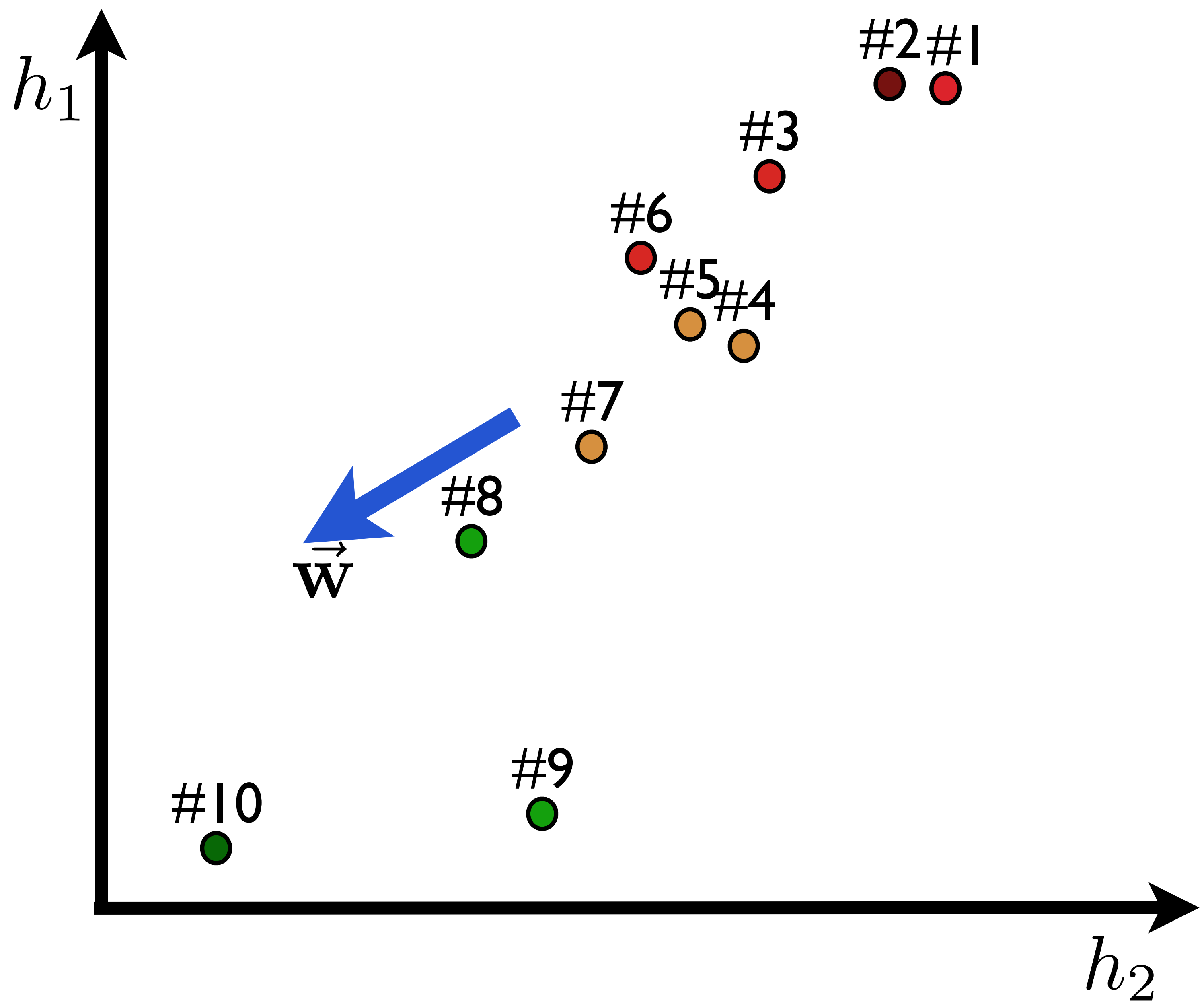


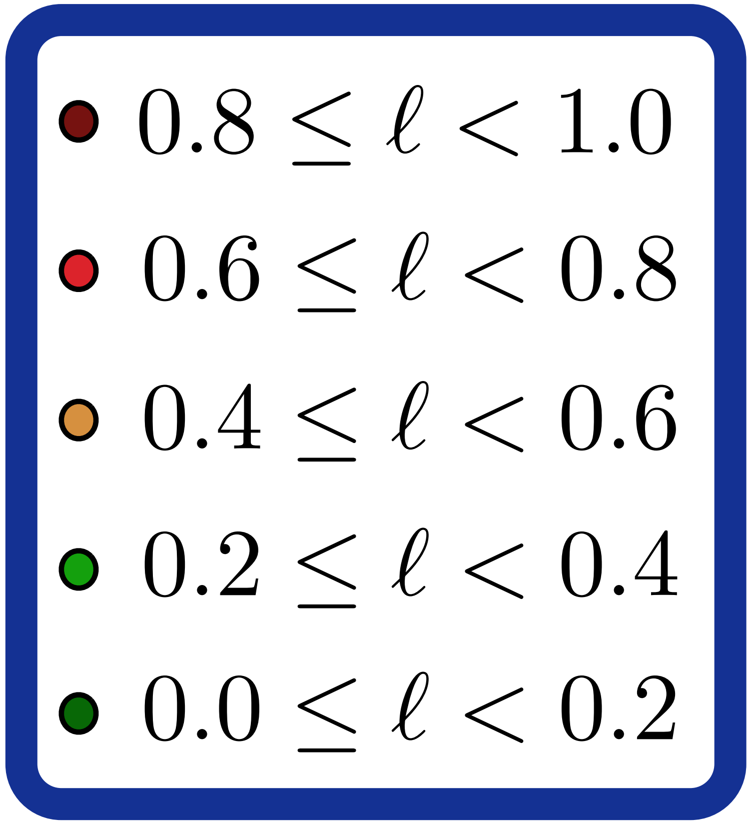
Fit a linear model



Fit a linear model

K-Best List Example





MERT

- **M**inimum **E**rror **R**ate **T**raining
- Directly target an automatic evaluation metric
  - BLEU is defined at the corpus level
  - MERT optimizes at the corpus level
- Downsides
  - Does not deal well with > ~20 features

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

$$m = (\mathbf{w} + \gamma \mathbf{v})^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$$

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

$$\begin{aligned} m &= (\mathbf{w} + \gamma \mathbf{v})^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a}) \\ &= \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a}) + \gamma\, \mathbf{v}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a}) \end{aligned}$$

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

$$\begin{aligned} m &= (\mathbf{w} + \gamma \mathbf{v})^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a}) \\ &= \underbrace{\mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{b} + \gamma \underbrace{\mathbf{v}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{a} \end{aligned}$$

$$m = a\gamma + b$$

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

$$\begin{aligned} m &= (\mathbf{w} + \gamma \mathbf{v})^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a}) \\ &= \underbrace{\mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{b} + \gamma \underbrace{\mathbf{v}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{a} \end{aligned}$$

$$m = a\gamma + b$$

MERT

Given weight vector $\mathbf{w}$, any hypothesis $\langle \mathbf{e}, \mathbf{a} \rangle$ will have a (scalar) score $m = \mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$

Now pick a **search vector** $\mathbf{v}$, and consider how the score of this hypothesis will change:

$$\mathbf{w}_{\text{new}} = \mathbf{w} + \gamma \mathbf{v}$$

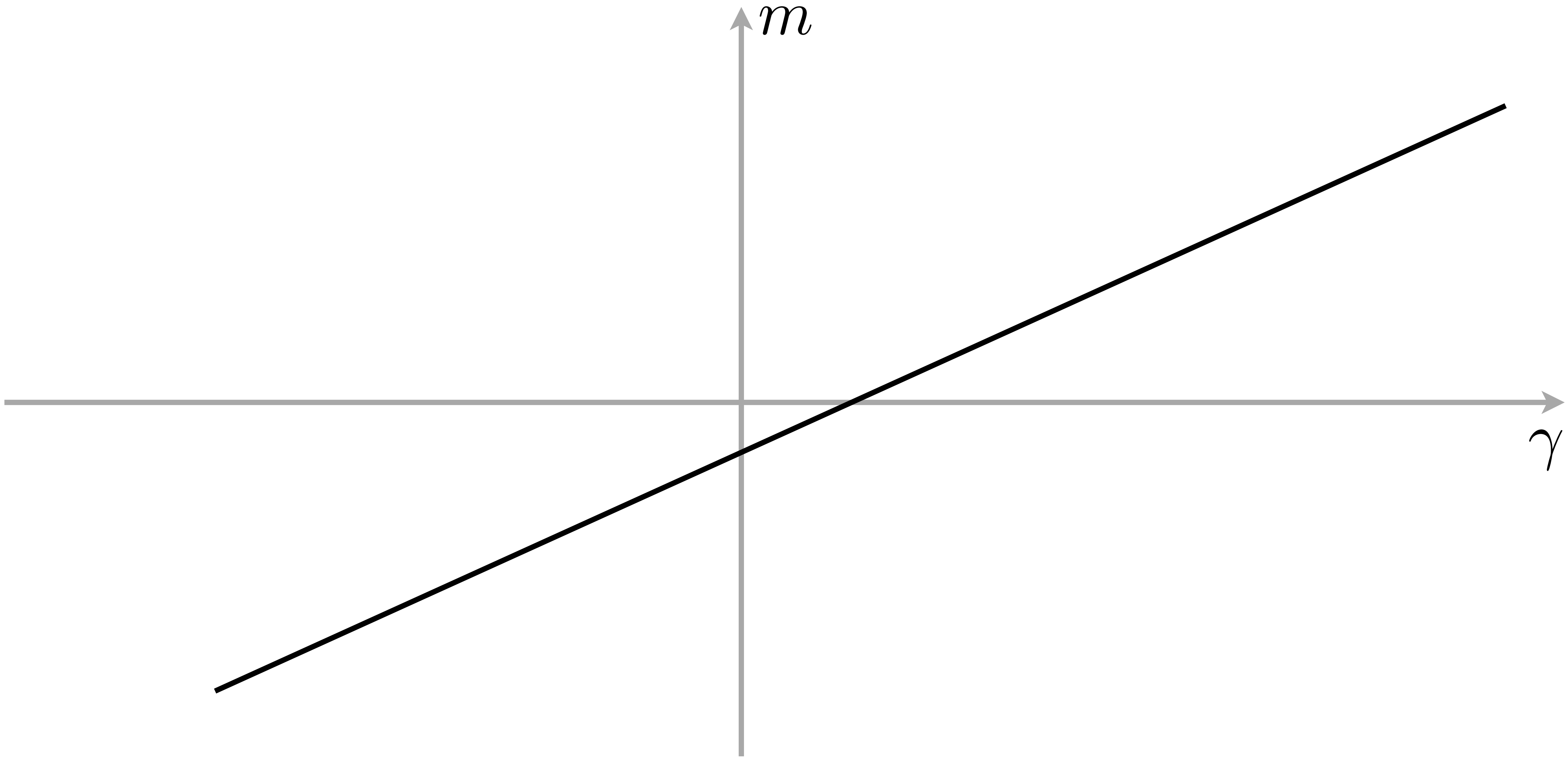
$$m = (\mathbf{w} + \gamma \mathbf{v})^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})$$

$$= \underbrace{\mathbf{w}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{b} + \gamma \underbrace{\mathbf{v}^\top \mathbf{h}(\mathbf{g}, \mathbf{e}, \mathbf{a})}_{a}$$

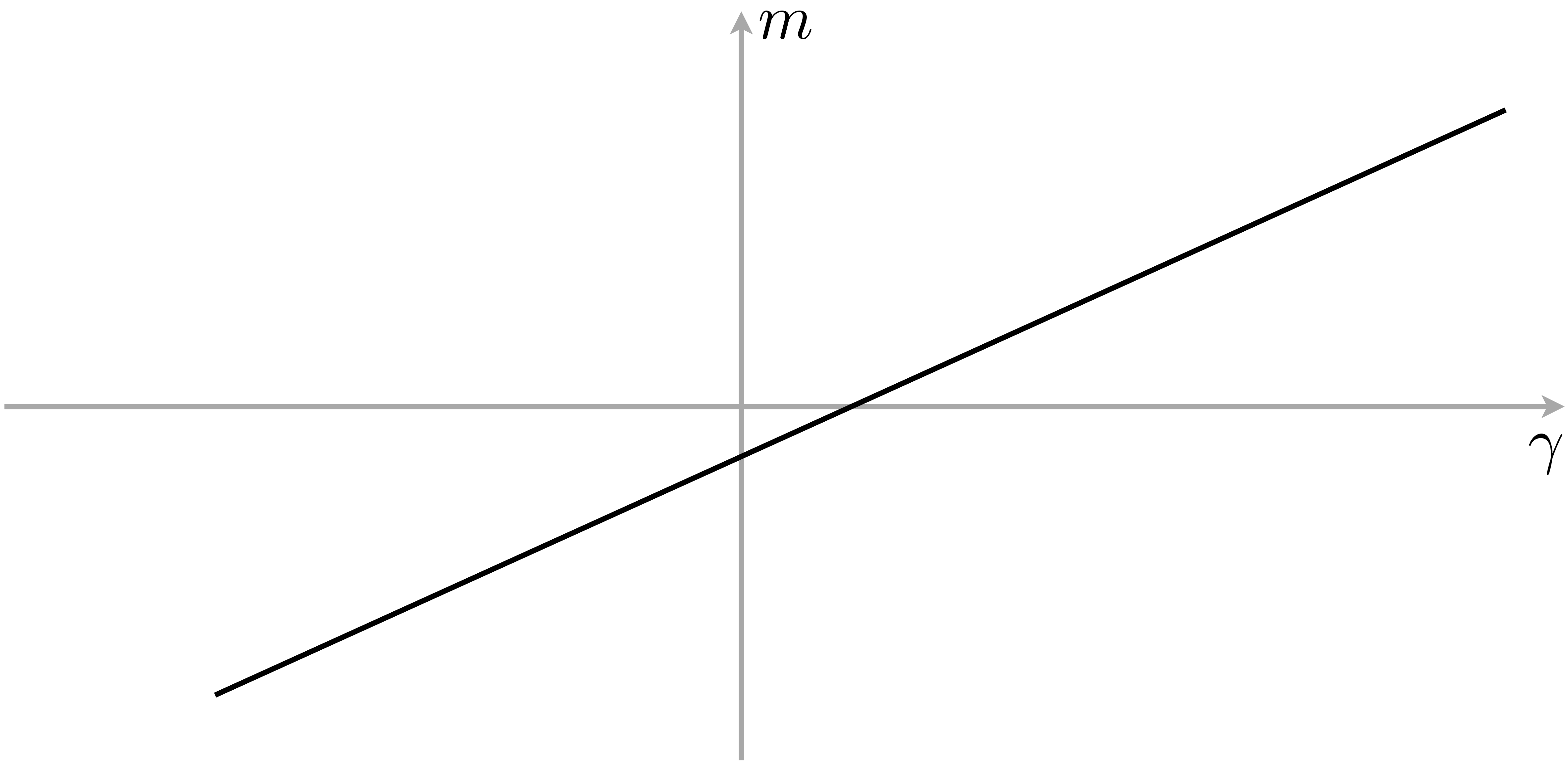
$$\boxed{m = a\gamma + b}$$

Linear function in 2D!

MERT

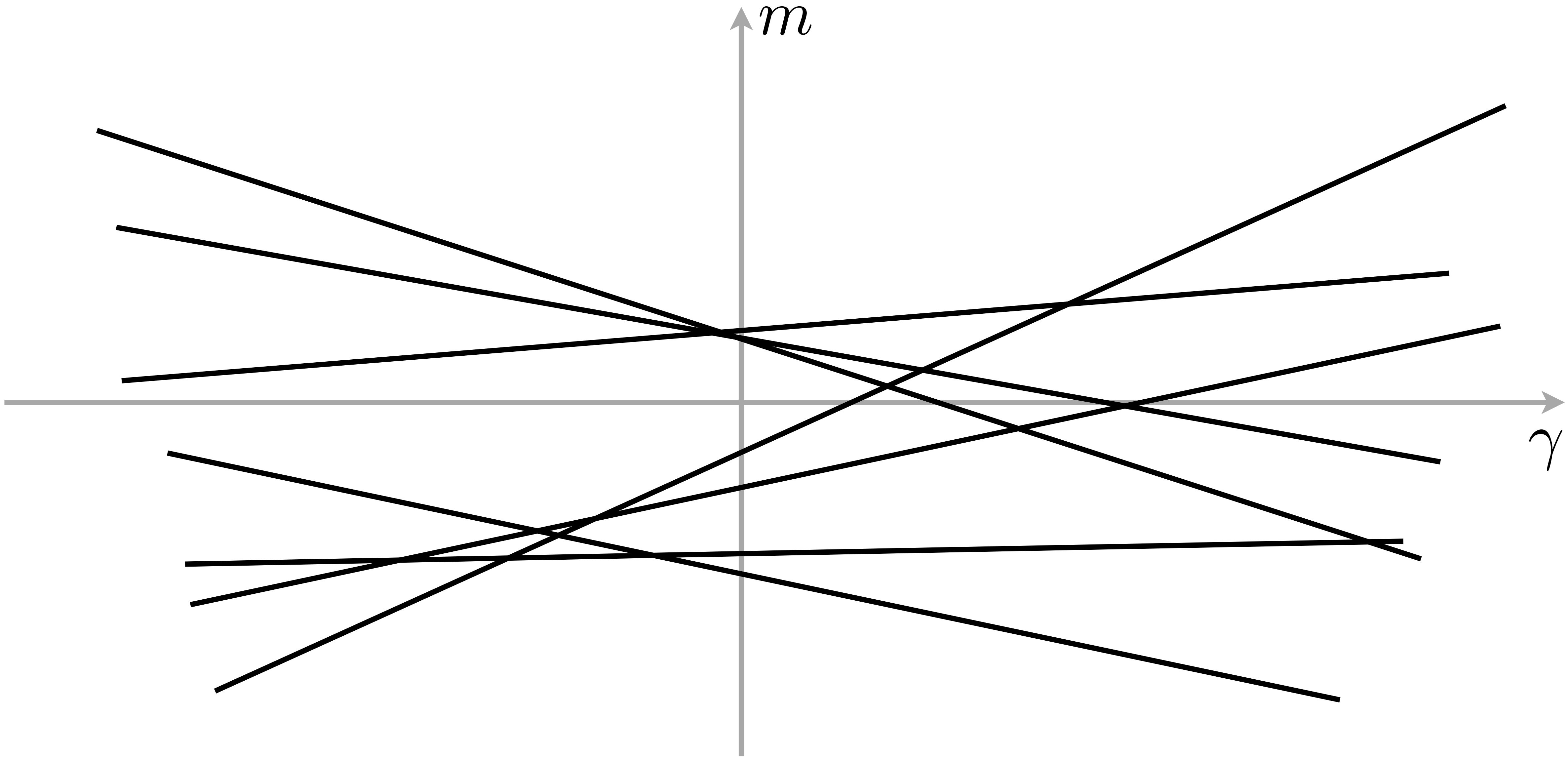


MERT



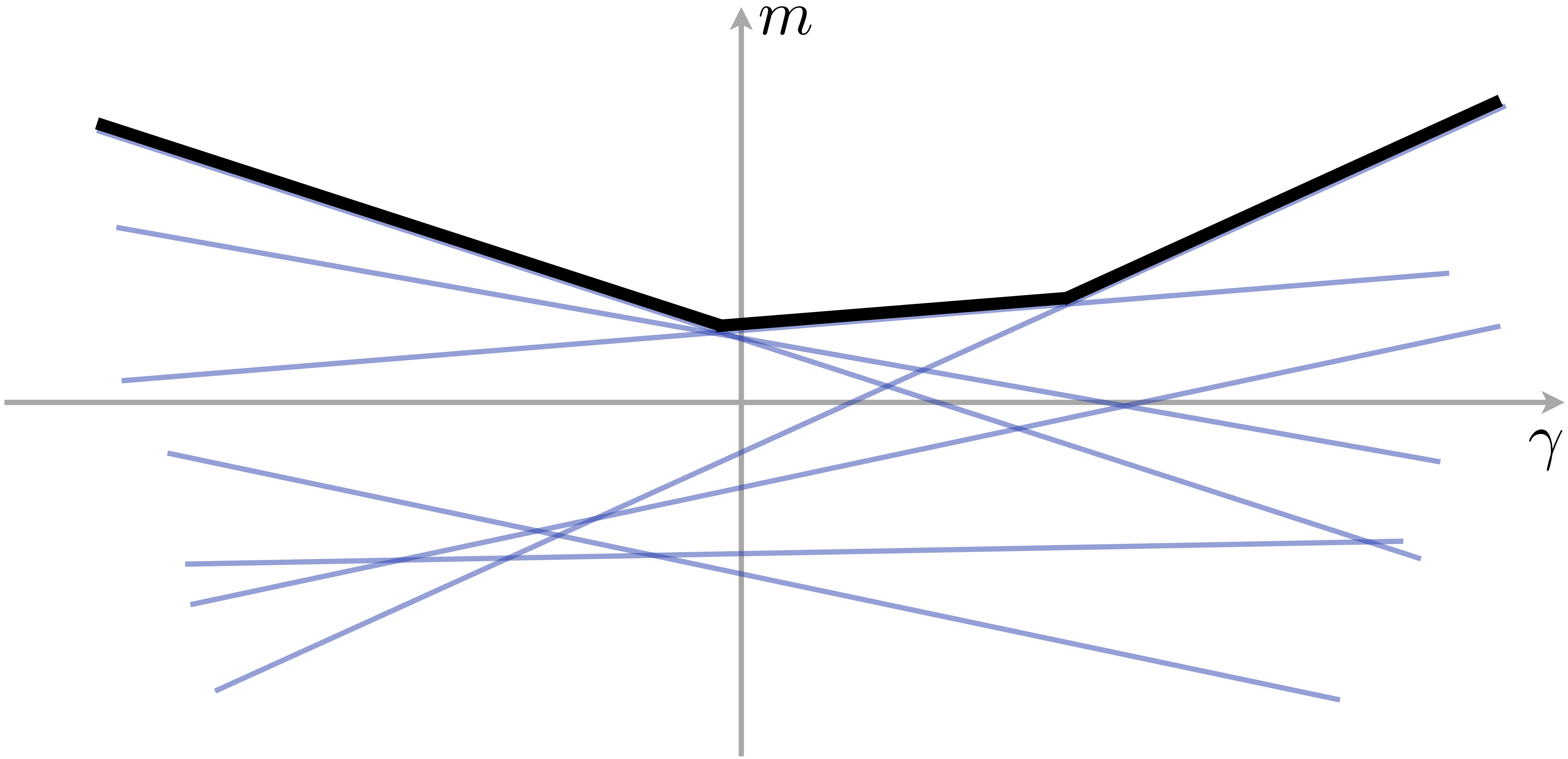
Recall our k-best set $\{\langle \mathbf{e}_i^*, \mathbf{a}_i^* \rangle\}_{i=1}^{K}$

MERT

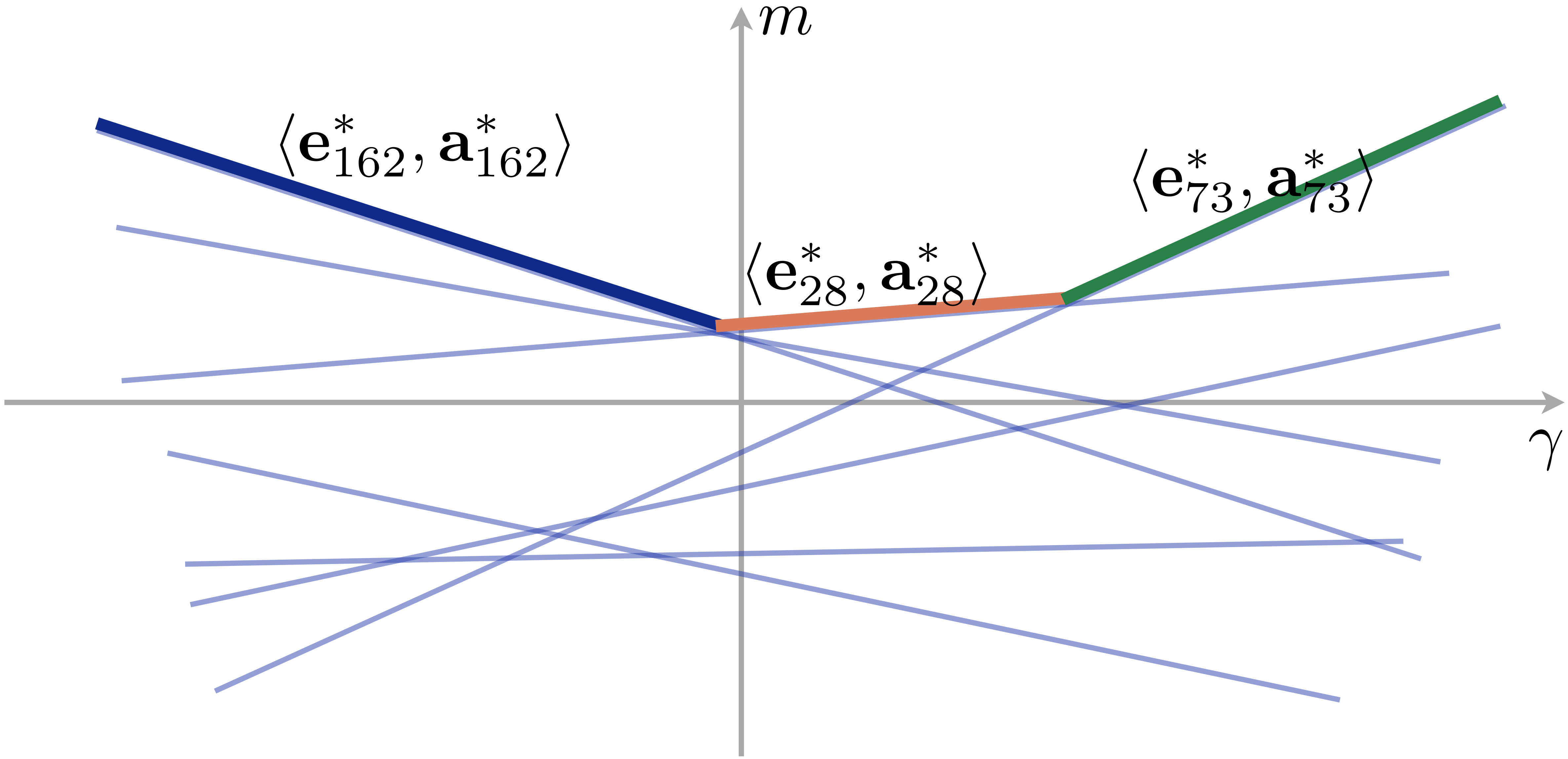


Recall our k-best set $\{\langle \mathbf{e}_i^*, \mathbf{a}_i^* \rangle\}_{i=1}^K$

MERT

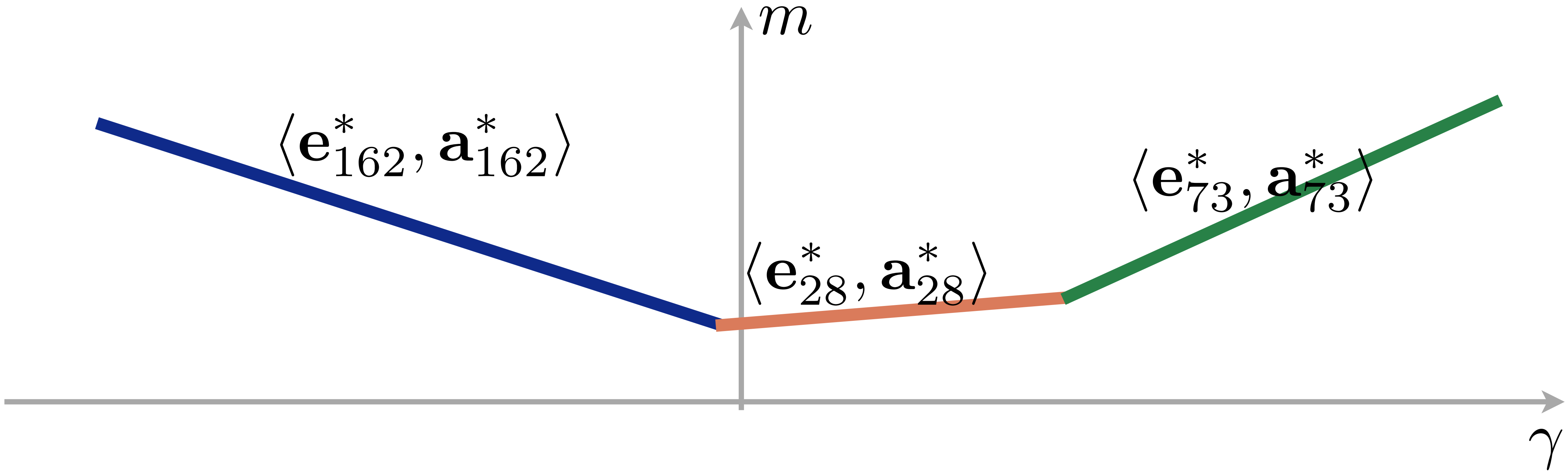


MERT



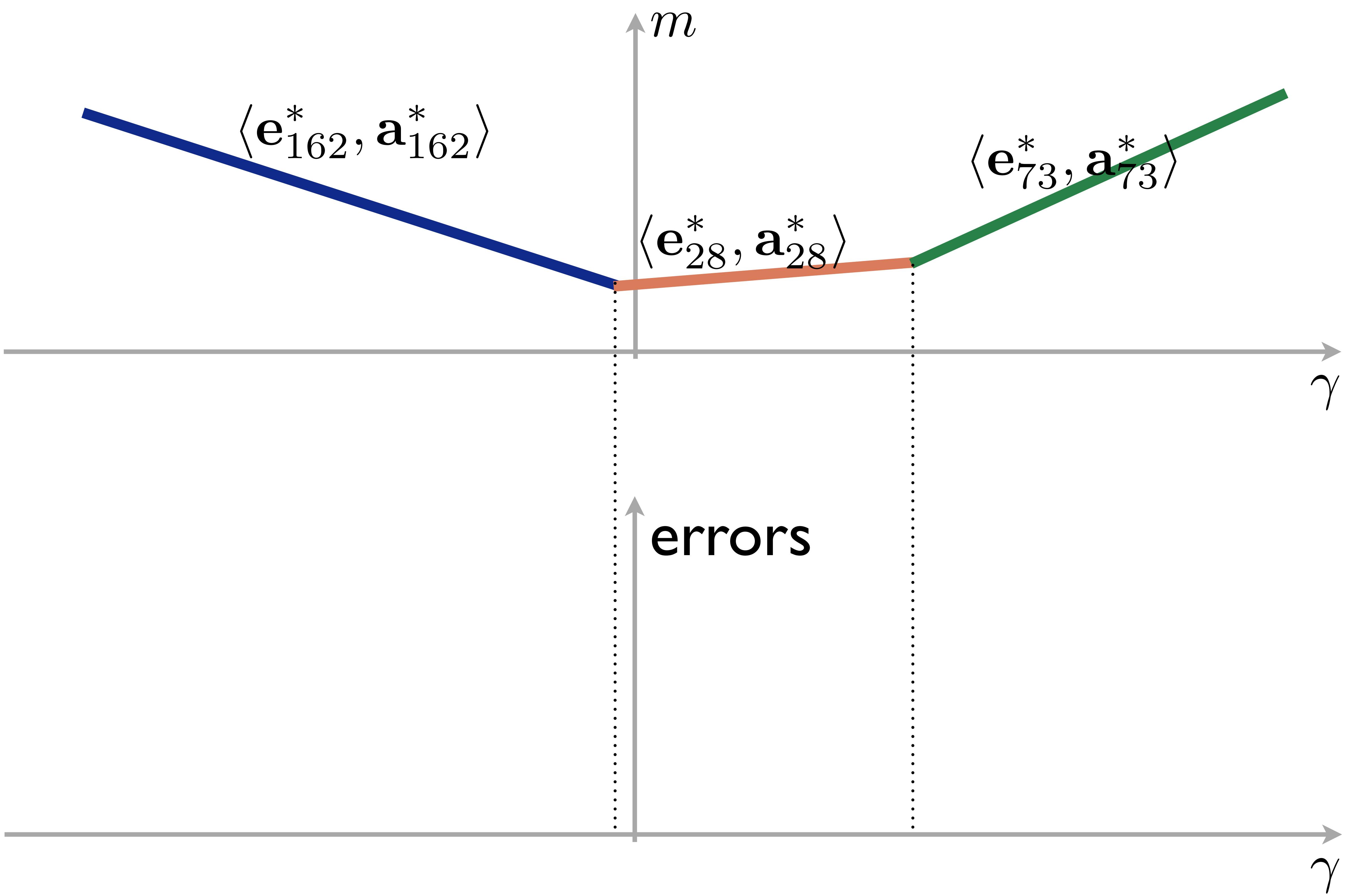
m
$\langle \mathbf{e}^*_{162}, \mathbf{a}^*_{162} \rangle$
$\langle \mathbf{e}^*_{73}, \mathbf{a}^*_{73} \rangle$
$\langle \mathbf{e}^*_{28}, \mathbf{a}^*_{28} \rangle$
γ

MERT



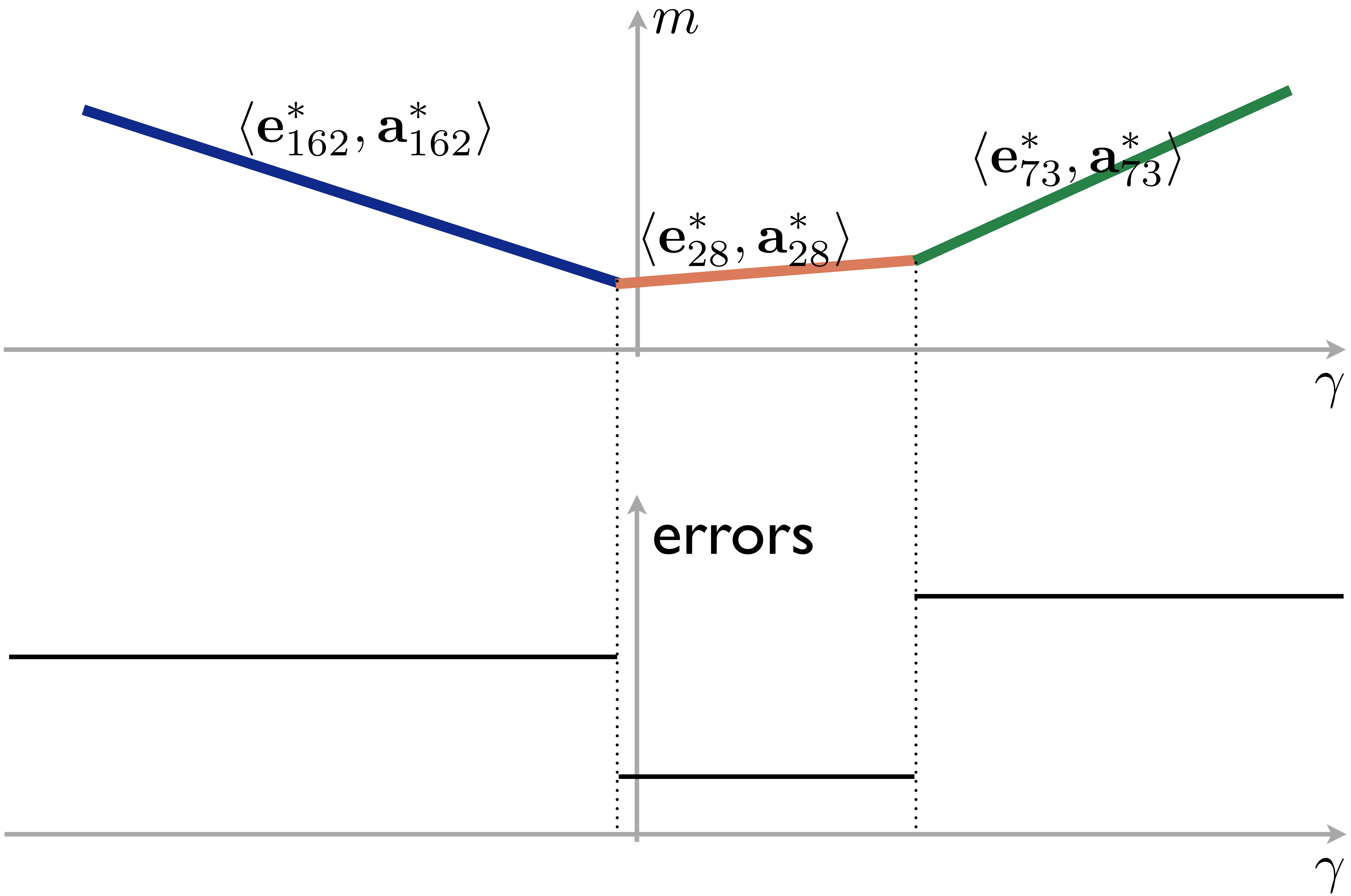
m
$\langle \mathbf{e}^*_{162}, \mathbf{a}^*_{162} \rangle$
$\langle \mathbf{e}^*_{28}, \mathbf{a}^*_{28} \rangle$
$\langle \mathbf{e}^*_{73}, \mathbf{a}^*_{73} \rangle$
γ

MERT

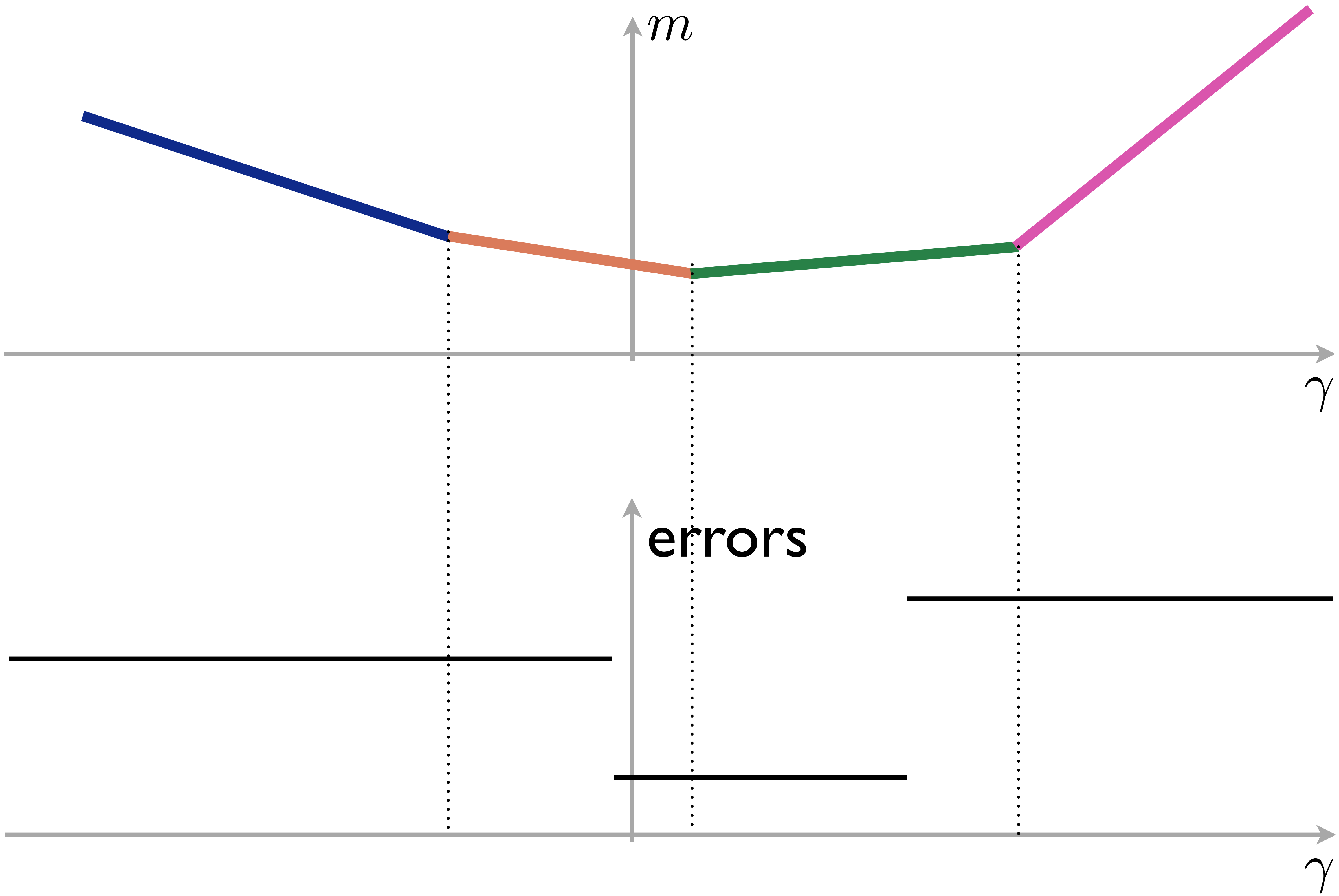


m
$\langle \mathbf{e}^*_{162}, \mathbf{a}^*_{162} \rangle$
$\langle \mathbf{e}^*_{73}, \mathbf{a}^*_{73} \rangle$
$\langle \mathbf{e}^*_{28}, \mathbf{a}^*_{28} \rangle$
γ
errors
γ

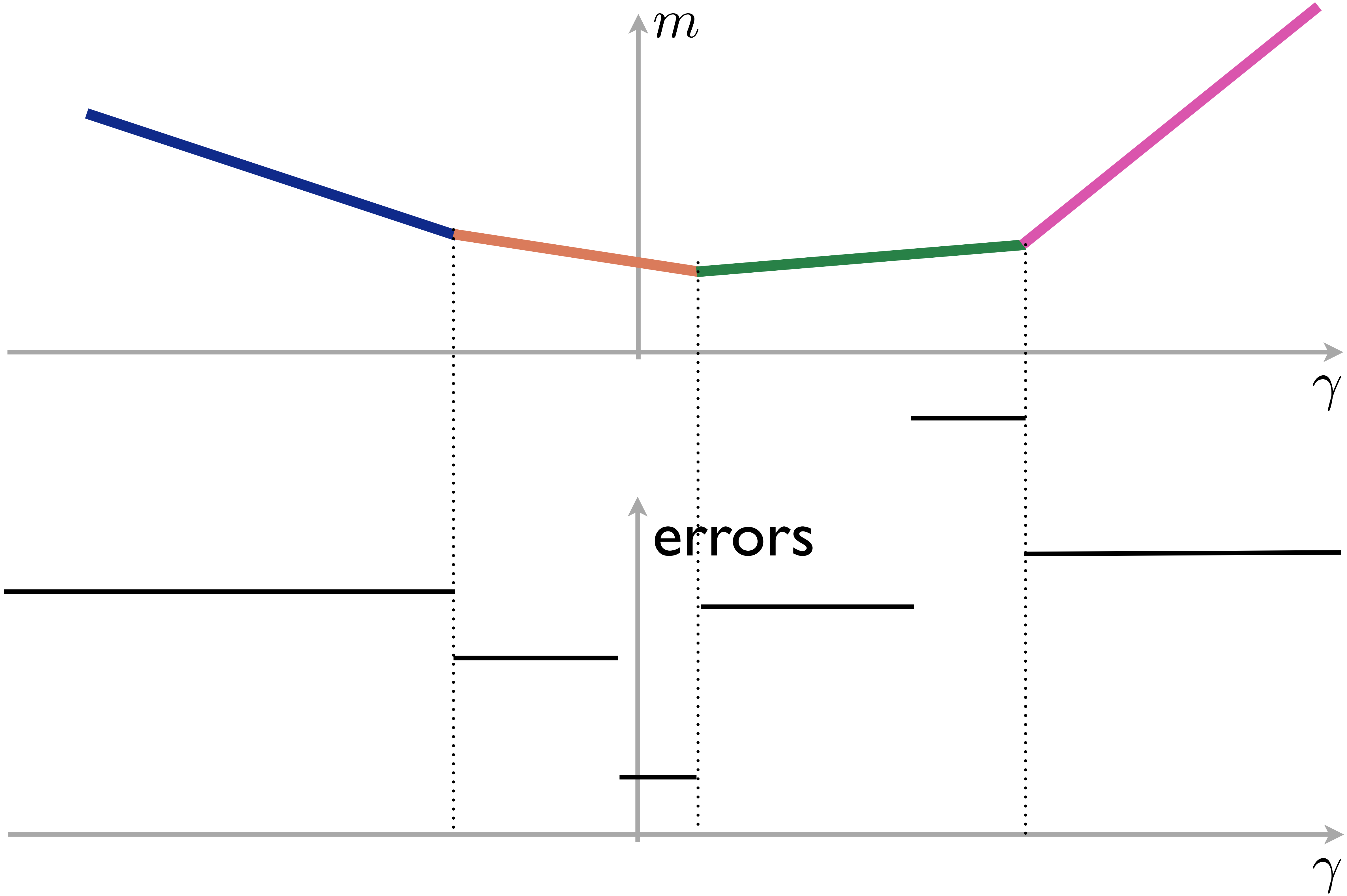
MERT

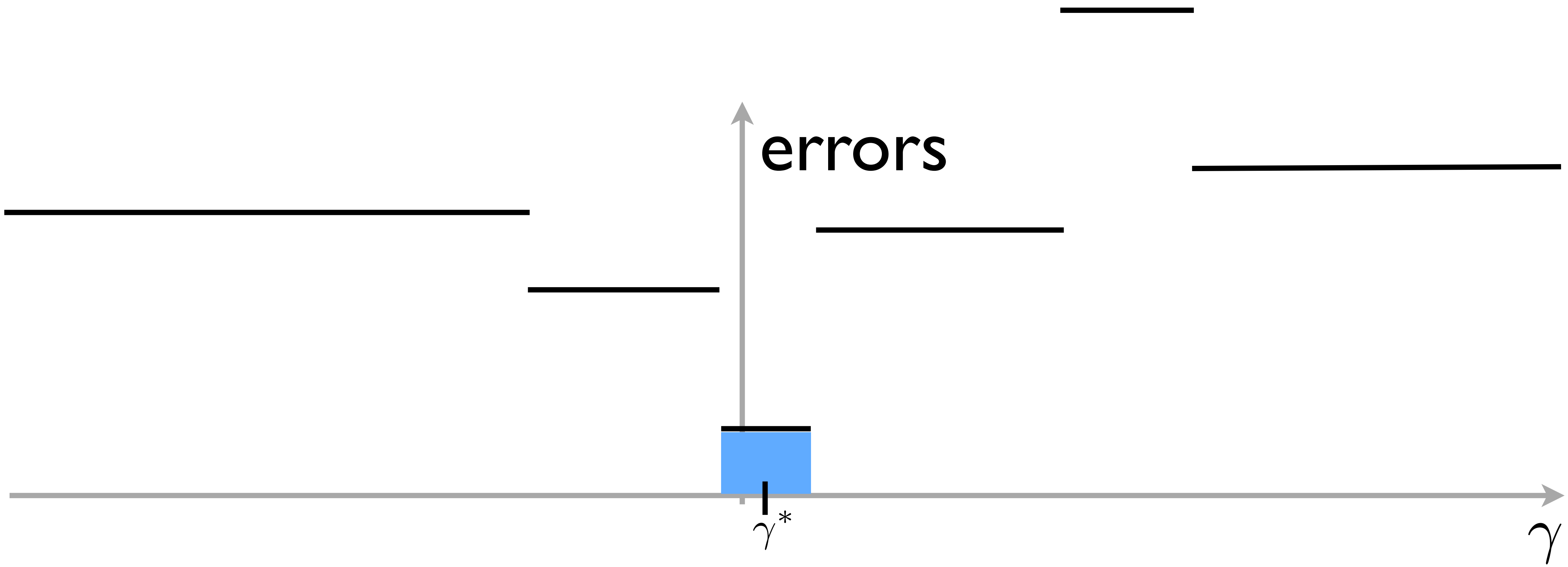

m
$\langle \mathbf{e}^*_{162}, \mathbf{a}^*_{162} \rangle$
$\langle \mathbf{e}^*_{28}, \mathbf{a}^*_{28} \rangle$
$\langle \mathbf{e}^*_{73}, \mathbf{a}^*_{73} \rangle$
γ
errors
γ

MERT



MERT





Let $\mathbf{w}_{\text{new}} = \gamma^* \mathbf{v} + \mathbf{w}$

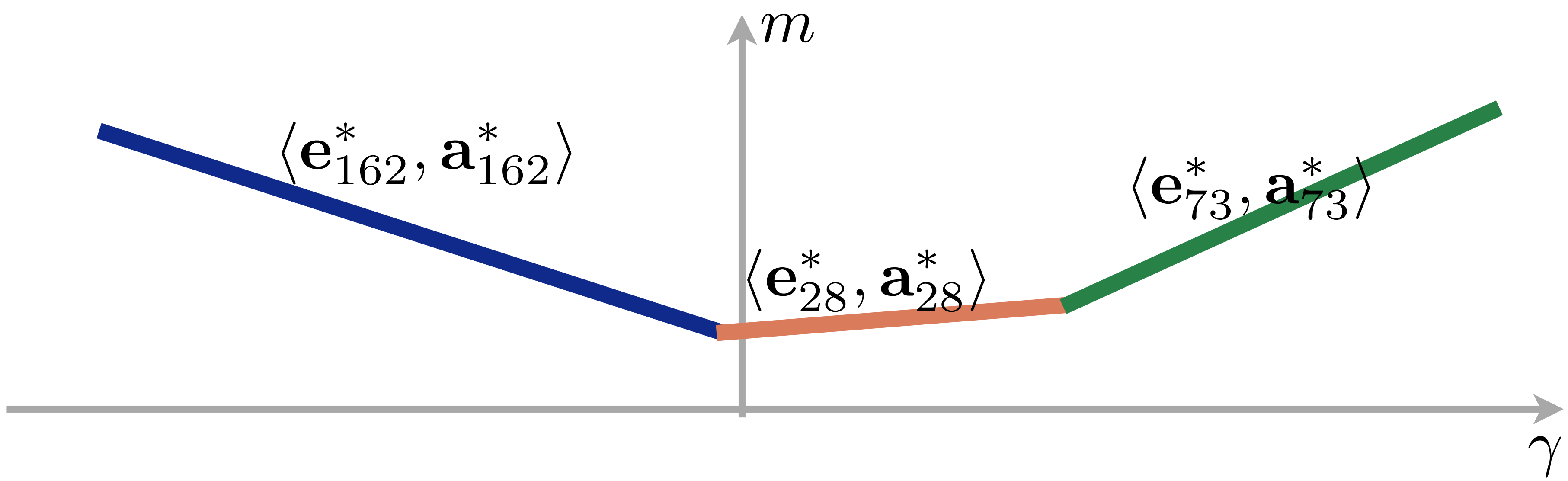
MERT

- In practice "errors" are sufficient statistics for evaluation metrics (e.g., BLEU)
  - Can maximize or minimize!
- Envelope can also be computed using dynamic programming
  - Interesting complexity bounds
- How do you pick the search direction?

Summary

- Evaluation metrics
  - Figure out how well we're doing
  - Figure out if a feature helps
  - But ALSO: train your system!
- What's a great way to improve translation?
  - **Improve evaluation**!

Thank You!



m
$\langle \mathbf{e}^*_{162}, \mathbf{a}^*_{162} \rangle$
$\langle \mathbf{e}^*_{28}, \mathbf{a}^*_{28} \rangle$
$\langle \mathbf{e}^*_{73}, \mathbf{a}^*_{73} \rangle$
γ