CANCER RECURRENCE PREDICTION

A Multimodal Deep Learning Approach

HARVARD HEALTH DATA SCIENCE GUEST LECTURE

Shashank Yadav, Ph.D., & Andrew Foong, Ph.D.

May 11th, 2026

MAYO CLINIC

Radiation Oncology
AI & Data Analytics
AIDA

Overview

1. Head & neck cancer
   - *Project overview & clinical motivation*
2. Survival analysis
   - *Modeling time-to-event data*
3. Maximum likelihood learning
   - *The objective function*
4. Discrete time survival models
   - *Beyond proportional hazards*
5. Integrating pathology
   - *From gigapixel images to risk scores*
6. Open Questions

Head & Neck Cancer

Project overview & clinical motivation

Goals

- HPV-positive oropharyngeal carcinoma:
  - A kind of head and neck cancer
  - *Oropharynx* is the middle part of the throat
- Clinical motivation:
  - Incidence is rapidly rising
  - Highly curable, *but*:
    - Treatments usually aggressive
    - Cause many side-effects
  - Would be great to know which patients are at higher risk of cancer recurrence
    - Higher-risk patients treated more aggressively
    - Lower-risk patients spared side-effects
- Idea: use AI to predict higher-risk patients

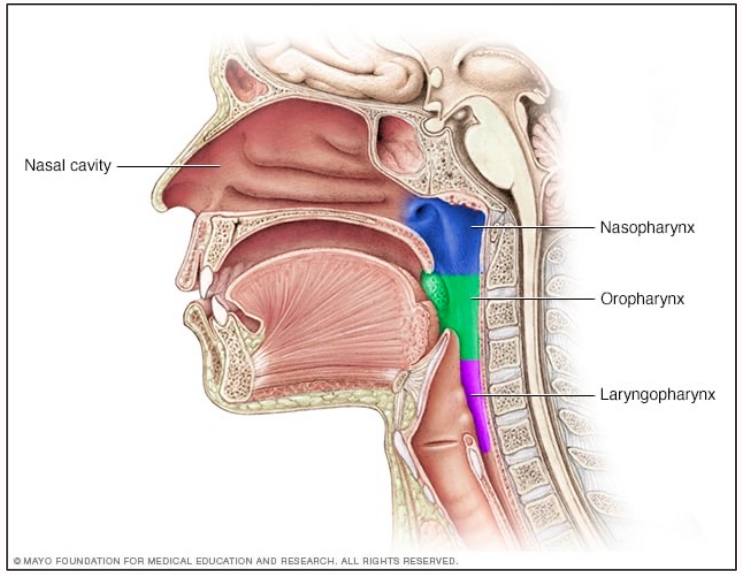


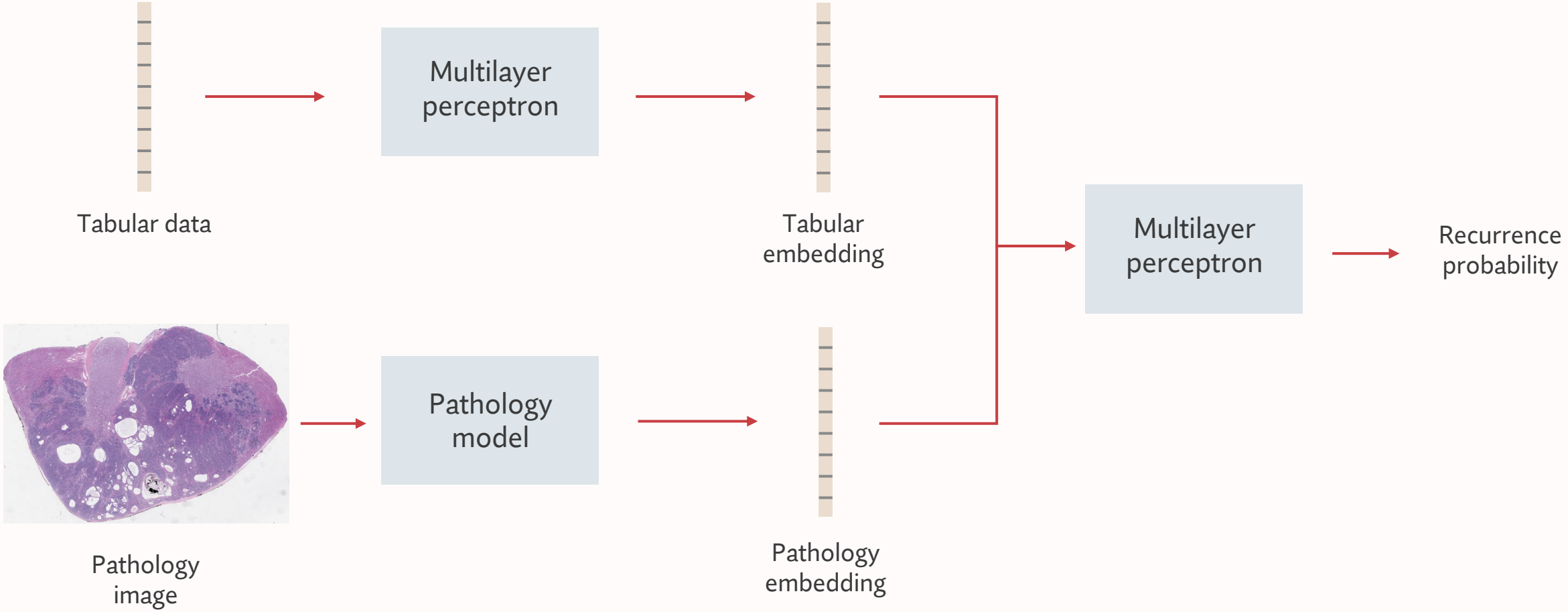
Image from mayoclinic.org

Strategy

- Dataset:
  - Approx. 1500 HPV+ oropharyngeal cancer patients at Mayo Clinic
  - Followed up longitudinally for varying time periods
  - Recurrences recorded
- Algorithm: *Multimodal deep learning*
  - There are many potentially relevant sources of information:
    - *"Tabular" data*:
      - Age – real number
      - Tumor stage – categorical
      - Has the cancer spread along nerves? – categorical
      - Etc.
    - *Pathology slides*:
      - Scan of a slice of the tumor – gigapixel image
  - How to merge these data sources into one model?

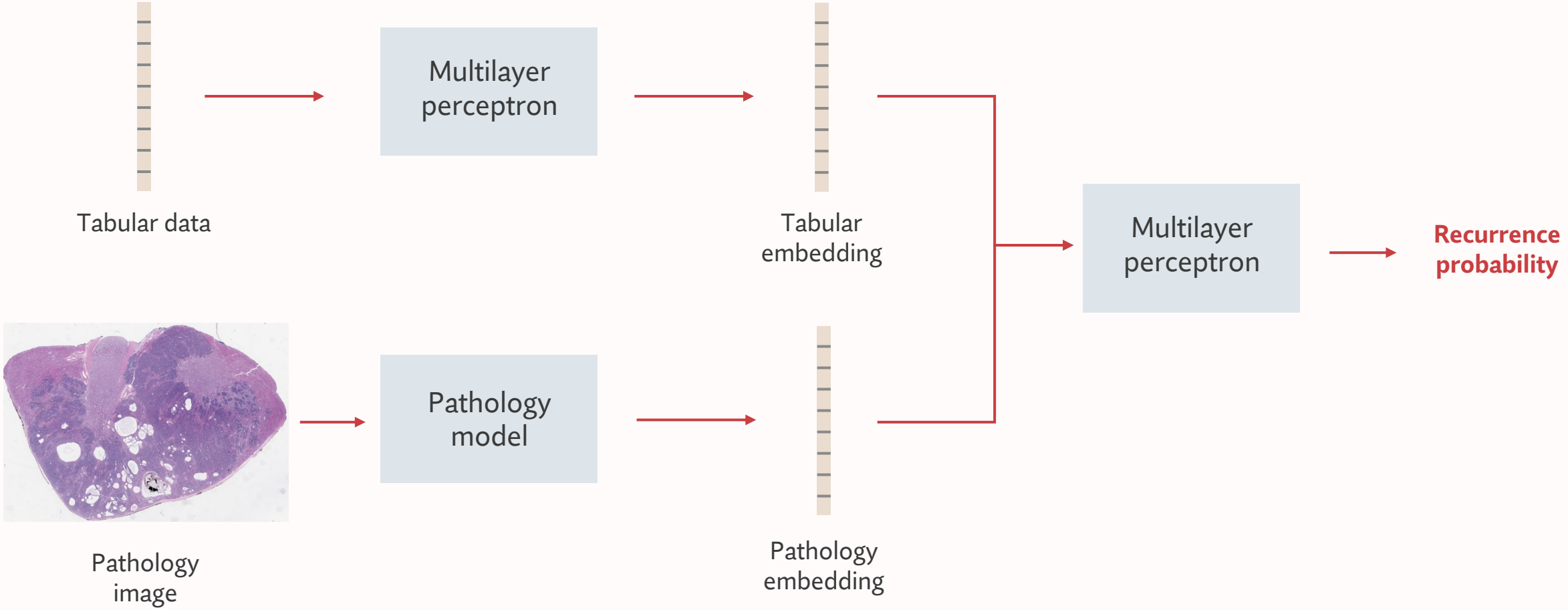
Multimodal fusion

1. Embed each modality with a neural network
2. Combine the embeddings



Multimodal fusion

1. Embed each modality with a neural network
2. Combine the embeddings



Survival analysis

Modeling time-to-event data

Problem statement

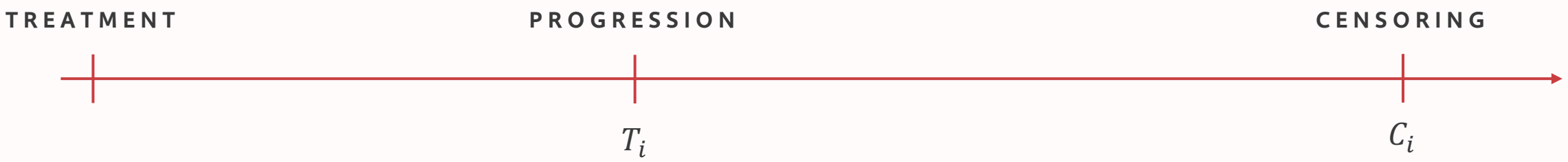
- Define an event of interest: cancer recurrence/progression
- We want to predict *who* will experience the event, and *when*
- *Problem*: for many patients, *we don't observe the event of interest*.
  - Patients drop out ("lost to follow-up")
  - The study ends (e.g., *today*), and we don't know what will happen to patients in the future
- This called censoring, and is ubiquitous in healthcare datasets
- Analyzing this kind of data is the central problem of *survival analysis*

Mathematical description

- Let T_i be the time between treatment and the event of interest (cancer progression/recurrence) for patient i
- Let C_i be the time at which we *stop* following the patient
  - *This is the* censoring *time*
- If $C_i > T_i$, then we observe the patient long enough to witness the event.
- If $C_i < T_i$, then we don't observe the event – all we know is that the event has *not* yet occurred by time C_i.
- We denote whether we observe the event by another variable, δ_i
  - $\delta_i = 1$ if we observe the event: $C_i > T_i$
  - $\delta_i = 0$ if we don't observe the event: $C_i < T_i$

Mathematical description

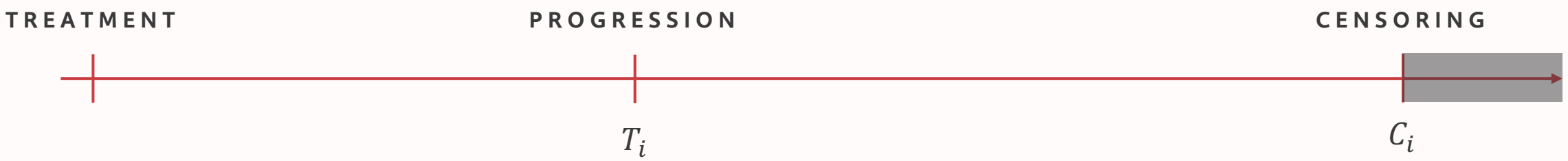
Example 1: uncensored data



Example 2: censored data

Mathematical description

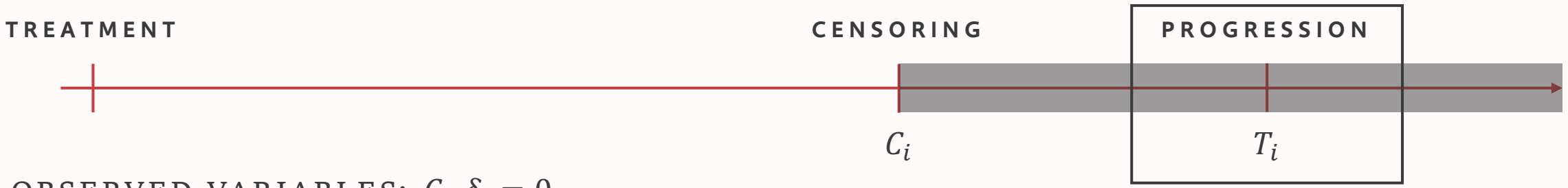
Example 1: uncensored data



OBSERVED VARIABLES: $T_i, \delta_i = 1$

Example 2: censored data

In a real dataset, we wouldn't see this event!



OBSERVED VARIABLES: $C_i, \delta_i = 0$

Mathematical description

- Ideal world:
  - Event time T_i
  - Censoring time C_i
- Observed world:
  - If the event happens first, we observe T_i
  - If censoring happens first, we observe C_i
  - We can summarize this by defining $Y_i = \min(T_i, C_i)$
  - We also observe whether the event occurred first or censoring occurred first, δ_i
  - *Summary*: in survival data, we only observe (Y_i, δ_i)
- Key challenge of survival analysis:
  - *How do we draw conclusions about T_i, when all we observe is (Y_i, δ_i)?*

Maximum Likelihood Learning

The objective function

The likelihood function

- Can view survival analysis through the likelihood function
- Gives *probability of data given model parameters*: $p(\text{data}|\ \text{parameters})$
  - Data is $\{(Y_i, \delta_i)\}_{i=1}^N$
  - *Parameters* depend on what model we choose: denoted by θ
  - We also have *features* x_i for each patient
    - Tabular variables
    - Pathology images
  - Then the likelihood is $p(\text{data}|\ \text{parameters}, \text{features})$
  - In our notation, this is $p\left(\{(Y_i, \delta_i)\}_{i=1}^N \middle|\ \{(x_i)\}_{i=1}^N, \theta\right)$
  - If each patient is *independent*, this becomes $\prod_{i=1}^N p(Y_i, \delta_i|\ x_i, \theta)$

Modeling choices

- Two immediate questions:
  1. How do we choose the model? This will determine the form of the likelihood.
  2. For a given model choice, how do we choose the parameters, θ?
- Our approach:
  1. The model is *a multimodal deep neural network*
  2. We choose θ by *maximum likelihood*

Maximum likelihood learning

- Choose θ that maximizes $\prod_{i=1}^{N} p(Y_i, \delta_i | x_i, \theta)$, written as:

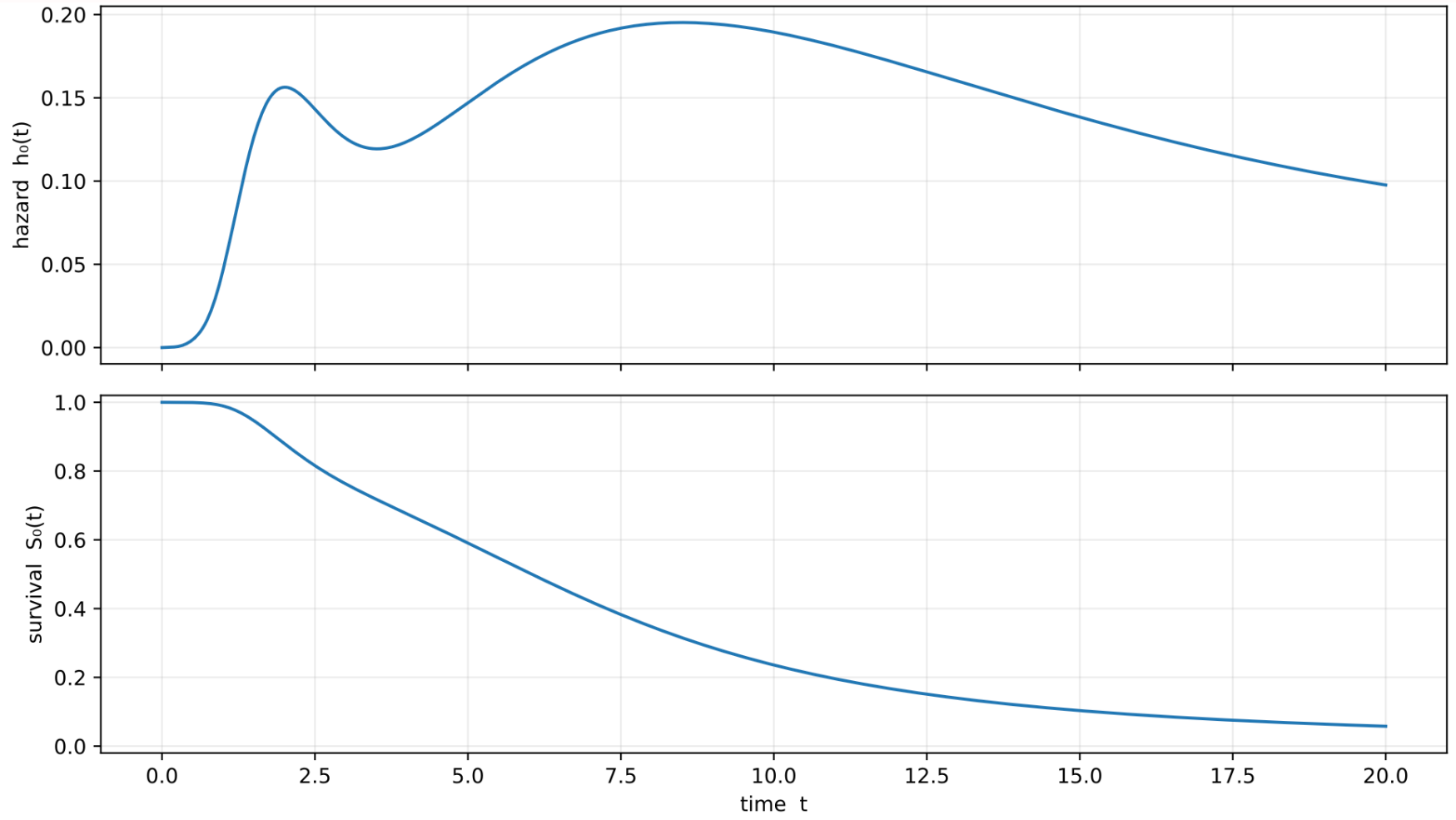
$$\prod_{i=1}^{N} \underbrace{f(y_i | x_i, \theta)^{\delta_i}}_{\text{Probability density for event at time } y_i} \times \underbrace{S(y_i | x_i, \theta)^{1-\delta_i}}_{\text{Probability of not having event before time } y_i}$$

- $S(y_i | x_i, \theta)$ is the *survival function*, defined as $\Pr(Y_i > y_i | x_i, \theta)$
- The δ_i determine which factor is present
  - $\delta_i = 1$; get event probability density
  - $\delta_i = 0$; get "survived until this time *without* an event" probability

Cox models and proportional hazards

- Typically, we now make the *proportional hazards assumption*.
- Roughly speaking:
  - Define the instantaneous probability of having an event as the *hazard*
  - Assume the hazard only depends on the patient features:
    1. Multiplicatively
    2. In a time-independent way
- These assumptions form the *Cox proportional hazards* model
- Extremely popular and convenient because:
  1. Handles continuous time naturally
  2. Likelihood only depends on *relative* hazards between patients
  3. No need to estimate the *shape* of the hazard function

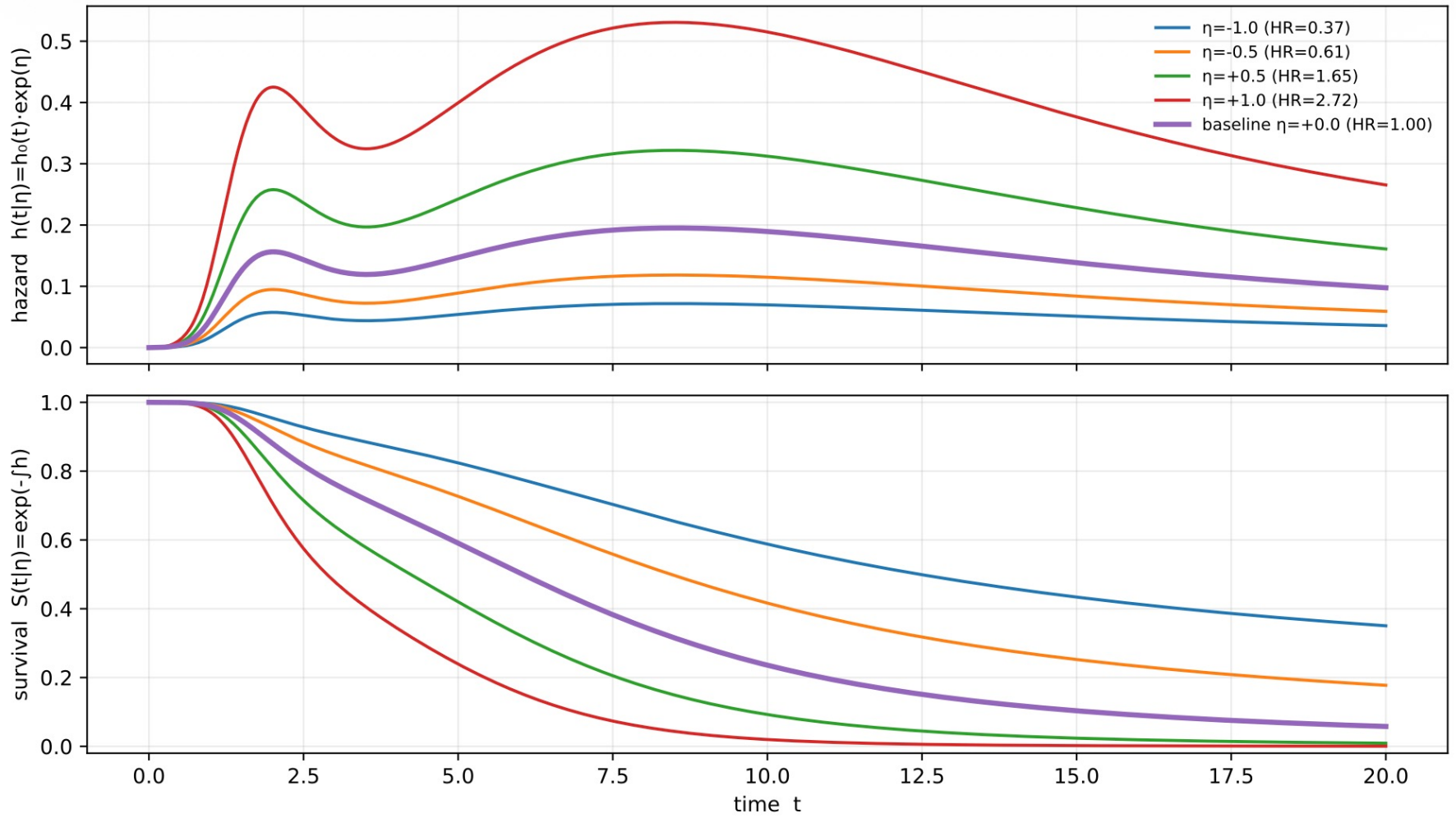
Understanding proportional hazards



Some arbitrary hazard function. “Instantaneous probability of an event”

The corresponding survival function, $\Pr(Y_i > y_i | x_i, \theta)$. Notice it is *decreasing*

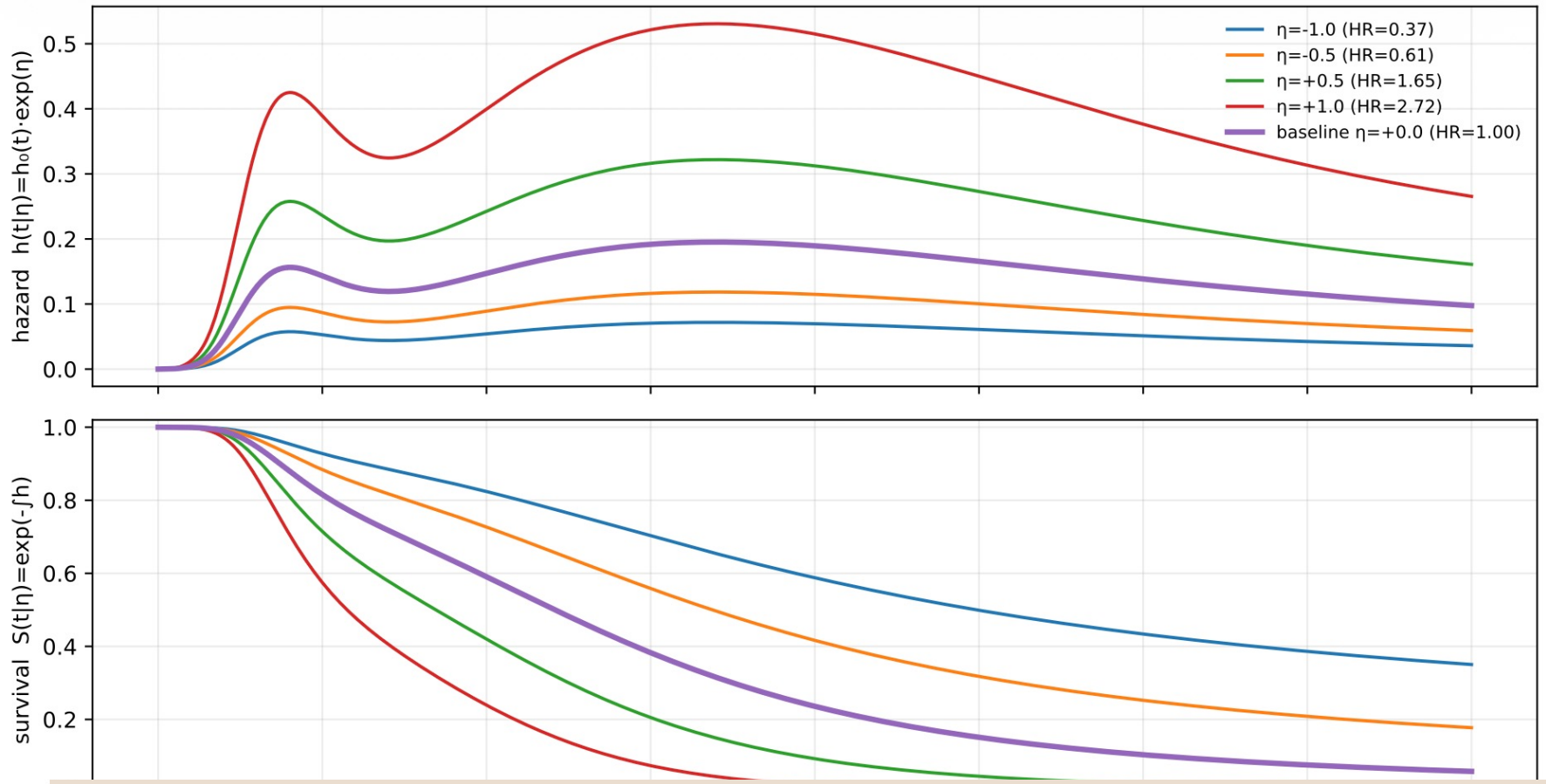
Understanding proportional hazards



Hazard functions obtained by multiplicative, time-independent patient feature variation

The corresponding survival functions. Different patients have different survival probabilities

Understanding proportional hazards



Hazard functions obtained by multiplicative, time-independent patient feature variation

The corresponding survival functions. Different patients have different survival probabilities

Note: no matter how we vary the patient features, it is impossible to make survival curves cross each other under the proportional hazards assumption.

Discrete-time survival models

Beyond proportional hazards

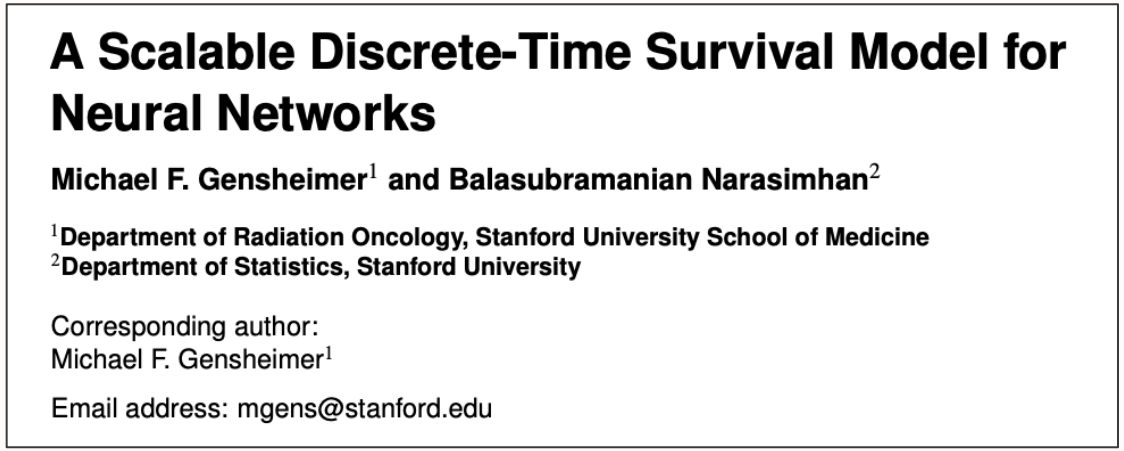
Discrete-time survival models

- We will *not* make the proportional hazards assumption
- *Deep learning isn't constrained by mathematical convenience!*

- Define the *discrete-time hazard function* as:
  - $h_{i,k} = \Pr(T \text{ in bin } k \mid T \text{ not in bins 1 to } k-1, x_i)$
- The likelihood can be written entirely in terms of h_{ik}:
  - Let t_i be the bin index of the event/censoring time Y_i
  - $p(Y_i, \delta_i | \, x_i, \theta) = \left[\prod_{k=1}^{t_i - 1} (1 - h_{i,k}) \right] (1 - h_{i,t_i})^{1-\delta_i} (h_{i,t_i})^{\delta_i}$

Discrete-time survival models

- We can easily compute the likelihood *provided* we specify $h_{i,k}$.
- Idea: parameterize $h_{i,k}$ with a deep neural network f_θ:
  - $h_{i,k} = f_\theta(x_i, k)$
  - Learn θ by maximum likelihood
- This is the idea behind *Nnet-survival* (2018)



A Scalable Discrete-Time Survival Model for Neural Networks

Michael F. Gensheimer[1] and Balasubramanian Narasimhan[2]

[1]**Department of Radiation Oncology, Stanford University School of Medicine**
[2]**Department of Statistics, Stanford University**

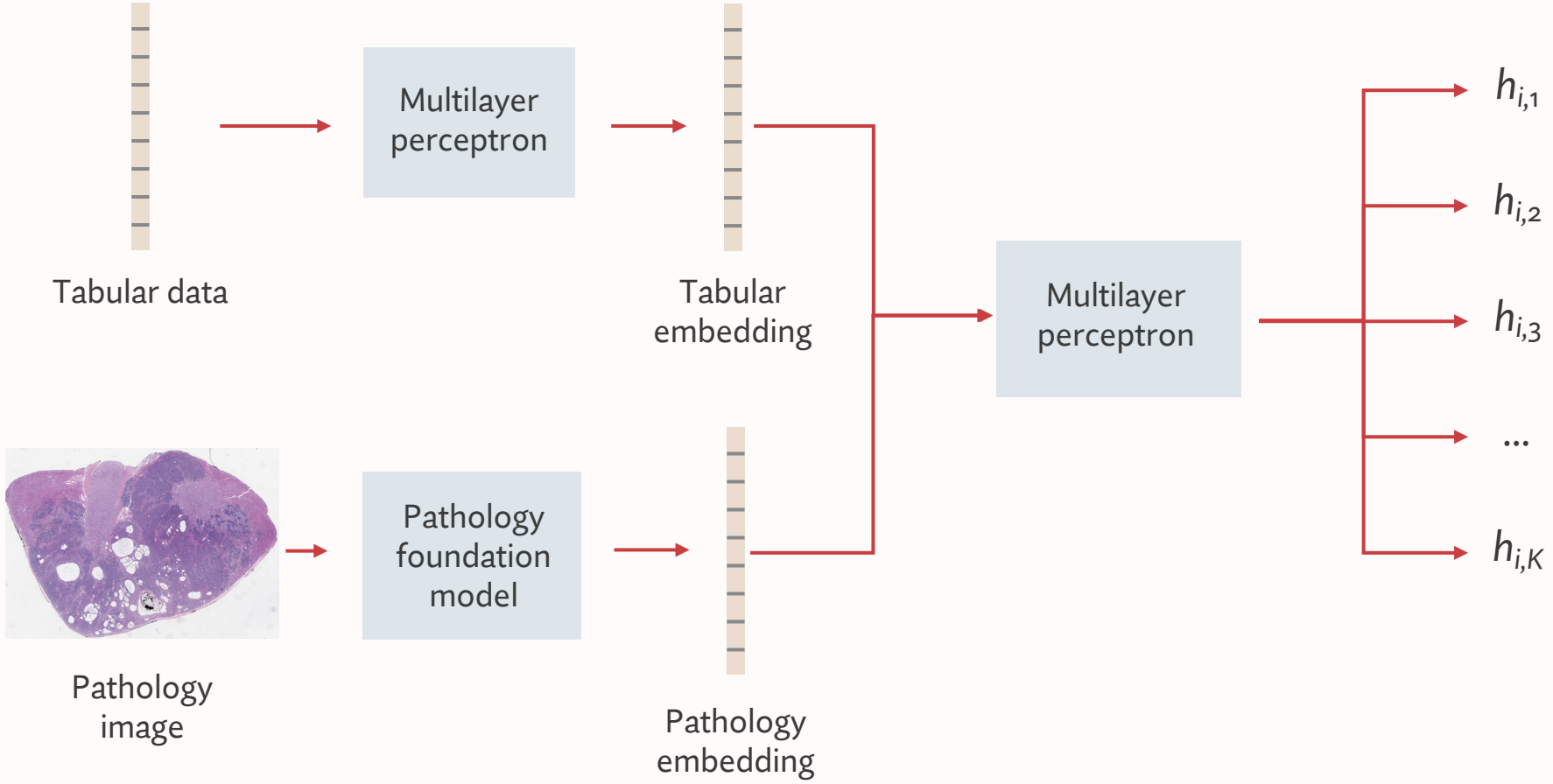
Corresponding author:
Michael F. Gensheimer[1]

Email address: mgens@stanford.edu

Gensheimer, Michael F., and Balasubramanian Narasimhan. “A scalable discrete-time survival model for neural networks.” PeerJ 7 (2019): e6257.

Discrete-time survival models

We can apply this directly to our project:



Discrete-time survival models

- At inference time, using $h_{i,k}$, we can predict the answer to *any* question related to cancer progression/recurrence
- Example:
  - *What is the probability of a recurrence between years two and four?*
    - Let t_2 be the bin index corresponding to two years
    - Let t_4 be the bin index corresponding to four years
  - $\Pr(2 \text{ years} < T_i \leq 4 \text{ years} | \, x_i, \theta) = \underbrace{\left[\prod_{k=1}^{t_2} (1 - h_{i,k}) \right]}_{\text{No event in first two years}} \underbrace{\left[1 - \prod_{k=t_3}^{t_4} (1 - h_{i,k}) \right]}_{\text{Event in year 3 or 4}}$

Integrating pathology

From gigapixel images to risk scores

Why Pathology?

- Tabular features (T-stage, N-stage, Age, Smoking, etc.) capture stage and clinical risk broadly.

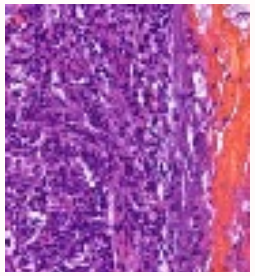
- Pathology Images capture tumor morphology
  - How strongly the immune system has responded?
  - How much tumor invades the surrounding tissue?

- Goal: extract features from pathology images and fuse it with tabular.

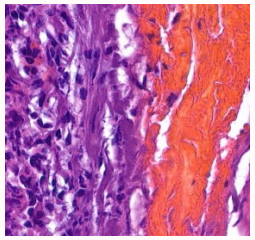
Whole-slide images (WSIs) are gigapixel

- A typical WSI is scanned at 20×/40× magnification
  - ~100k × 100k pixels | 1–4 GB per slide
- Information-dense at each magnification
- Cannot input directly to CNNs / Transformers
  - Tumor occupies a small fraction
  - GPU memory constraints
- Must process in tiles (small patches)

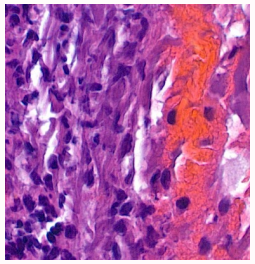
5×



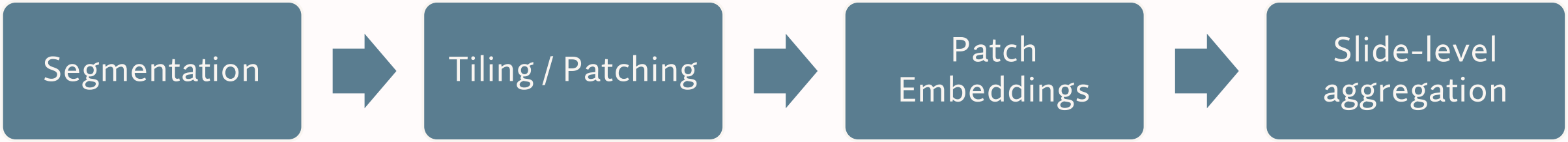
20×



40×



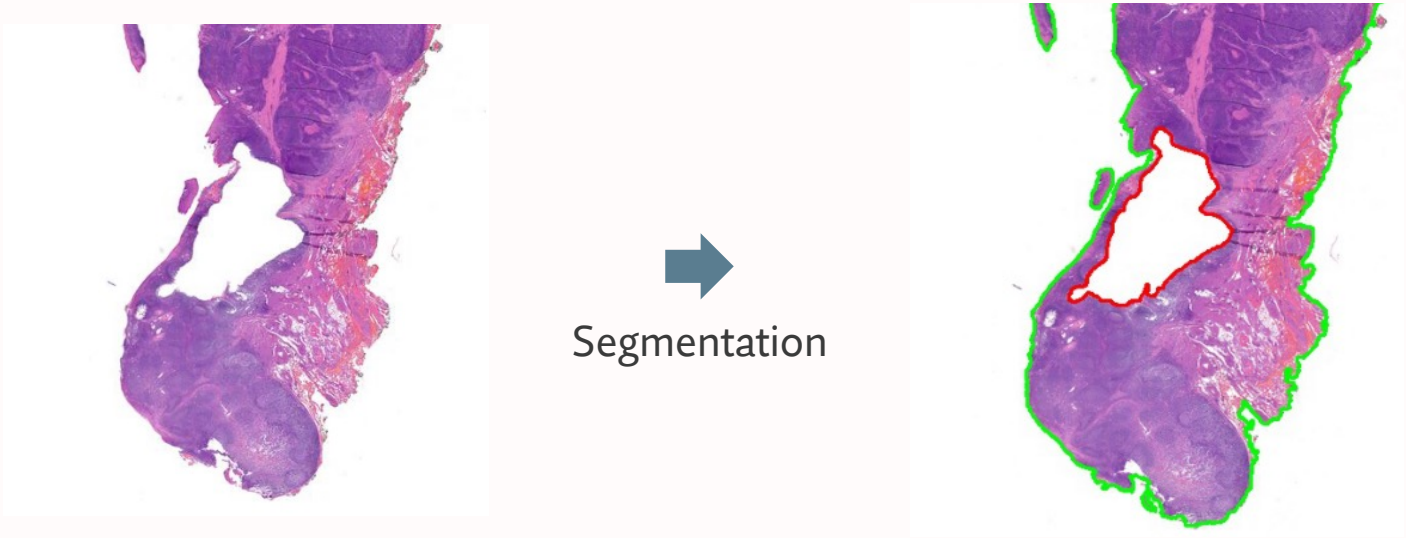
Standard WSI processing pipeline



Output: a fixed-sized embedding per patient

Tissue Segmentation

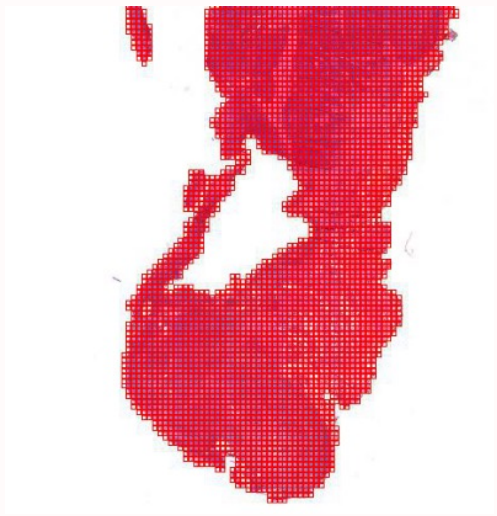
- WSIs have lots of non-tissue space
- Goal: keep relevant tissue



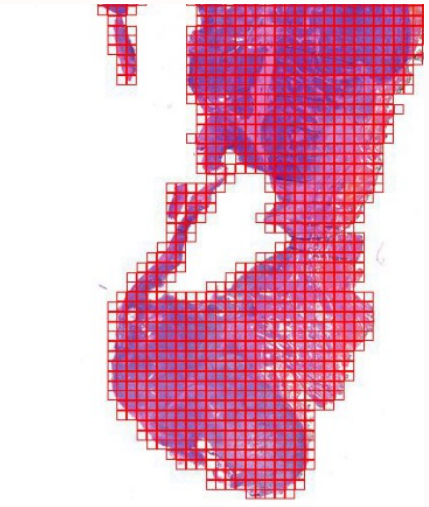
- Drops a significant slide area → major compute save.

Tiling / Patching strategy

- Crops segmented tissue into small patches
- Design choices
  - Patch Size: 256×256 or 512×512
  - Magnification: 20× (≈0.5 µm/pixel)
  - Stride: non-overlapping
- Output: 5,000–20,000 patches per WSI
- Each slide = a bag of patches



Patch size: 256×256



Patch size: 512×512

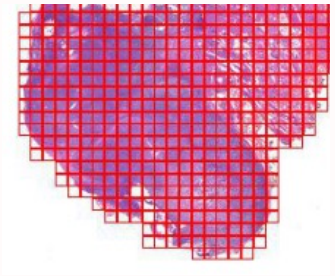
Pathology foundation models

- Problem: Training a CNN from scratch on small cohorts overfits.

- Pathology foundation models are pre-trained on millions of WSIs with self supervision
  - UNI, CONCH, TITAN (Harvard)
  - GigaPath (Microsoft)
  - ATLAS (Mayo Clinic)

- Goal: generate patch embeddings without task supervision

Patch Embeddings



N patches

Foundation model

Each patch is encoded into a feature vector: $\boldsymbol{h}_n \in \mathbb{R}^D$

A bag for one patient = $\mathbf{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\}$

Multiple Instance Learning (MIL)

- Setup: Each WSI is a bag of N patch embeddings.
- Goal: Learn slide-level labels from patches.
- Requires a patch aggregator that:
  - Handles variable size bags
  - Identifies the most informative patches
  - Outputs a fixed-size slide-level embedding
  - Differentiable end-to-end

Attention-based MIL (ABMIL)

Setup: One bag of embeddings per patient = $\mathbf{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\}$, $\boldsymbol{h}_n \in \mathbb{R}^{768}$ for CONCH

Core Idea: Let the model decide which patches matter.

Compute an attention score α_n for every patch:

$$\alpha_n = \frac{\exp(w^\top \tanh(V h_n))}{\sum_{j=1}^{N} \exp\left(w^\top \tanh\left(V h_j\right)\right)}$$

$V \in \mathbb{R}^{128\times768}$ *and* $w \in \mathbb{R}^{128}$ are learned weights shared across all patients and

$$\alpha_n \in [0,1] \text{ and } \sum_n \alpha_n = 1$$

Ilse, Maximilian, Jakub Tomczak, and Max Welling. "Attention-based deep multiple instance learning." *International conference on machine learning*. PMLR, 2018.

Attention-based MIL (ABMIL)

Aggregation: collapse the whole bag into one vector

$$z = \sum_{n=1}^{N} \alpha_n \cdot h_n \in \mathbb{R}^{768}$$

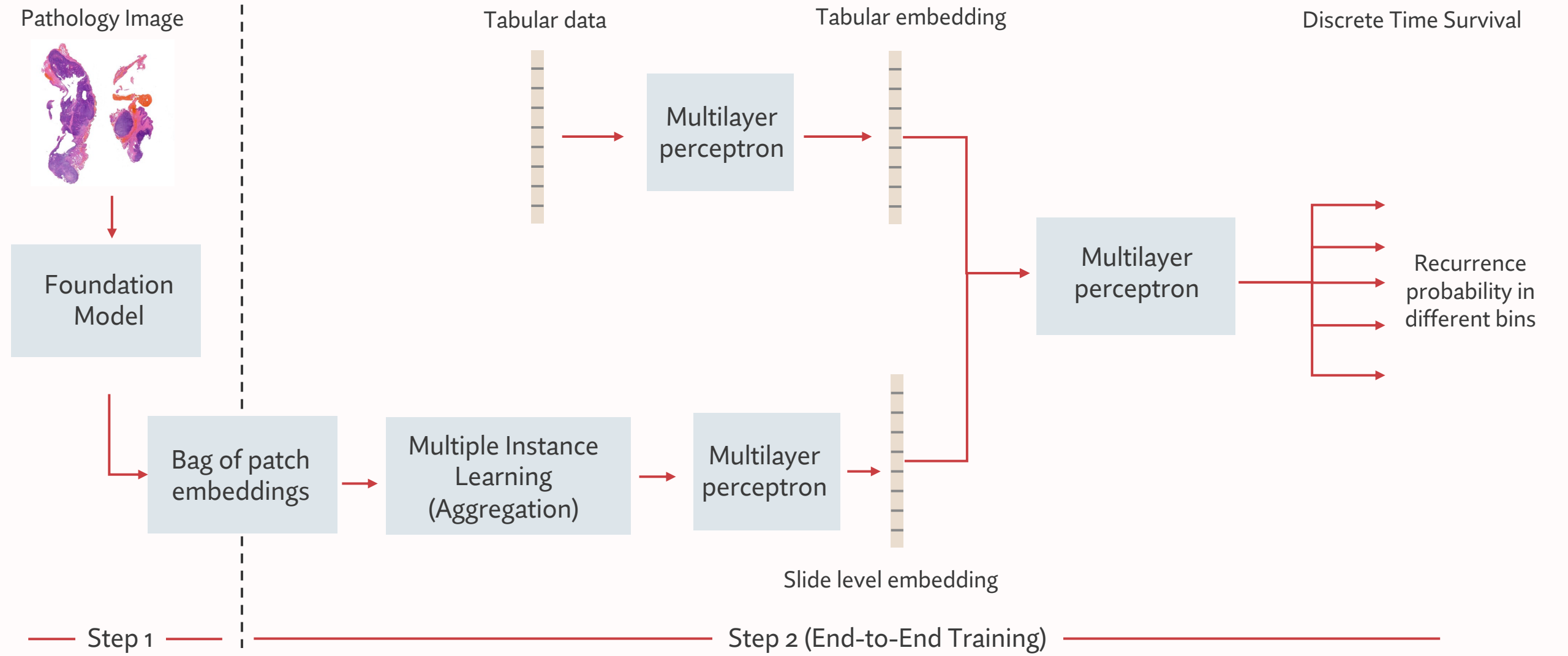
Patches with high α_n dominate z

z is concatenated with tabular MLP branch → slide-level prediction

Several variants:

- CLAM: adds clustering loss for patch-level supervision
- TransMIL: patch-to-patch attention.

Multimodal fusion



Evaluating model performance

- Harell's Concordance Index:
  - Also known as C-Index, C-statistic
  - Standard metric of survival model evaluation.

- Generalization of AUROC to censored time-to-event data

- 0.5 = random; 1.0 = perfect; ≥0.75 is generally considered good

Evaluating model performance

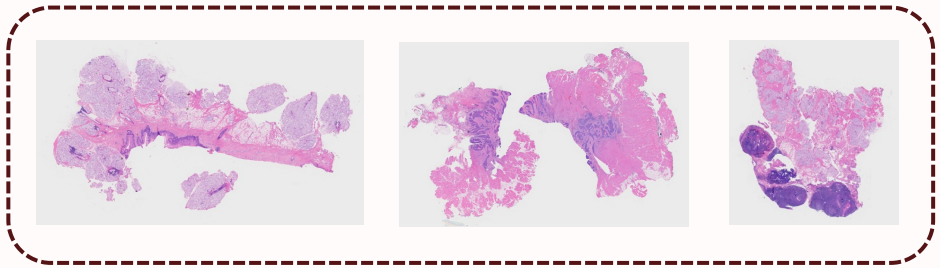
- Task: Predict progression free survival (PFS)
  - Foundational Model: CONCH
  - Aggregator: ABMIL
  - 5-fold cross validated C-index
- Multimodal > pathology image only > tabular only
- Good model performance is necessary, not sufficient.

Open Questions

One patient ≠ One Image

- Current Setup: one primary tumor WSI per patient
  - Assumption: this slide represents the disease.

- However, a single resection yields 5–10 WSIs for primary tumor.



- Involvement of Lymph Nodes is also a prognostic factor in HPV+ head-neck cancer → 5–10 WSIs for lymph nodes.

- What we need: Patient-Level Multiple Instance Learning.

Select first, embed later

- Current Setup:
  - Embed first, select later
  - Extract embeddings for every patch
  - ABMIL assigns low attention scores to most patches.
- Most compute is spent on patches which the model later ignores.
- Better: Select first, embed later

Key Takeaways

- Pathology foundation models helps generalize over small cohorts.
- Multimodal fusion improves time-to-event prediction compared to single modality.
- Open questions:
  - How to use all the available WSIs?
  - Compute efficient patch selection strategy.

Question & Answer