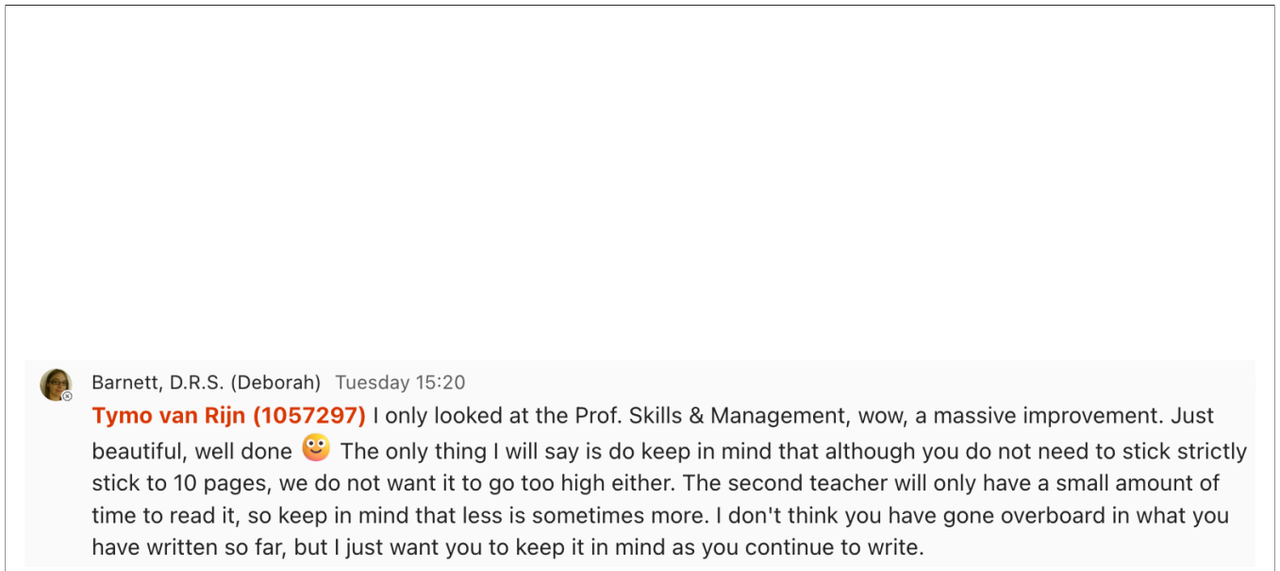
Week 5-6

Tymo’s progress

By Tymo!



School Supervisor happy , Tymo happy

Agenda

Morphed **F2V**

Tunnel **V**ision

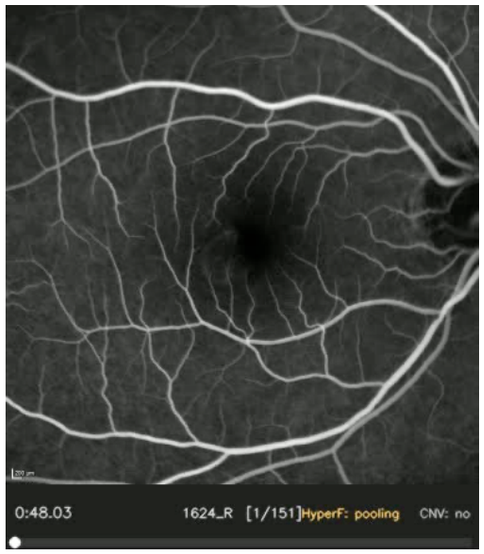
Linear **D**iscriminant **A**nalysis (**LDA**)

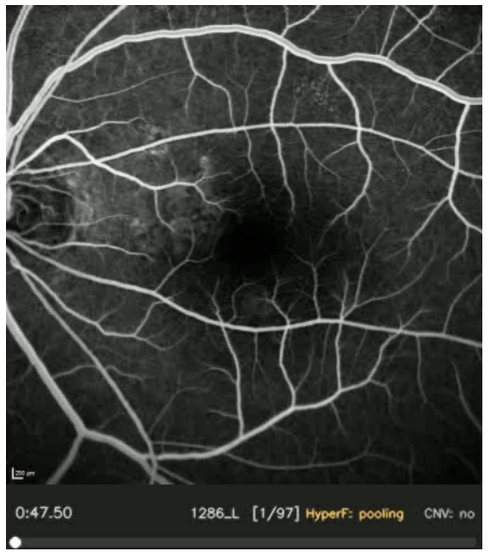
Temporal **S**eparability **B**enchmark (**TSB**)

Morphed F2V

Morphed F2V

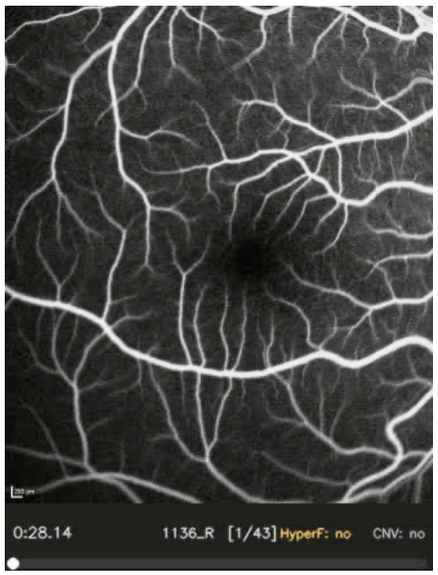
Pooling

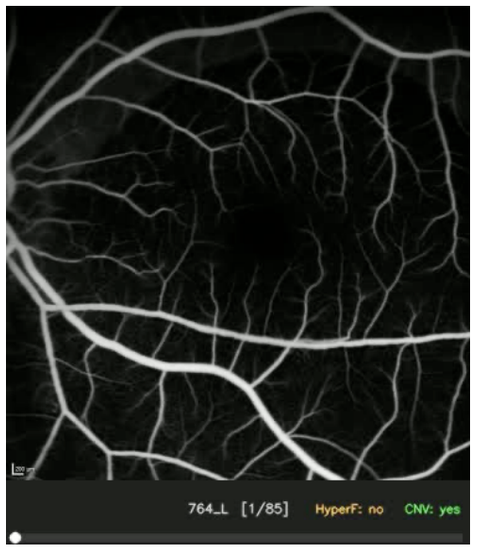




Morphed F2V

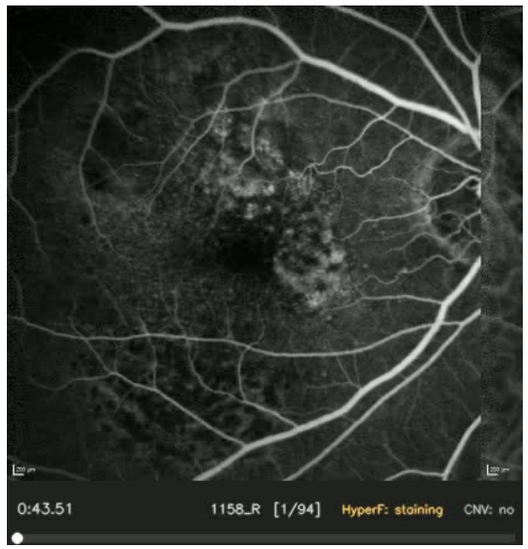
None





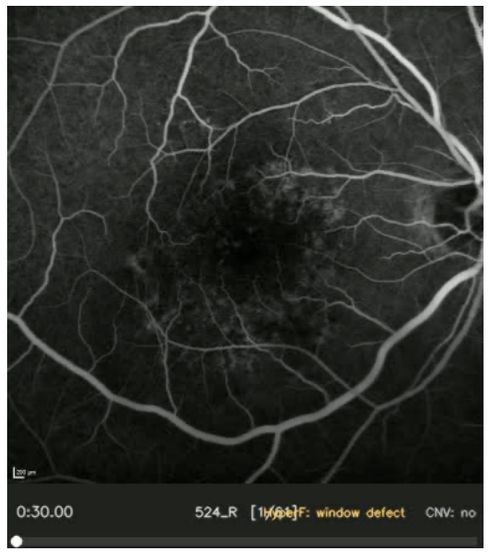
Morphed F2V

Staining



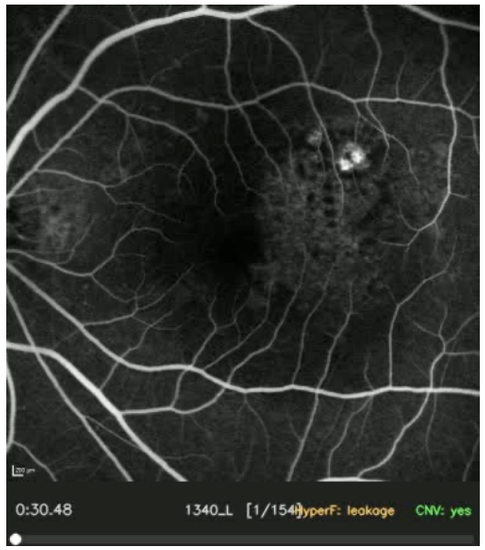
Morphed F2V

Window Defect



Morphed F2V

Leakage



Morphed F2V

How?

Farneback **O**ptical **F**low

(From the OpenCV library)

It answers:

“Where did this point in in **Frame A** move in **Frame B**?”

Optical Flow Interpolation:

Using that motion field to create new intermediate frames between two real frames (so synthetic?)

Tunnel Vision

Spread the focus

Tunnel Vision

Studies

Harnessing the power of longitudinal medical imaging for eye disease prognosis using **Transformer-based** sequence modelling (*2024, Holste et al.)*

Irregular

$$(x_1, t_1), (x_2, t_2), \ldots, (x_n, x_t)$$

Time to Vector

$$\tilde{z}_t = z_t + e_t$$

Tunnel Vision

Studies

Harnessing the power of longitudinal medical imaging for eye disease prognosis using **Transformer-based** sequence modelling (*2024, Holste et al.)*

Image -> ViT -> z_t

Time -> embedding -> e_t

Add -> $\tilde{z}_t$

Stack all timestamps -> Sequence

Feed into Transformer

Tunnel Vision

Studies

L-MAE: Longitudinal masked auto-encoder with time and severity-aware encoding for diabetic retinopathy progression prediction (*2025, Zeghlache et al.*)

Transformers assume equal spacing -> Wrong in medicine

Time encoding on patch level

Previous paper = extra feature

This paper = **position** in time

Tunnel Vision

Studies

L-MAE: Longitudinal masked auto-encoder with time and severity-aware encoding for diabetic retinopathy progression prediction (*2025, Zeghlache et al.*)

Reconstruction (with **MAE**) now depends on time

Progession patterns matter

Gives better temporal structure

Tunnel Vision

Key point

Elapsed time encoding

Not just order (binning)

Time-embedding approach

Linear Discriminant Analysis (LDA)

Just like PCA, this was also very new to me, since like I said, I have not background in statistics or data science whatsoever.
So instead of kaggling my way up like last time, I decided to actually spend a lot of time trying to gain some intuition on this concept.

Which again, was a bit hard sometimes, since it required some mathematical knowledge that I had not yet obtained. But nevertheless, it felt great actually trying to gain this knowledge and get feeling for how this actually works under the hood, instead of just implementing it using sckit library.

LDA

What is it?

"LDA works by **finding directions** in the feature space that best **separate** the classes.

It does this by **maximizing** the difference between the class means while **minimizing** the spread within each class."

Im pretty sure you know how LDA works, so I don't think it's worth it to go over all of this again.
But like I talked about last meeting, I had some trouble proving that I am spending time studying different concepts, which need to be apparent in my final document.
So I made these slides to give a clearer picture as of how LDA works in general, to give myself a good thinking exercise and to show that I really took the time to try and understand it profoundly.

LDA

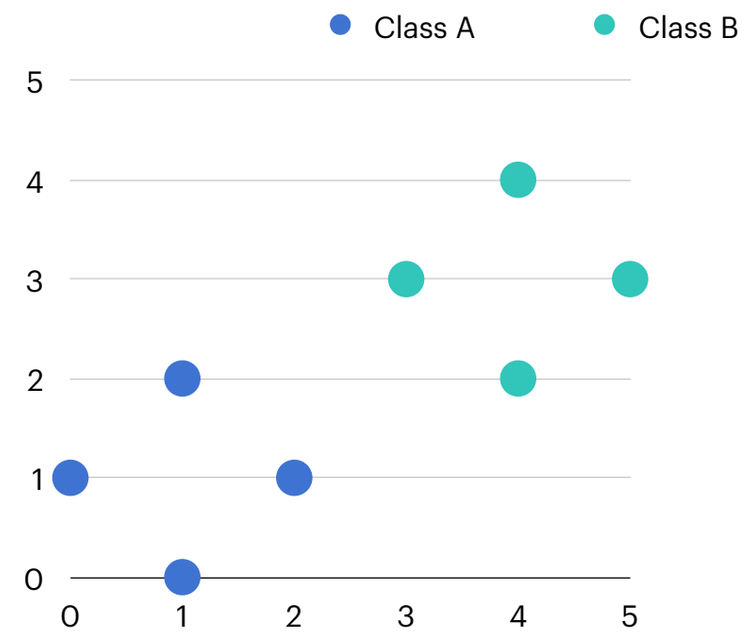
Worked out 2D example

Class A

(0,1) (1,0) (1,2) (2,1)

Class B

(3,3) (4,2) (4,4) (5,3)



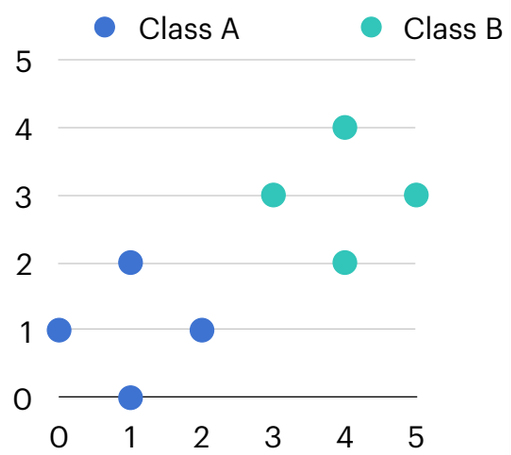
LDA

Worked out 2D example

1. Compute the class means (μ)

$$\mu_A = \frac{1}{4}((0,1) + (1,0) + (1,2) + (2,1)) = (1,1)$$

$$\mu_B = \frac{1}{4}((3,3) + (4,2) + (4,4) + (5,3)) = (2\frac{1}{2},2)$$



LDA

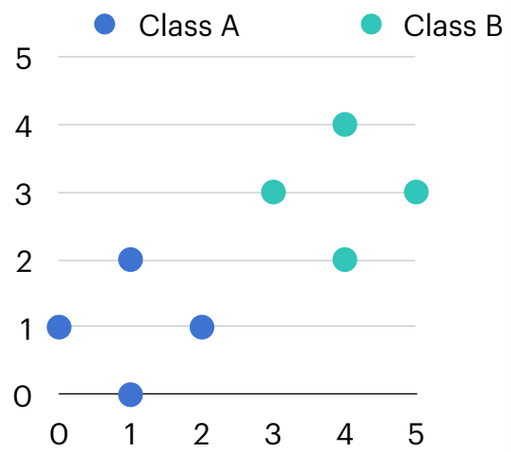
Worked out 2D example

2. Compute the within-class scatter (S_w)

$$S_w = S_A + S_B$$

$$S_A = \sum_{x \in C_A} (x - \mu_A)(x - \mu_A)^T$$

$$S_B = \sum_{x \in C_B} (x - \mu_B)(x - \mu_B)^T$$



I’ve got to say, I now know that what is presented on the screen is actually just a simple calculation, but when I showed this to my parents they thought I was going to be a rocket scientist after this internship.

I think it just goes to show that what I was talking about earlier, that some equations in study papers scared me away, because I don’t see myself as someone who could understand those. I think if I take the time to try and understand what the equations are trying to say, it’s actually not that big of a deal

LDA

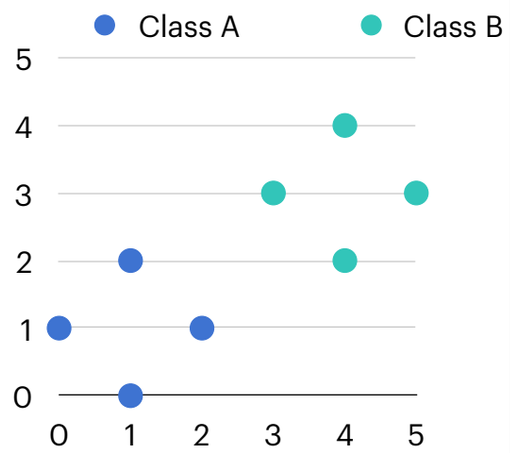
Worked out 2D example

2. Compute the within-class scatter

$$S_A = \sum_{x \in C_A} (x - \mu_A)(x - \mu_A)^T$$

Example for datapoint (0,1):

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mu_A = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



LDA

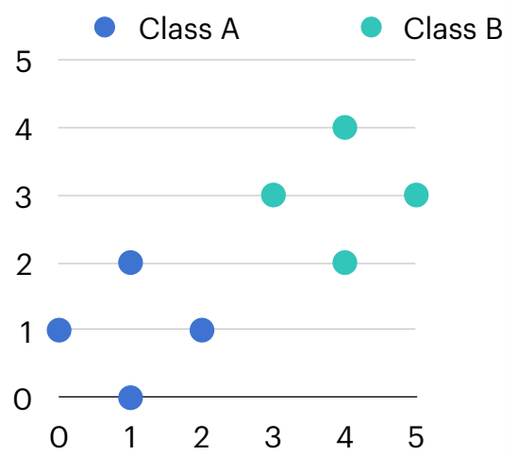
Worked out 2D example

2. Compute the within-class scatter

$$S_A = \sum_{x \in C_A} (x - \mu_A)(x - \mu_A)^T$$

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mu_A = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$(x - \mu_A)(x - \mu_A)^T = \begin{bmatrix} 0 \\ -1 \end{bmatrix} [0 \quad -1] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$



LDA

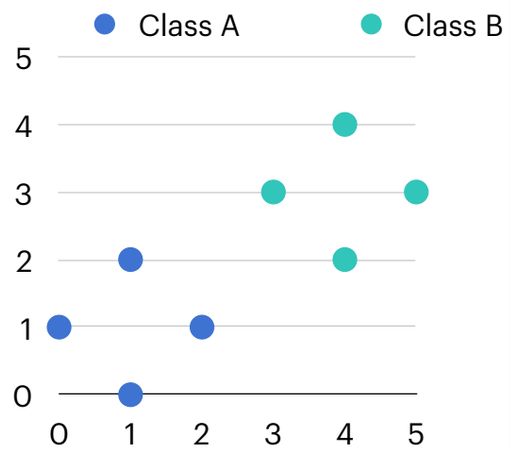
Worked out 2D example

2. Compute the within-class scatter

$$S_A = \sum_{x \in C_A} (x - \mu_A)(x - \mu_A)^T$$

Once done for every x,

$$S_A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



LDA

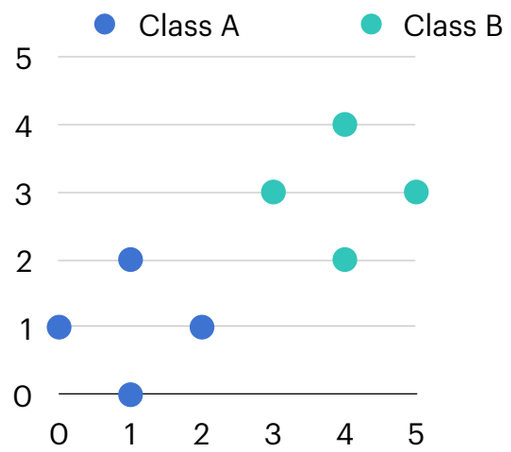
Worked out 2D example

2. Compute the within-class scatter

$$S_B = \sum_{x \in C_A} (x - \mu_B)(x - \mu_B)^T$$

Once done for every x,

$$S_B = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$



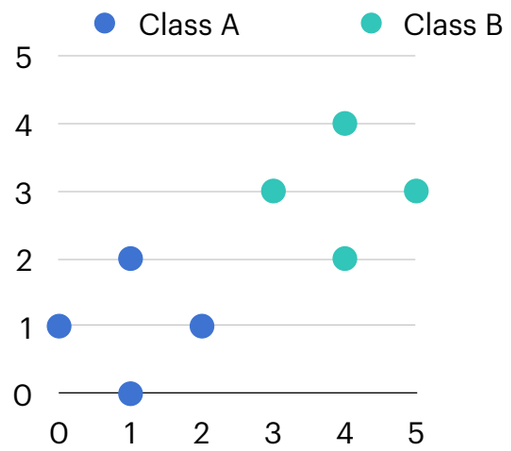
LDA

Worked out 2D example

2. Compute the within-class scatter

$$S_w = S_A + S_B$$

$$S_w = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$



LDA

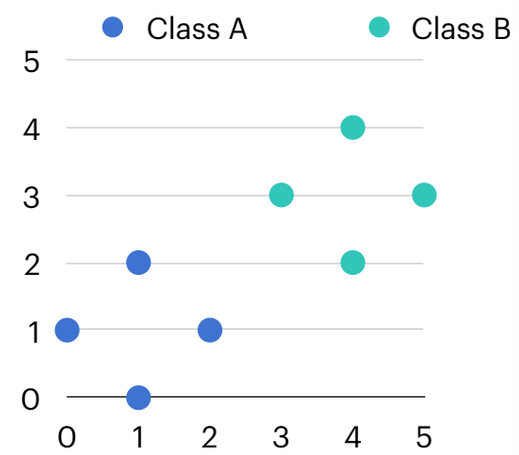
Worked out 2D example

3. Compute the between-class scatter (S_b)

$$S_b = \sum_{C \in A,B} n_c(\mu_c - \mu)(\mu_c - \mu)^T$$

$$n_A = n_B = 4$$

μ = Overall mean

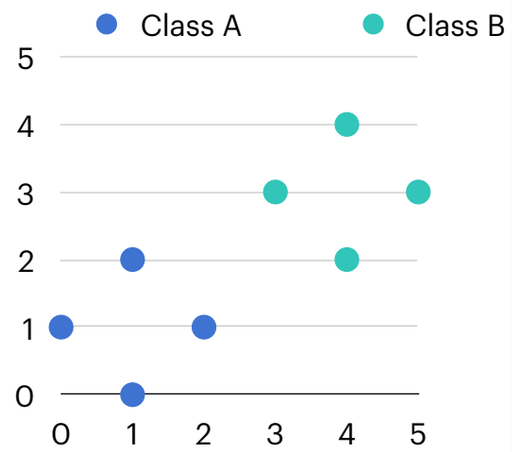


LDA

Worked out 2D example

3. Compute the between-class scatter (S_b)

$$S_b = \begin{bmatrix} 18 & 12 \\ 12 & 8 \end{bmatrix}$$



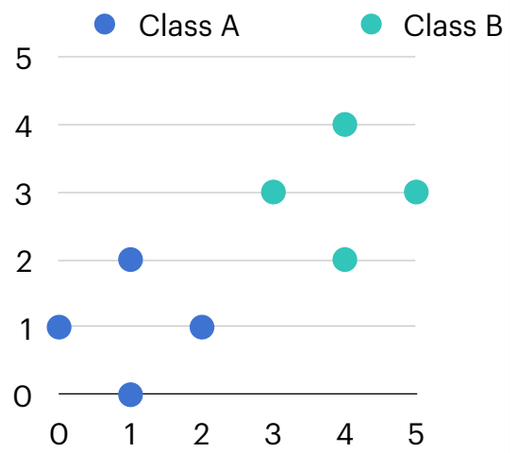
LDA

Worked out 2D example

4. Compute $S_w^{-1} S_b$

Since S_w is $\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$, its inverse is

$$S_w^{-1} = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix}$$



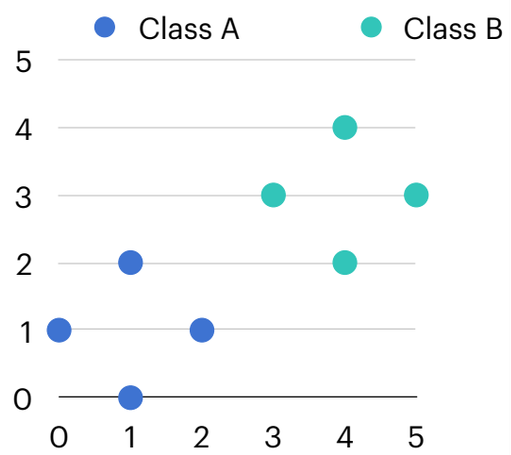
LDA

Worked out 2D example

4. Compute $S_w^{-1} S_b$

The multiply with S_b

$$\begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 18 & 12 \\ 12 & 8 \end{bmatrix} = \begin{bmatrix} \frac{9}{2} & 3 \\ 3 & 2 \end{bmatrix}$$



LDA

Worked out 2D example

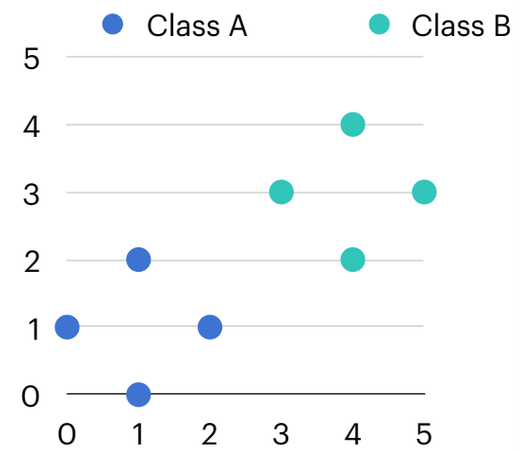
5. Find the eigenvectors

$$\text{Solve} \begin{bmatrix} \frac{9}{2} & 3 \\ 3 & 2 \end{bmatrix} w = \lambda w$$

$$\lambda_1 = 6.5$$

$$\lambda_2 = 0$$

$$\left(\begin{bmatrix} \frac{9}{2} & 3 \\ 3 & 2 \end{bmatrix} - 6.5I \right) w = 0$$



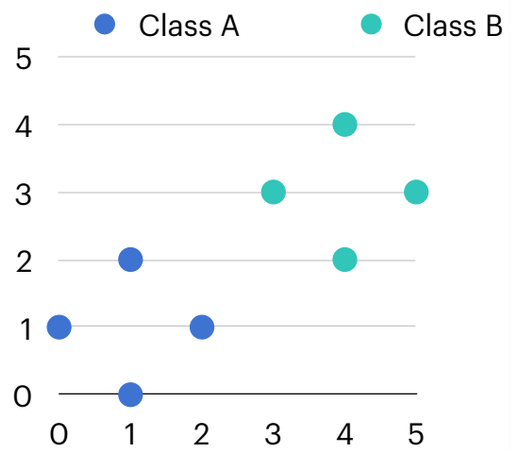
LDA

Worked out 2D example

5. Find the eigenvectors

$$\left(\begin{bmatrix} \frac{9}{2} & 3 \\ 3 & 2 \end{bmatrix} - 6.5I\right)w = 0$$

$$w = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$



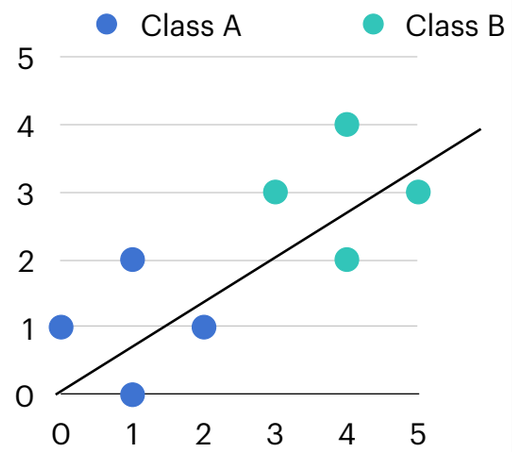
LDA

Worked out 2D example

5. Find the eigenvectors

$$\left(\begin{bmatrix} \frac{9}{2} & 3 \\ 3 & 2 \end{bmatrix} - 6.5I\right)w = 0$$

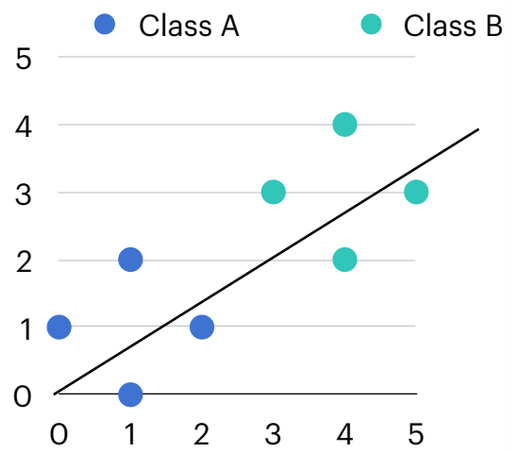
$$w = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$



LDA

Worked out 2D example

6. Project the datapoints on the line, using Scalar Projection



Temporal Separability Benchmark (*TSB*)

TSB

Motivation

Temporal sequence models are most useful when the embedding space encodes meaningful **temporal structure.**

Choosing the right model and temporal binning strategy is currently done by **expensive trial-and-error**.

A lightweight diagnostic tool that measures *how well* embeddings separate temporal states before training could substantially reduce this cost and provide **principled, explainable** model selection.

TSB

Research Question

*"Does temporal <u>separability</u> in an embedding space, measured via a **linear discriminant** probe, predict the <u>downstream performance</u> of a temporal sequence model trained on that same embedding space?"*

TSB

Hypotheses

H1:

*If temporal separability measure by an **LDA** probe is a valid proxy for temporally useful structure in an embedding space, then embedding/binning configurations with higher separability scores will achieve better **downstream performance** when the same temporal model is trained on them*

TSB

Hypotheses

H2:

*If foundation models differ in how well they **encode temporal structure**, then the separability scores produced by the **LDA** probe will differ systematically across foundation models under comparable binning settings*

TSB

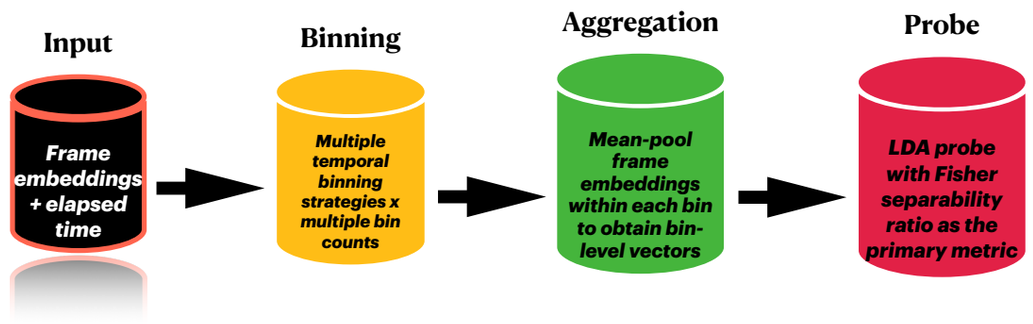
Hypotheses

H3:

If temporal binning influences how strongly temporal structure is presented independently of the embedding model, then varying the binning strategy or bin count while holding the embedding model fixed will lead to ***measurable changes*** *in separability scores*

TSB

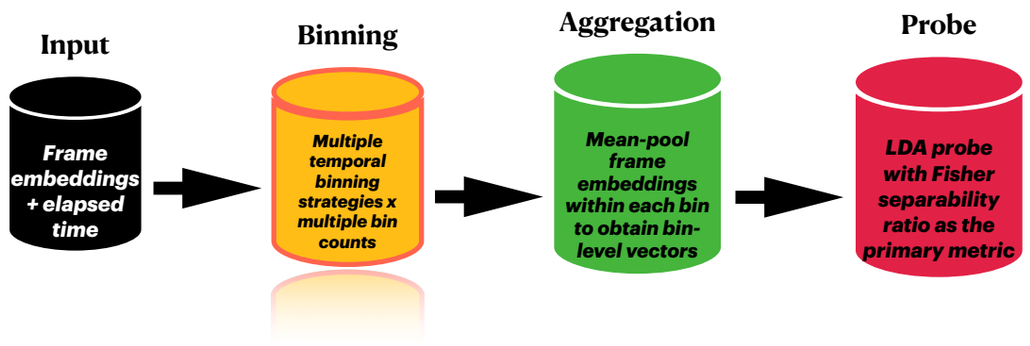
Method Sketch



Each exam yields a sequence of frame embeddings ($x_t \in \mathbb{R}^d$) with associated elapsed times (τ_t)

TSB

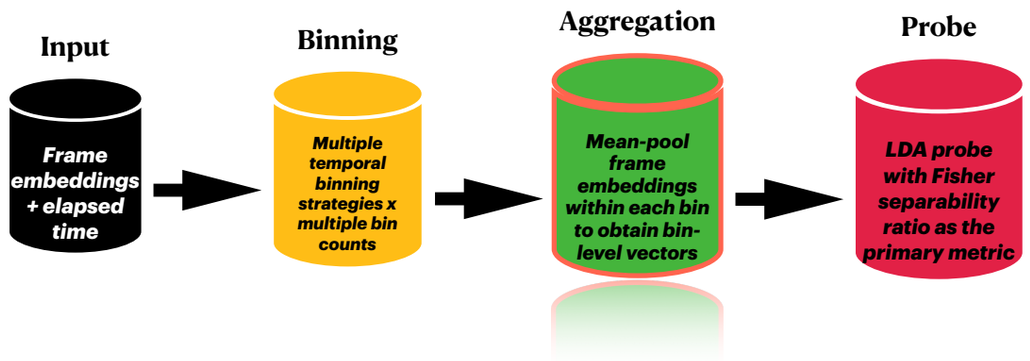
Method Sketch



Frames are assigned to bins based on (τ_t) or order. We test multiple strategies (uniform, quantile, last + first only) and multiple bin counts (B).

TSB

Method Sketch



Frame-embeddings within each bin are mean-pooled:

Let S_b *be the set of frame embeddings assigned to bin* (b)

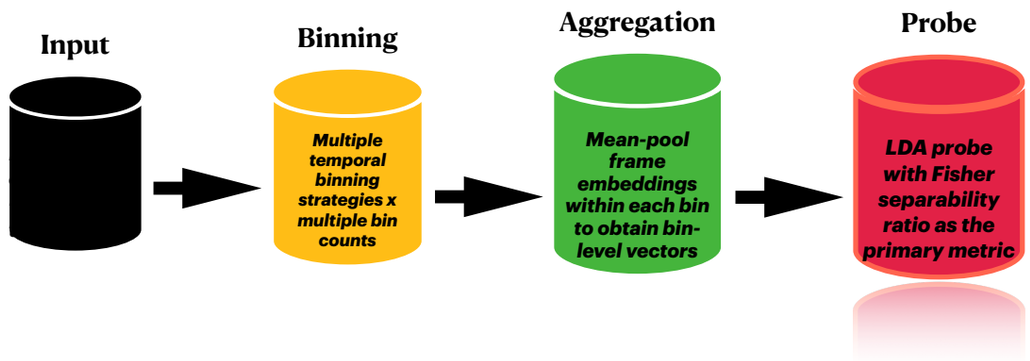
$$z_b = \frac{1}{S_b} \sum_{x_t \in S_b} x_t$$

Each exams becomes an order sequence $(z_1, \ldots, z_B)$

So lets say there are 3 bins, with 30 frames in an examination. That would mean every bin would have 10 frames, these 10 frames would then be mean pooled. Which works like shown here

TSB

Method Sketch



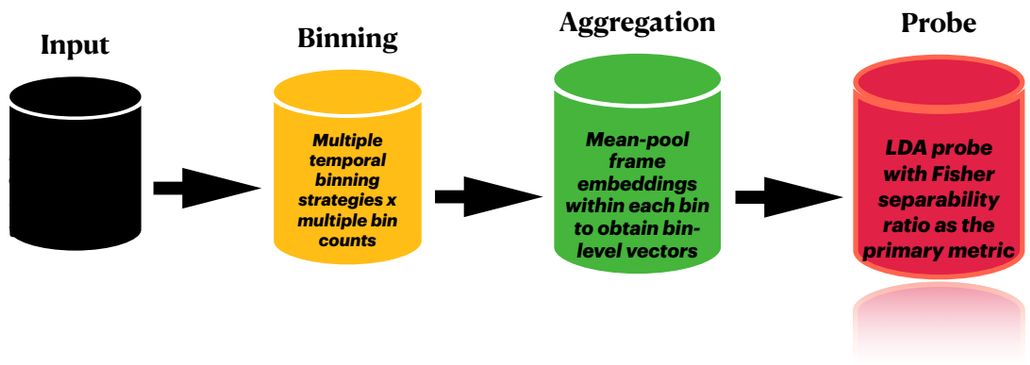
LDA is fit on all bin-level vectors across exams, using ***bin index*** *as a class label. The* ***Fisher ratio***

$$(\mathcal{F} = \frac{tr(S_B)}{tr(S_W)})$$

ranks configurations by temporal separability

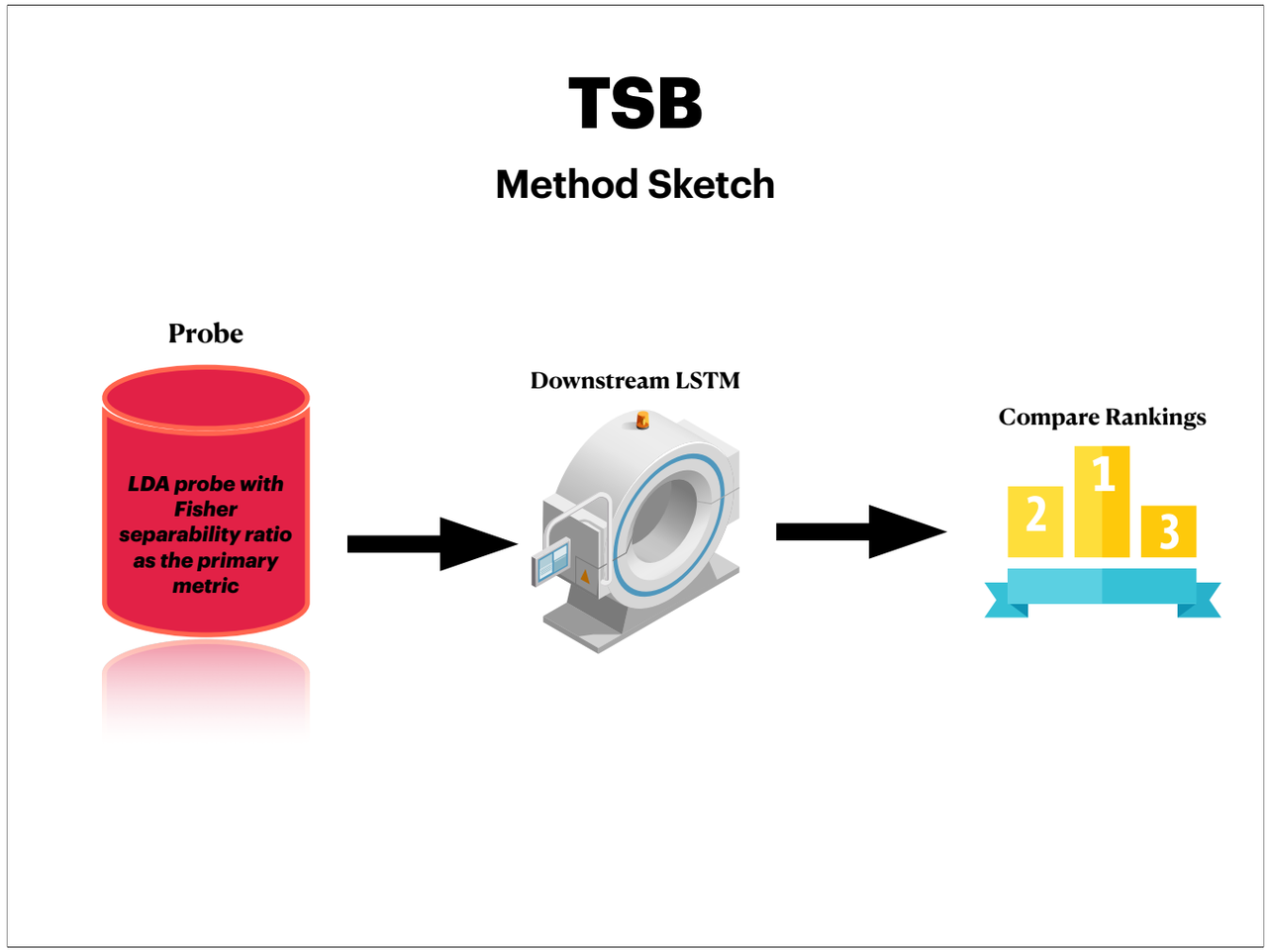
TSB

Method Sketch



After some thought, I don't think taking the mean-pool frame embeddings is the correct way to work

Picking the middle frame in such a bin would create better separation.



So after the probe, the same configuration is used to let it go through an LSTM with the HyperF_Type classification task. The metrics used are ROC-AUC and AP.

So then we have those 2 metrics, and the fisher separability ratio. We use the spearmans rho to see if there exists a correlation between those 2 rankings. Which then should educate us about H1

TSB

Data Protocol + “Fairness”

Train / Val / Test **split** (by hyperftype.json)

n_test = 211

Embeddings are precomputed for train/val/test. Probe uses train only.
Downstream uses train+val+test normally.
And importantly, all configs are evaluated with the same test size: 211 sequences

TSB

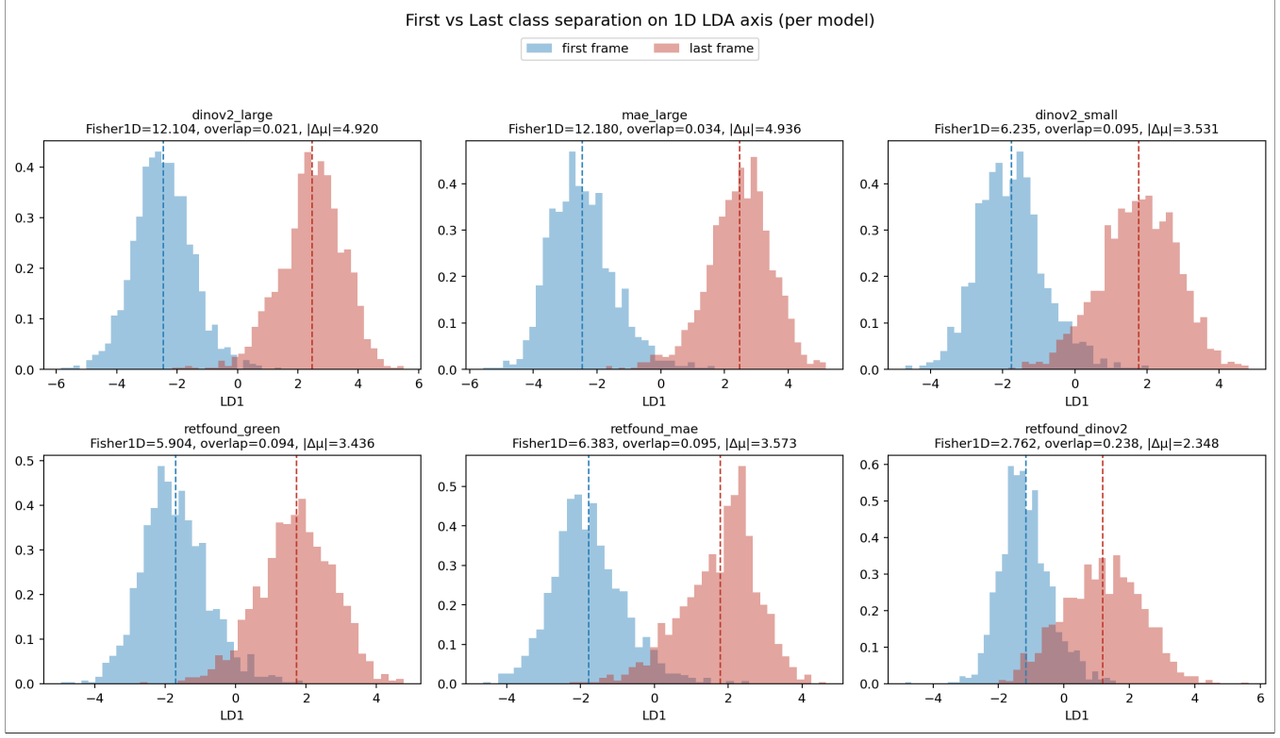
(6) Models used

General Foundation model	Domain-specific foundation model
Dinov2 Large (1024D)	RETFound Dinov2 (1024D)
Mae Large (1024D)	RETFound MAE (1024D)
DinoV2 Small (384D)	RETFound Green (384D)

Embedding dimensionality differs across backbones (384 vs 1024), so we treat within-model configuration ranking as the primary inference and cross-model Fisher magnitude as secondary context.

TSB

Sanity check



Before the full benchmark, I checked whether temporal signal exists. Most models separate early vs late reasonably well. So there is temporal structure worth studying.

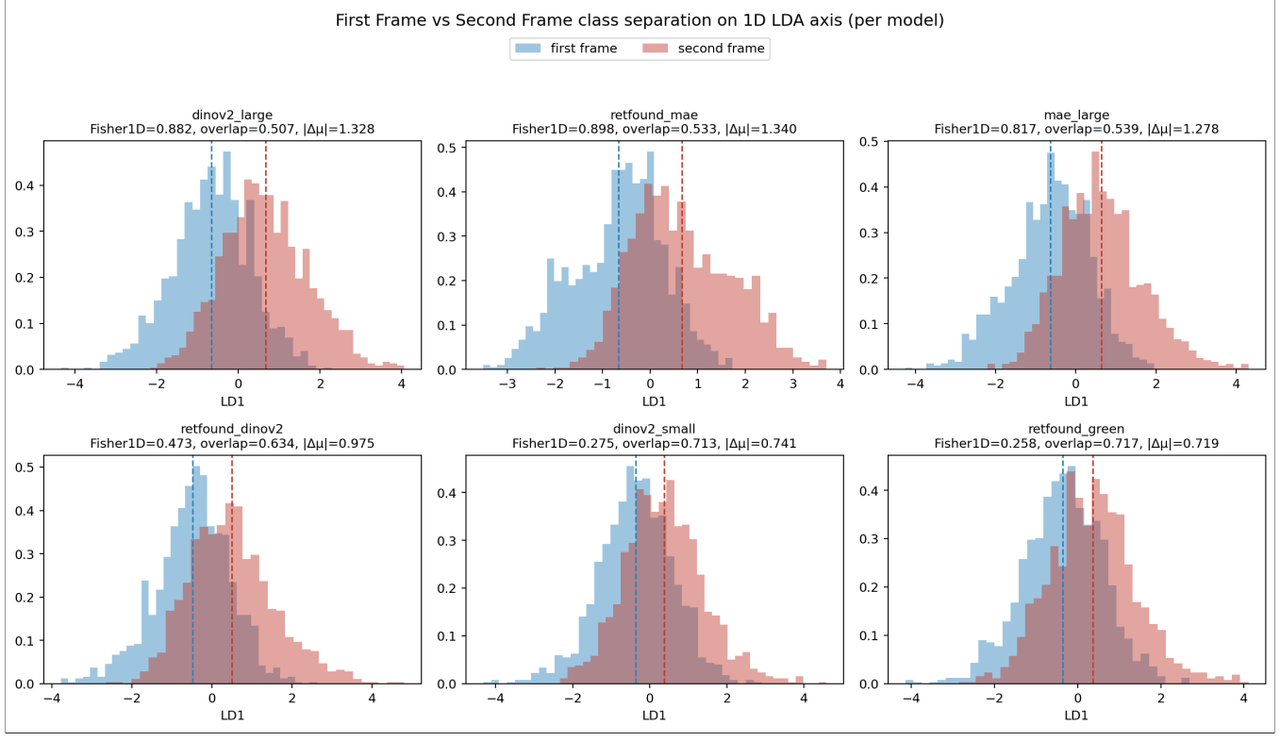
The things now is, if I were to put this exact configuration through the LSTM, and compare them across models, it would make zero sense to me, sense the output of the LSTM is way more dependent on the pretrained data than on the temporality. So that is why I decided to make it a within-model experiment.

I do think this good be of good interest for us, to see which binning strategy could actually lead us to the most significant results. By using these “probes”, we could afterwards train the model on the specific best performing model with the best performing probed binning strategy.

this is strong evidence of separability, not temporal modeling.

TSB

Sanity check



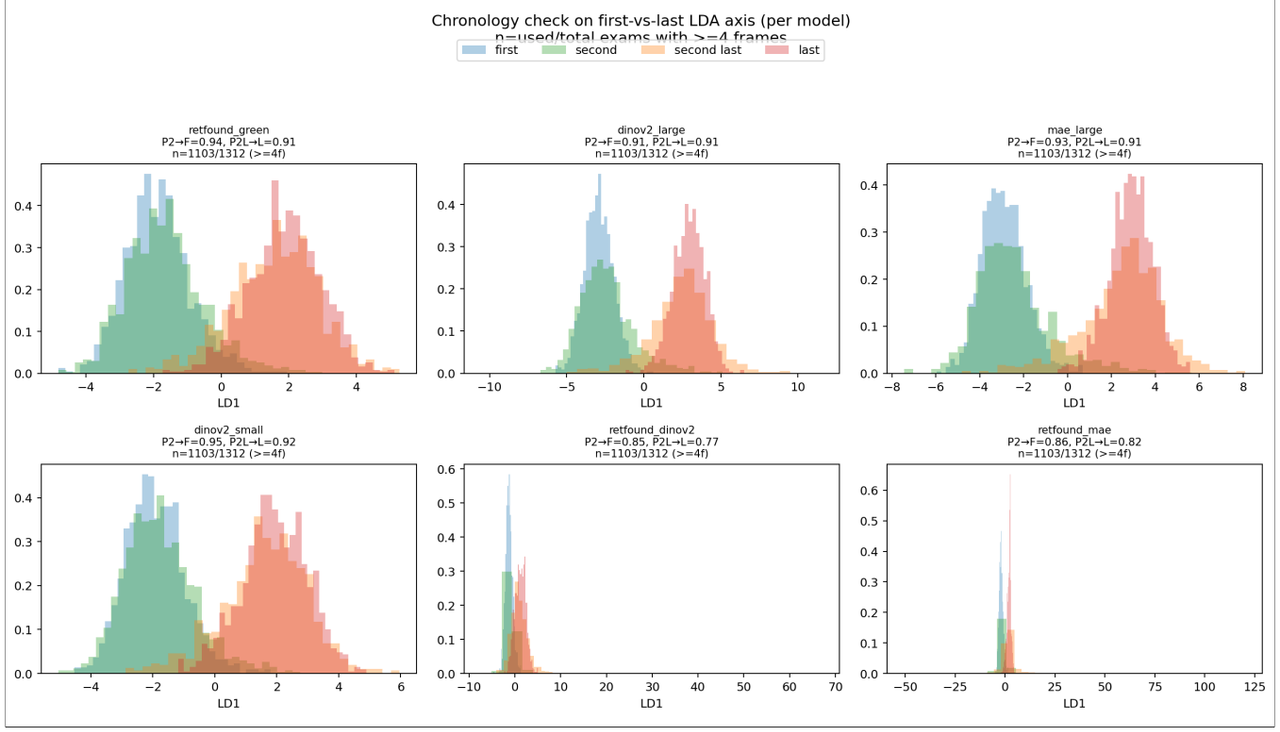
To make sure this separation was not just luck, based on Oscar’s advice, I made another experiment, to see if it would separate the first vs second frame just as well, which would then mean, my results from the previous slide wouldn’t mean much.

The separability increases with temporal distance.

Turns out, the separation between first and second frame scored a lot worse than the first vs last, which goes to show it was not just random luck

TSB

Sanity check



Then I tried to see if there was some actual chronological order.

The LDA direction trained on (first vs last) acts as a monotonic temporal axis.

When you project embeddings onto this axis:

Early frames → low values

Late frames → high values

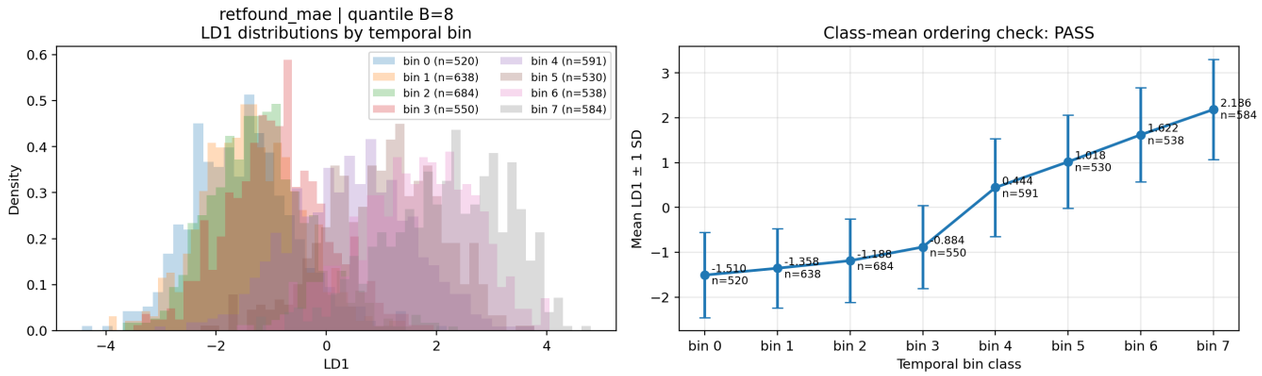
Intermediate frames → lie in between

There exists a 1D projection of the embedding space that is correlated with time progression.

TSB

Sanity check

Quick visual check: quantile B=8 class means on LD1



Then I tried to see if there was some actual chronological order.

The LDA direction trained on (first vs last) acts as a monotonic temporal axis.

When you project embeddings onto this axis:

Early frames → low values

Late frames → high values

Intermediate frames → lie in between

There exists a 1D projection of the embedding space that is correlated with time progression.

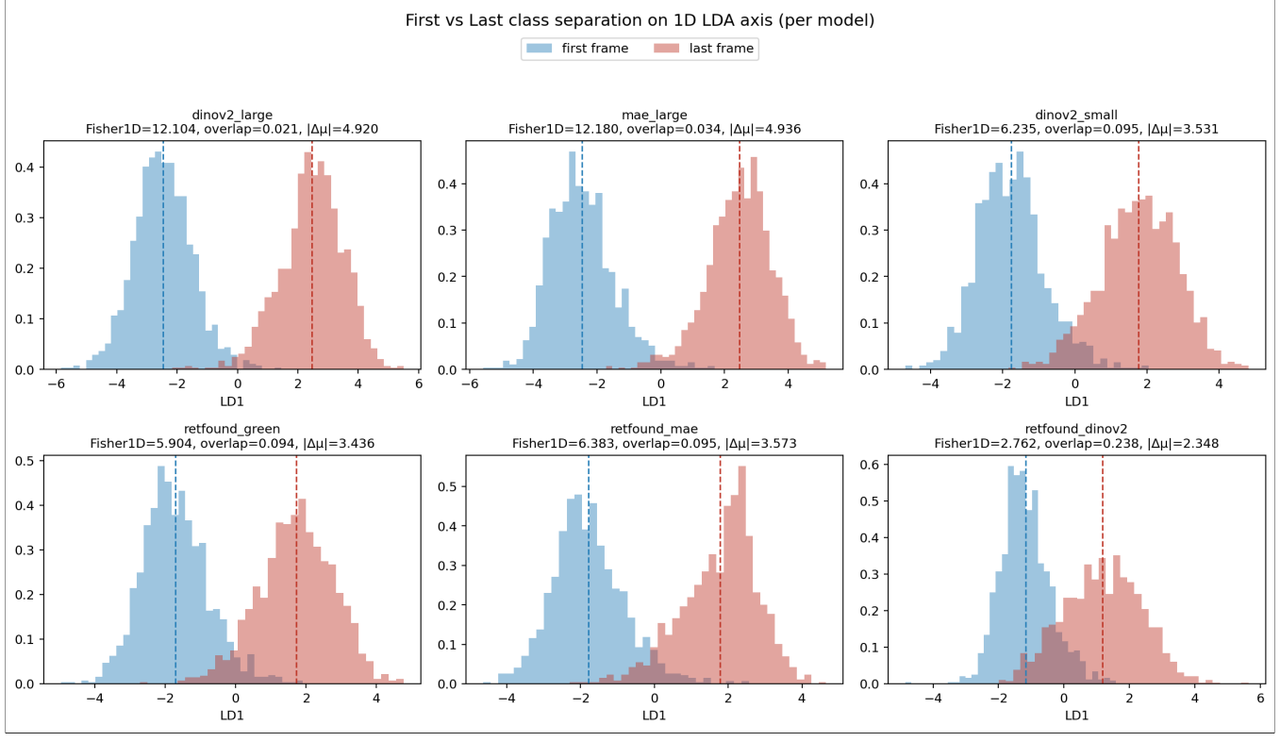
TSB

Sanity check

The embeddings are **not** temporally random.
They contain a approximately monotonic structure aligned with progression from early to late frames.

TSB

Continuing from these results



Before the full benchmark, I checked whether temporal signal exists. Most models separate early vs late reasonably well. So there is temporal structure worth studying.

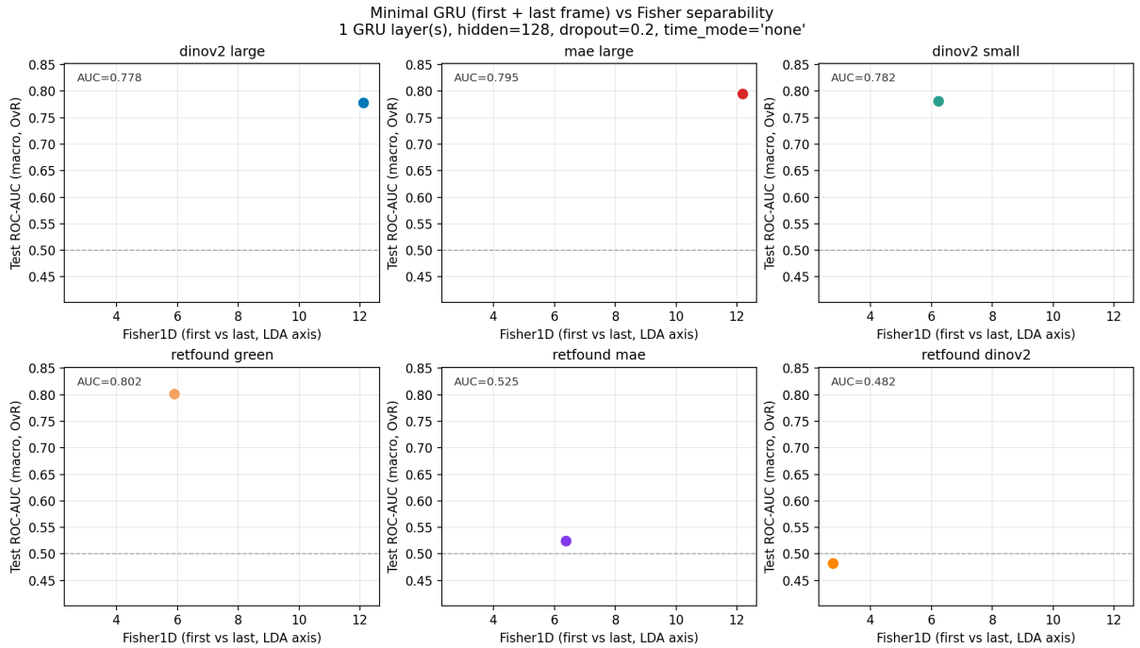
The things now is, if I were to put this exact configuration through the LSTM, and compare them across models, it would make zero sense to me, sense the output of the LSTM is way more dependent on the pretrained data than on the temporality. So that is why I decided to make it a within-model experiment.

I do think this good be of good interest for us, to see which binning strategy could actually lead us to the most significant results. By using these “probes”, we could afterwards train the model on the specific best performing model with the best performing probed binning strategy.

this is strong evidence of separability, not temporal modeling.

TSB

Continuing from these results



Before the full benchmark, I checked whether temporal signal exists. Most models separate early vs late reasonably well. So there is temporal structure worth studying.

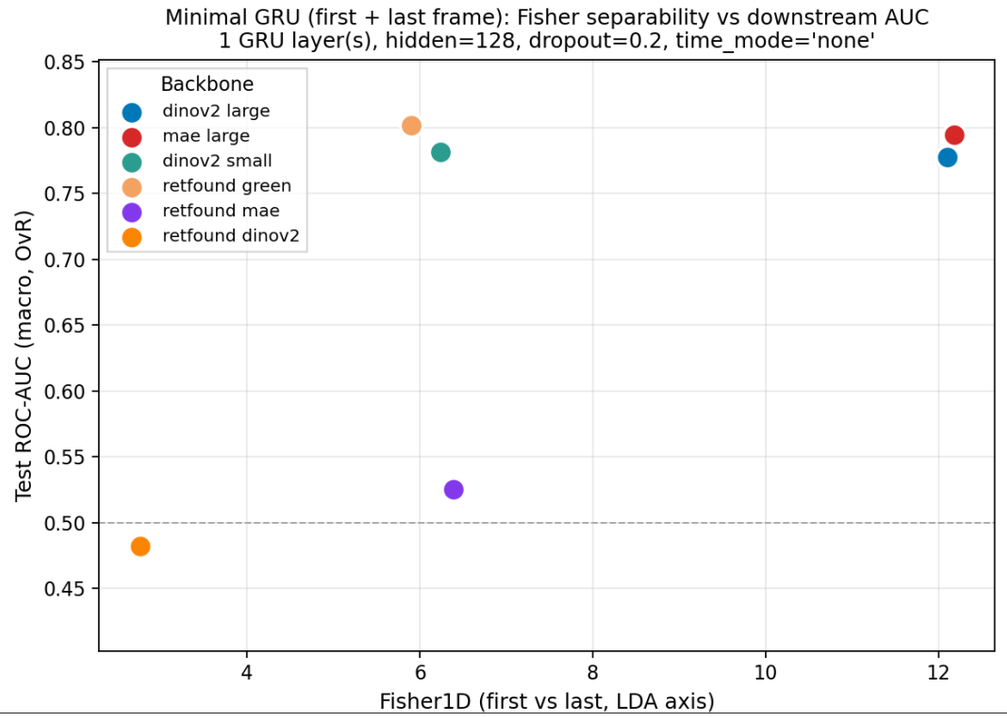
The things now is, if I were to put this exact configuration through the LSTM, and compare them across models, it would make zero sense to me, sense the output of the LSTM is way more dependent on the pretrained data than on the temporality. So that is why I decided to make it a within-model experiment.

I do think this good be of good interest for us, to see which binning strategy could actually lead us to the most significant results. By using these “probes”, we could afterwards train the model on the specific best performing model with the best performing probed binning strategy.

this is strong evidence of separability, not temporal modeling.

TSB

Continuing from these results



Before the full benchmark, I checked whether temporal signal exists. Most models separate early vs late reasonably well. So there is temporal structure worth studying.

The things now is, if I were to put this exact configuration through the LSTM, and compare them across models, it would make zero sense to me, sense the output of the LSTM is way more dependent on the pretrained data than on the temporality. So that is why I decided to make it a within-model experiment.

I do think this good be of good interest for us, to see which binning strategy could actually lead us to the most significant results. By using these “probes”, we could afterwards train the model on the specific best performing model with the best performing probed binning strategy.

this is strong evidence of separability, not temporal modeling.