A Machine-Learned Comorbidity Index

Suleman Baloch[1], Kishlay Jha[2], Alberto M. Segre[1], Philip M. Polgreen[3], Bijaya Adhikari[1]

University of Iowa

[1]Department of Computer Science, University of Iowa
[2]Department of Electrical and Computer Engineering, University of Iowa
[3]Department of Internal Medicine, University of Iowa

Connect with me!

Presenter: Suleman Baloch

Background and Motivation

A **comorbidity index** summarizes the diagnosis burden recorded for a hospital admission into a single scalar score.

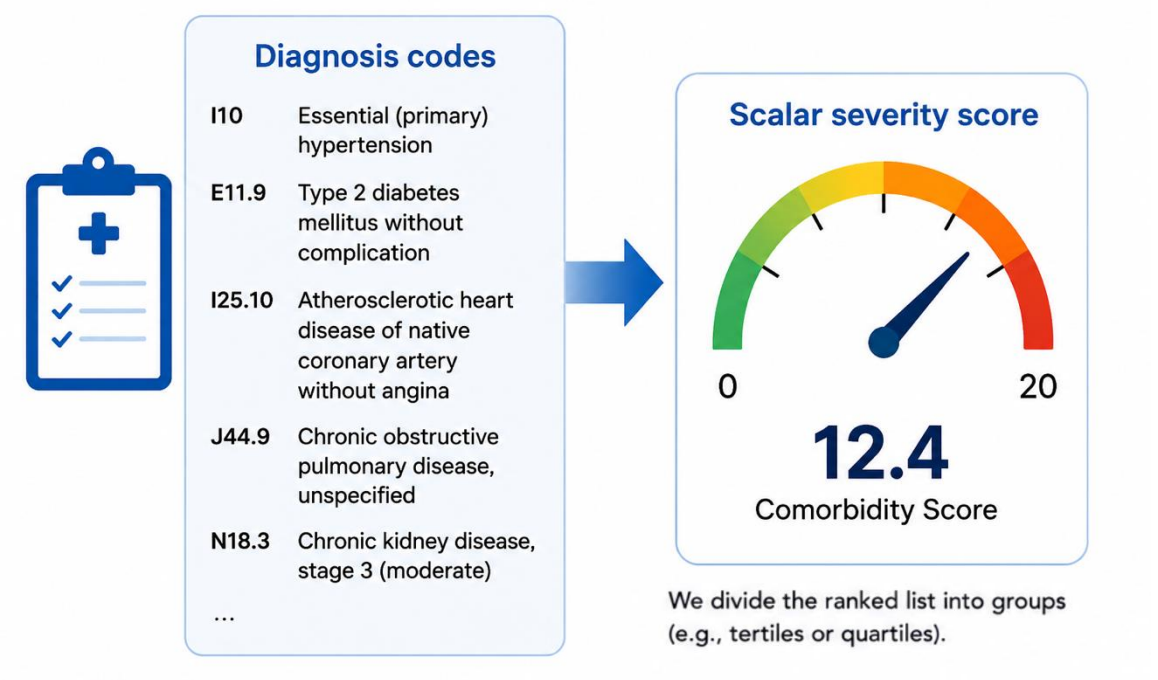
For a hospital admission:

Diagnosis codes → scalar severity score

Higher score usually means the admission has more severe or complex comorbid illness.

Comorbidity scores are widely used in healthcare:

1. Patient stratification
2. Risk adjustment
3. Comparing outcomes across patient groups
4. Identifying high-risk admissions



Traditional Comorbidity Scores

Widely used traditional comorbidity indices include:

1. Charlson Comorbidity Index (CCI[1])
2. van Walraven Elixhauser Index (ECI[2,3])

How they work:

1. Identify predefined comorbidity conditions from ICD codes
2. Assign each condition a fixed weight
3. Sum the weights to get one scalar score

	Comorbid Condition	Weight
1	Myocardial infarction	1
2	Congestive heart failure	1
3	Peripheral vascular disease	1
4	Cerebrovascular disease	1
5	Dementia	1
6	Chronic pulmonary disease	1
7	Connective tissue disease / rheumatologic disease	1
8	Peptic ulcer disease	1
9	Mild liver disease	1
10	Diabetes without end-organ damage	1
11	Diabetes with end-organ damage	2
12	Hemiplegia or paraplegia	2
13	Moderate or severe renal disease	2
14	Any tumor / malignancy without metastasis	2
15	Leukemia	2
16	Lymphoma	2
17	Moderate or severe liver disease	3
18	Metastatic solid tumor	6
19	AIDS / HIV	6

Charlson Comorbidity Index (CCI) Formula

$$\text{CCI} = \sum_k w_k \times I\{\textit{condition } k \textit{ present}\}$$

Where:

- w_k = weight for comorbid condition k (from table above)
- $I\{\textit{condition } k \textit{ present}\} = 1$ if condition k is present, 0 otherwise

1. Charlson ME, Pompei P, Ales KL, MacKenzie CR. A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. J Chronic Dis. 1987;40(5):373-83. doi: 10.1016/0021-9681(87)90171-8. PMID: 3558716.
2. Elixhauser A, Steiner C, Harris DR, Coffey RM. Comorbidity measures for use with administrative data. Med Care. 1998 Jan;36(1):8-27. doi: 10.1097/00005650-199801000-00004. PMID: 9431328.
3. van Walraven C, Austin PC, Jennings A, Quan H, Forster AJ. A modification of the Elixhauser comorbidity measures into a point system for hospital death using administrative data. Med Care. 2009 Jun;47(6):626-33. doi: 10.1097/MLR.0b013e31819432e5. PMID: 19433995.

Limitations of Comorbidity Indices

Limitation 1: Mortality-centric fixed weights

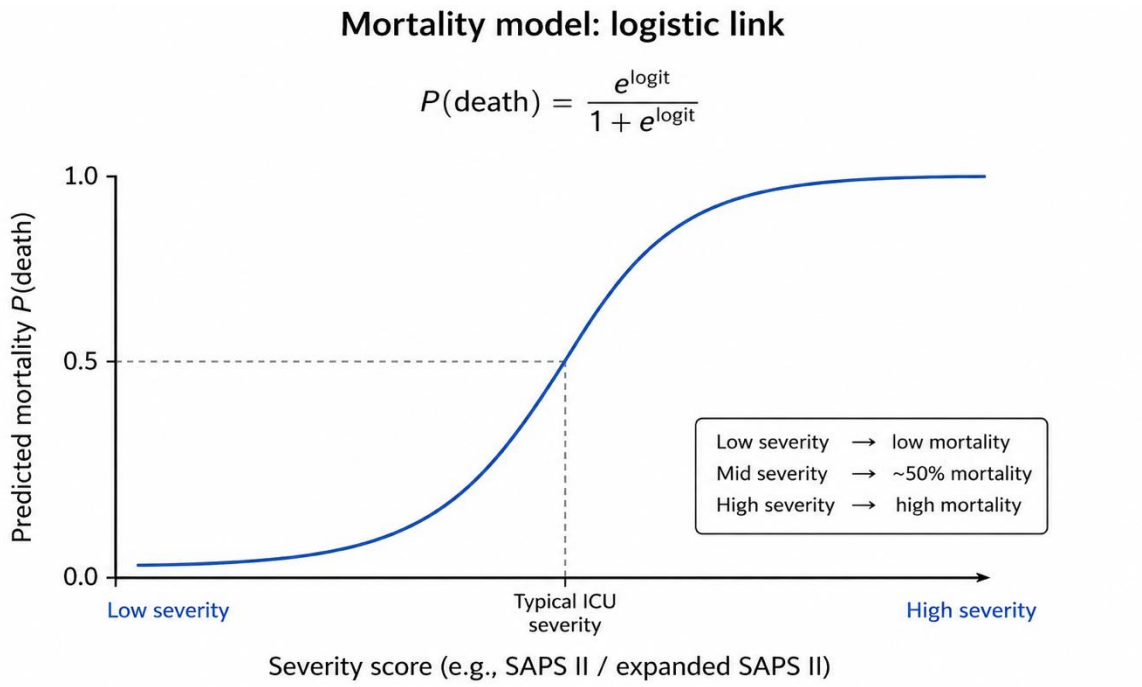
Traditional indices were designed mainly for mortality but are often used for other outcomes.

Comorbidity	Score
Prior myocardial infarction	1
Congestive heart failure	1
Peripheral vascular disease	1
Cerebrovascular disease	1
Dementia	1
Chronic pulmonary disease	1
Rheumatologic disease	1
Peptic ulcer disease	1
Mild liver disease	1
Diabetes	1
Cerebrovascular (hemiplegia) event	2
Moderate-to-severe renal disease	2
Diabetes with chronic complications	2
Cancer without metastases	2
Leukemia	2
Lymphoma	2
Moderate or severe liver disease	3
Metastatic solid tumor	6
Acquired immuno-deficiency syndrome (AIDS)	6

Fixed weights calibrated to predict mortality

Limitation 2: Linear, rule-based structure

They are linear and rule-based, so they may miss nonlinear diagnosis–outcome relationships.



Framing Comorbidity Scoring as a Machine Learning Problem

These two limitations suggest a **learning problem:**

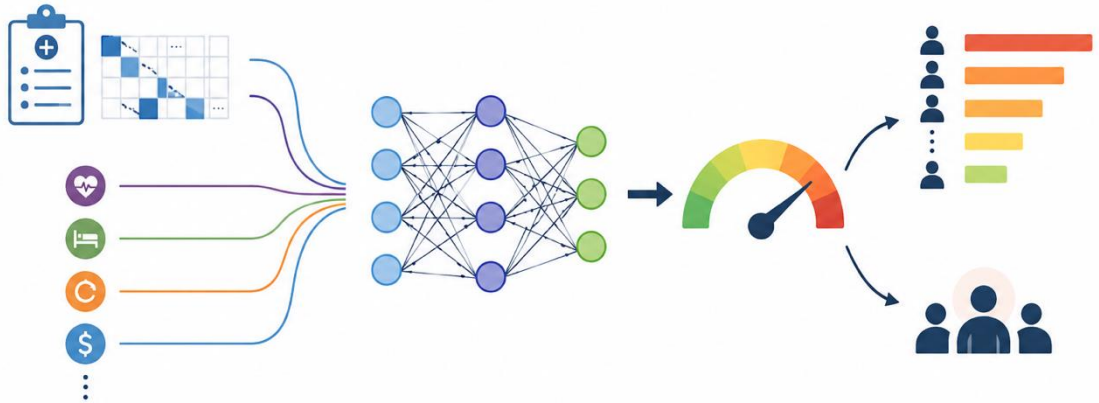
"Can we learn a scalar comorbidity score from data that
aligns with multiple outcomes and captures nonlinear risk–outcome interactions?"

We therefore formulate comorbidity scoring as a **machine learning problem**: learn a score that is

1. **Scalar**: preserves the practical simplicity of traditional indices
2. **Multi-outcome aligned**: captures shared comorbidity burden across outcomes
3. **Nonlinear**: captures complex diagnosis–outcome relationships
4. **Stratifiable**: ranks admissions and identifies a high-severity group
5. **Data-driven:** learned from diagnosis codes and outcomes

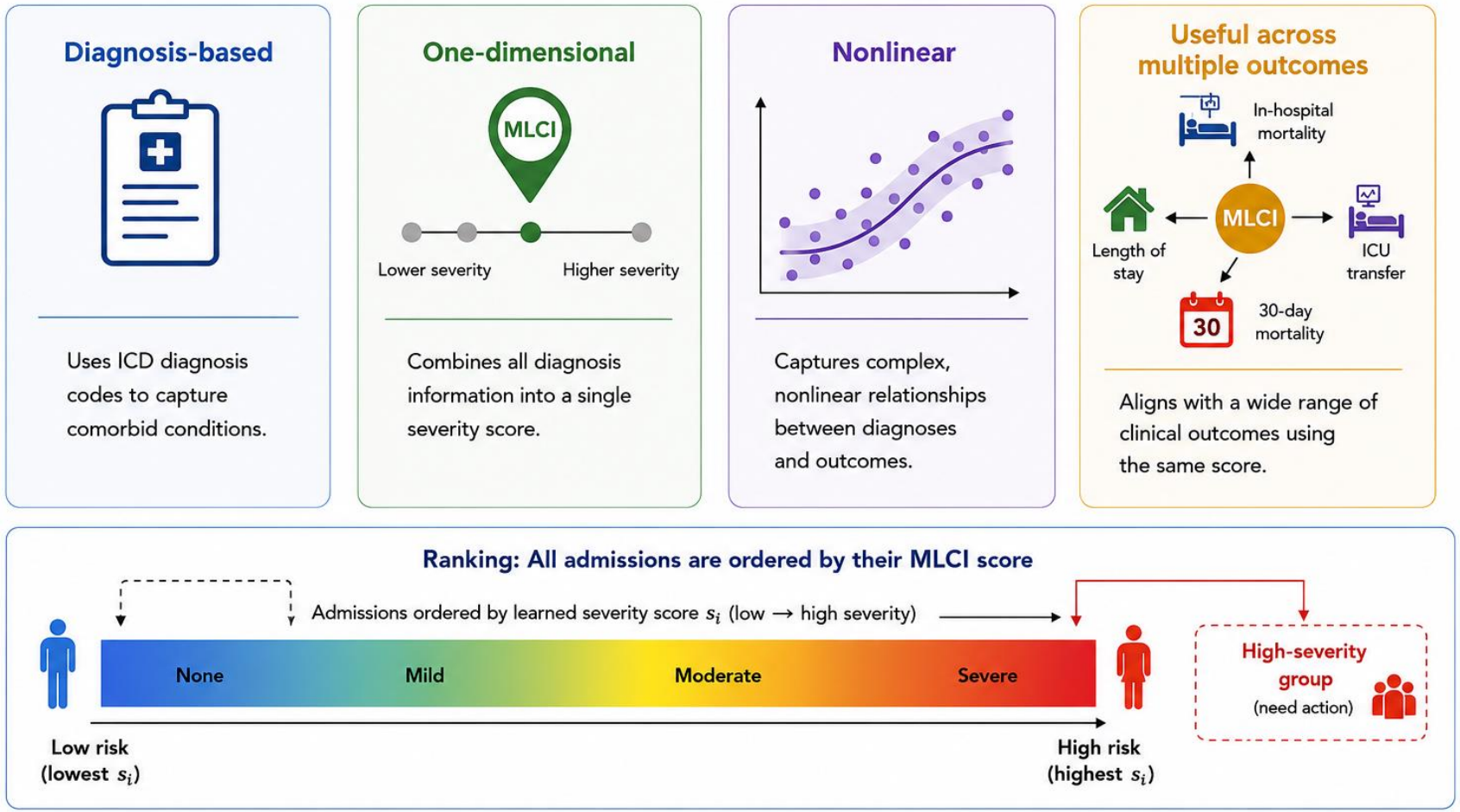
To address this, we introduce MLCI:

A Machine-Learned Comorbidity Index



MLCI Overview

MLCI learns one data-driven scalar admission-level score by maximizing normalized dependence with multiple clinical outcomes.



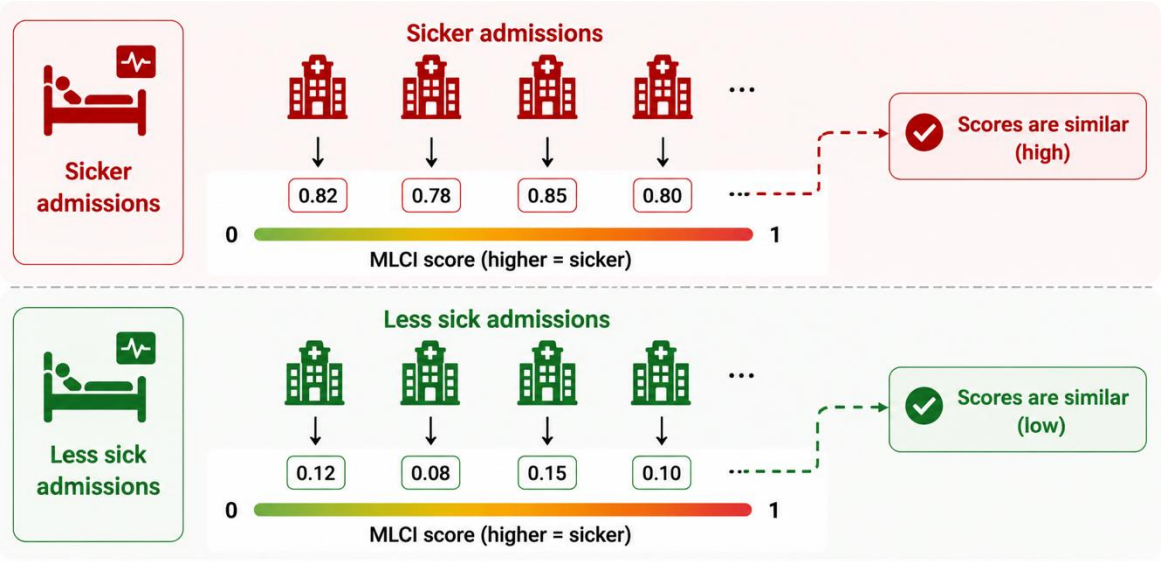
MLCI: Intuition

Severity-ordering intuition: Admissions with greater latent comorbidity burden should receive higher learned scores after orientation. Therefore, after orientation, higher MLCI score = higher learned severity.

Intuition: For every pair of admissions, we pose two questions

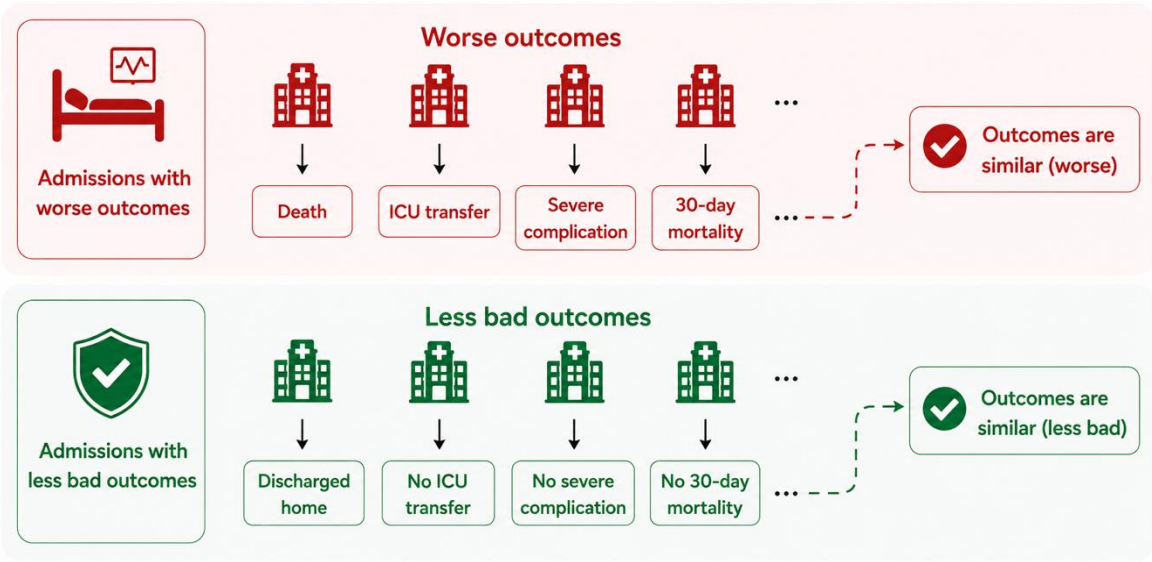
Are their scores similar?

→ Sicker admissions should have similar high scores; less sick admissions should have similar low scores.



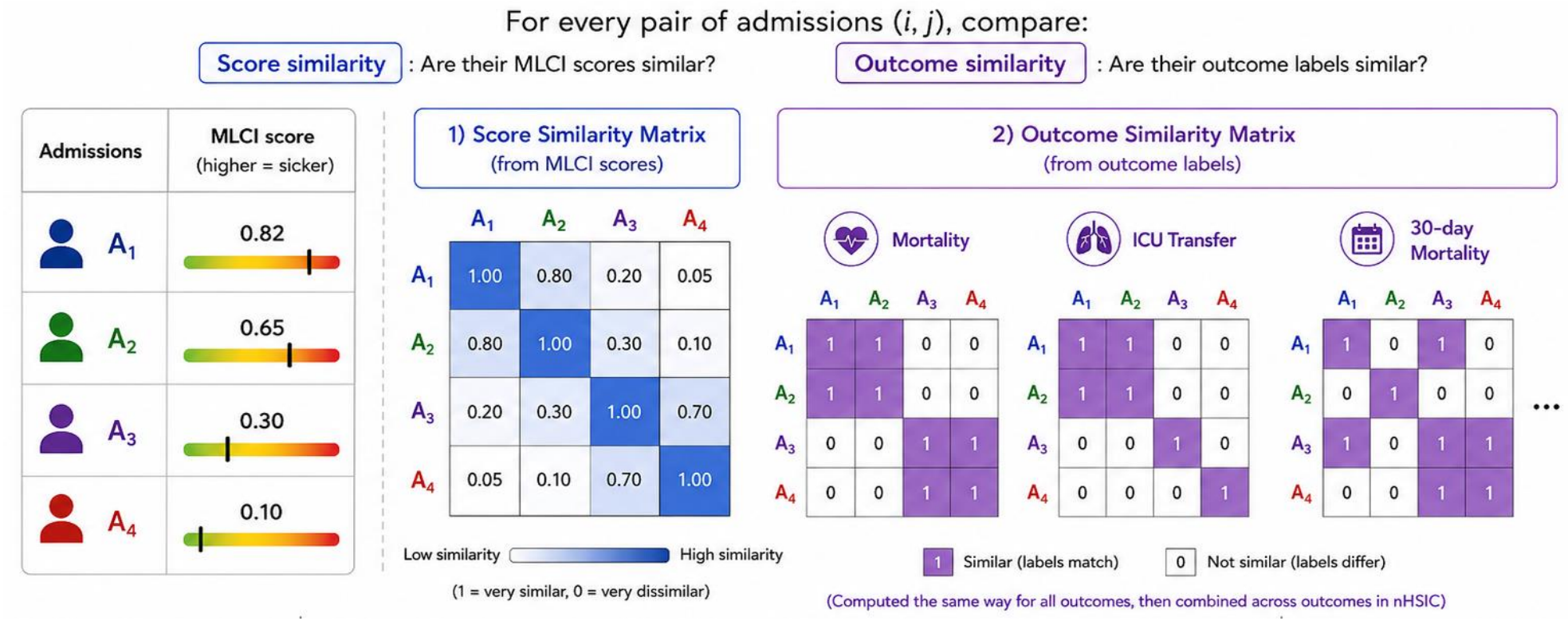
Are their outcomes similar?

→ Admissions with matching outcome labels should be similar.



MLCI: Intuition

Pairwise view: For every pair of admissions, MLCI compares two similarities:
1. Score similarity: Are their MLCI scores close?
2. Outcome similarity: Are their clinical outcome labels similar?
MLCI then maximizes the alignment between these two similarity matrices using **nHSIC**.
Goal: admissions that are close in MLCI score should also have similar outcome labels.



MLCI Architecture

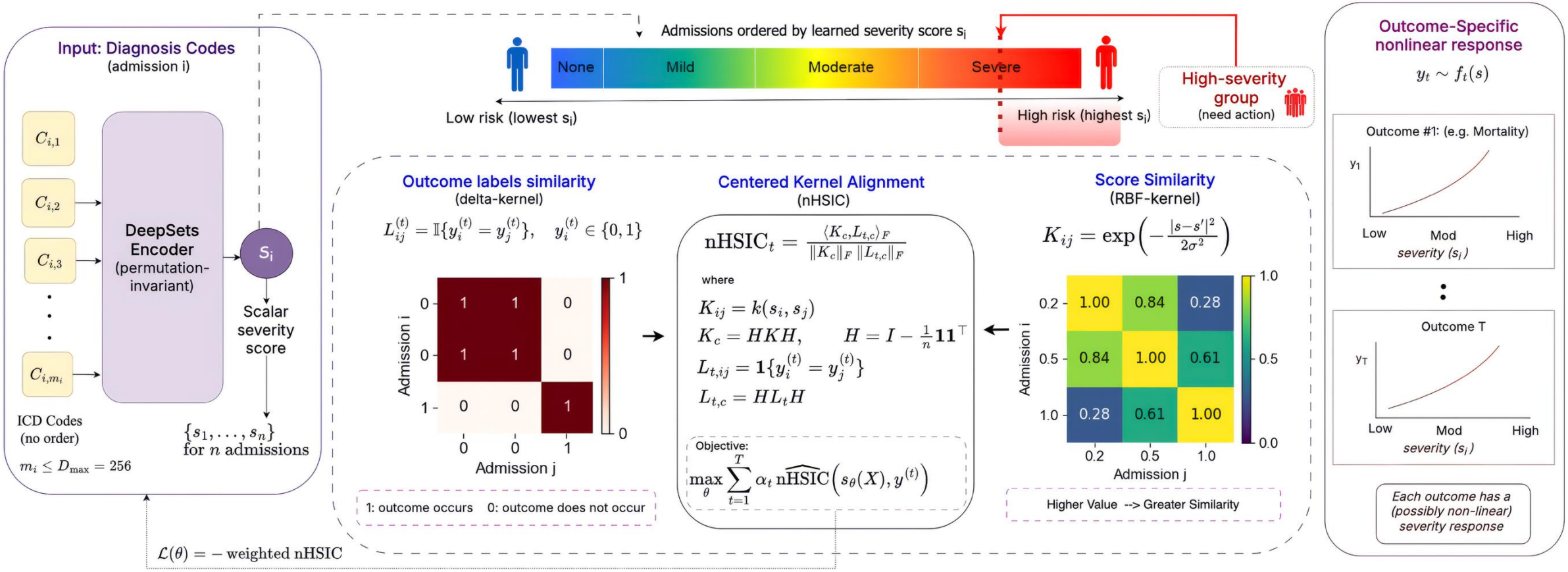


Figure guide: A DeepSets encoder maps diagnosis codes to a scalar MLCI score. During training, MLCI learns this score by aligning score similarity with outcome-label similarity using nHSIC. At inference, only diagnosis codes are needed to compute the MLCI score and rank admissions. Post hoc outcome-specific curves map the learned MLCI score to outcome-specific risk patterns.

Theoretical Analysis: Oracle Setup

Theoretical goal: Our analysis characterizes when one oracle comorbidity score can support an approximate shared admission-level ordering across multiple clinical outcomes.

Setup: We analyze this in a finite-sample oracle setting.

1. Assume each admission has an unobserved latent severity

$$z_i = \text{latent disease burden for admission } i$$

2. Each outcome depends on severity through its own function

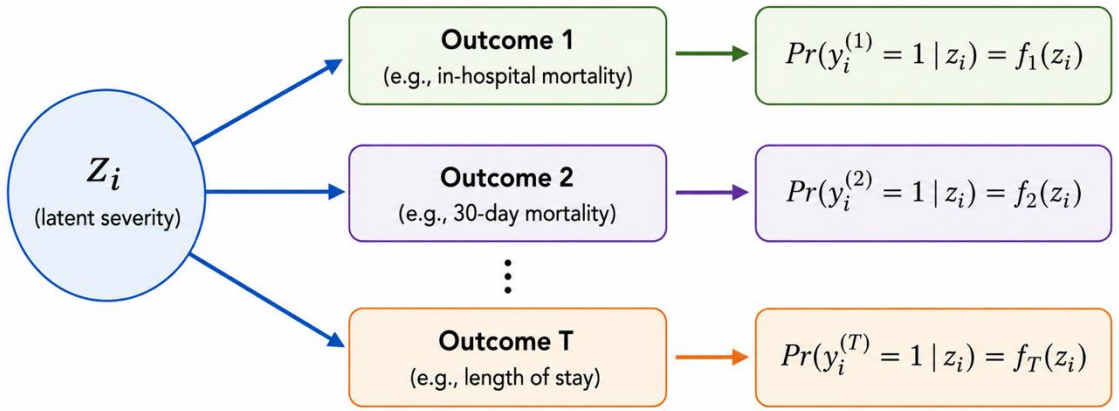
$$\Pr\left(y_i^{(t)} = 1 \mid z_i \right) = f_t(z_i)$$

Thus, different outcomes may respond differently to severity.

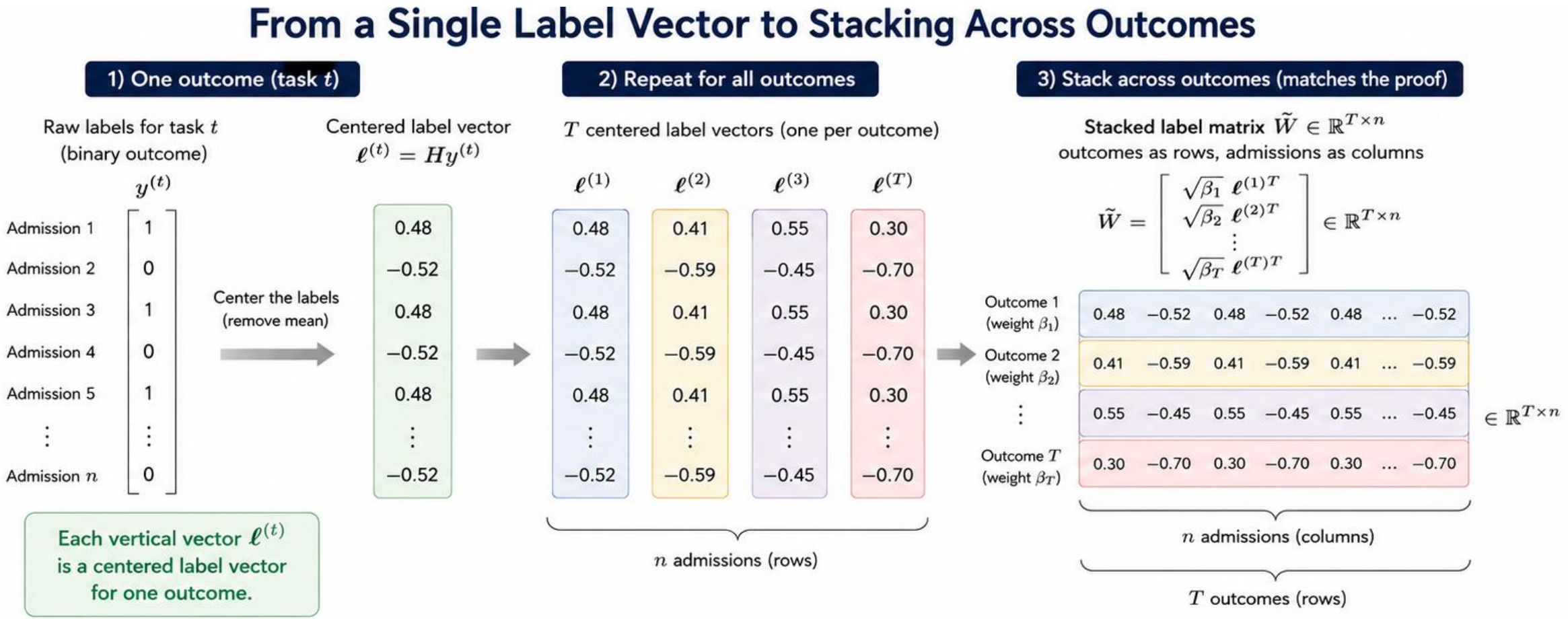
Goal: recover the admission ordering

$$z_i < z_j \Rightarrow r_i \leq r_j$$

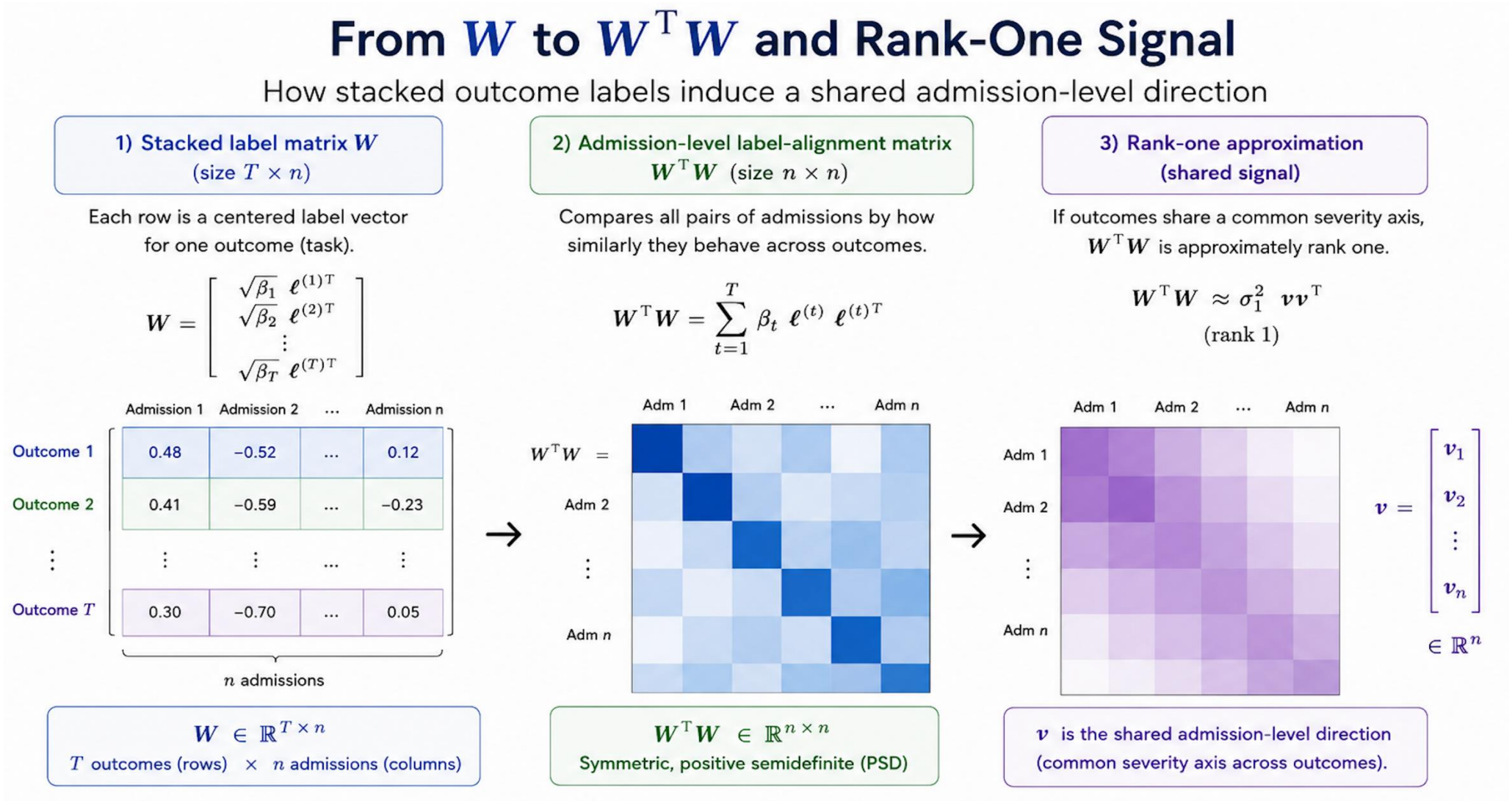
where r_i is the **oracle severity score** used in the theory.



Theoretical Analysis: Cross-Task Label Matrix



Theoretical Analysis: Rank-One Projection



Linking Theory to MLCI

In the theory, we define an **oracle severity score:**

$$r_i = \text{oracle severity score for admission } i$$

It preserves the **latent severity ordering:**

$$z_i < z_j \Rightarrow r_i \leq r_j$$

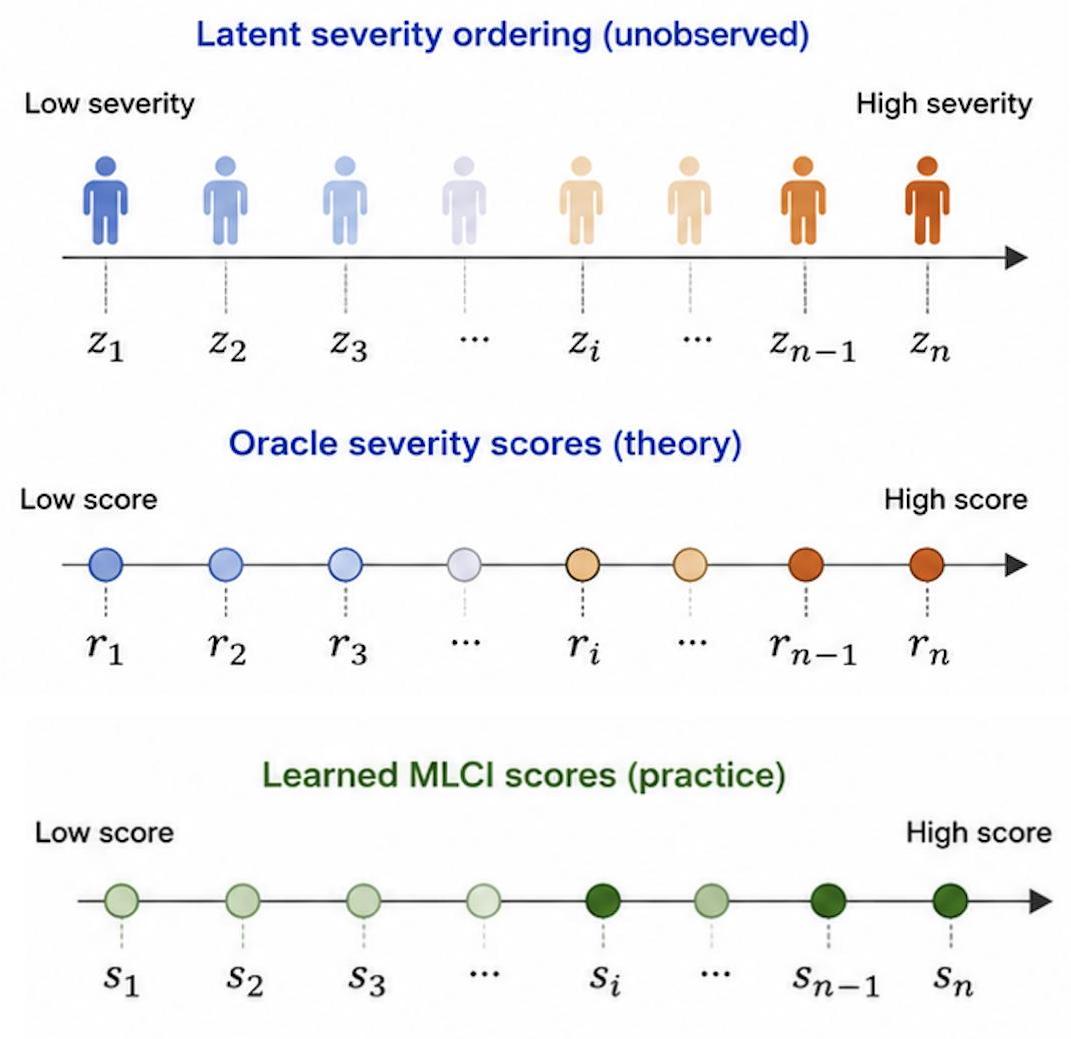
In practice, z_i and r_i are **unobserved**.

MLCI learns

$$s_i = s_\theta(X_i)$$

from diagnosis codes as a **data-driven approximation to** r_i:

$$s_i \approx r_i$$



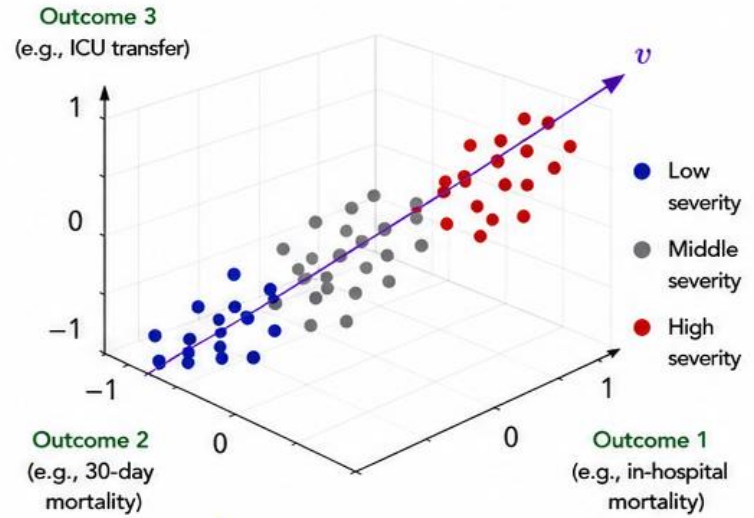
Theoretical Analysis: High Severity Threshold

ICML International Conference On Machine Learning

The same theory also gives a high-severity cutoff.

1 Order admissions by the learned MLCI score s_i

Compute each admission's MLCI score $s_i = s_\theta(X_i)$ from diagnosis codes and sort from low to high.



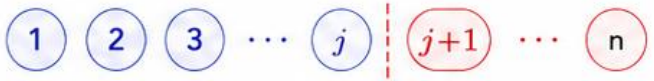
After computing s_i, sort admissions from lowest to highest.

Result: a ranking of all admissions from lower to higher severity.

2 Scan possible split points

A. Setup (after sorting)

We have n admissions sorted by MLCI score:



n = number of admissions

B. What does a split point j mean?

A split point j (where $1 \le j \le n-1$) divides the sorted list into two groups:

Left group (not selected): $\{1, 2, \ldots, j\}$ (lower or equal severity)

Right group (candidate high-severity tail): $R_j = \{j+1, \ldots, n\}$

C. Scan all possible split points

We try every possible split point:

$$j = 1, 2, \ldots, n-1$$

For each j, form threshold vector g_j (1 for indices in R_j, 0 otherwise) and compute its alignment with v.

D. Choose the best split

$$j^* \in \arg\max_j \text{alignment}(v, g_j)$$

3 Choose the best split (cutoff)

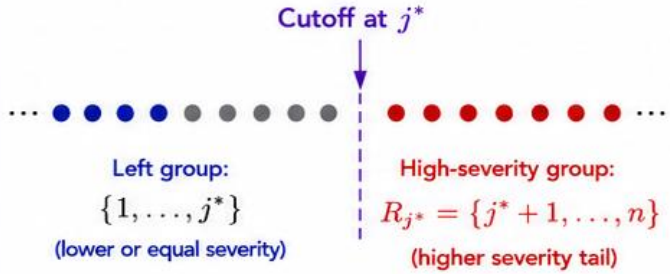
The best split point is j^*.

The selected high-severity group is:

$$R_{j^*} = \{j^*+1, \ldots, n\}$$

Interpretation:

Admissions in R_{j^*} are consistently high severity across all outcomes — they align most with the shared direction v.



Threshold vector at j^* (used to evaluate alignment):

$$g_{j^*} = (\underbrace{0, 0, \ldots, 0}_{j^* \text{ zeros}}, \underbrace{1, 1, \ldots, 1}_{n - j^* \text{ ones}})$$

Experimental Results: Distance Correlation (DCorr)

ICML
International Conference
On Machine Learning

Table 1. **Distance Correlation between each model's unified risk score and clinical outcomes.** Higher is better. Bolded values highlight the best performer per outcome. *For readability, each column is scaled by the power of 10 shown in the header; divide by that factor to recover the original dCorr.*

Model	MIMIC-IV				MIMIC-III			
	MORT ($\times 10^2$)	**30M** ($\times 10^2$)	**LOS** ($\times 10^2$)	**ICU** ($\times 10^2$)	**MORT** ($\times 10^2$)	**30M** ($\times 10^2$)	**LOS** ($\times 10^2$)	**ICU** ($\times 10^2$)
Traditional Clinical Indices								
Charlson (CCI)	12.59	18.44	24.55	16.09	15.54	20.26	15.28	13.07
Elixhauser (ECI)	19.98	23.87	33.15	25.98	21.38	24.70	23.10	13.38
CCI+ECI (scalar)	17.46	22.67	30.89	22.44	19.79	24.06	20.82	14.15
Classical Machine Learning Baselines								
kNN	22.40	23.98	30.60	34.79	22.56	23.22	23.55	11.15
Multinomial Naive Bayes	32.09	32.02	35.78	45.69	24.89	25.21	25.04	16.73
Complement Naive Bayes	32.08	32.05	35.85	45.58	24.69	25.02	23.89	16.76
FusedLogits LR	34.99	34.92	51.02	57.23	31.19	31.20	46.05	18.96
Factorization Machine (FM score)	36.41	35.82	50.80	56.77	31.96	31.28	45.38	20.51
Gradient Boosted Trees (Score/Logit)	34.54	34.58	51.00	55.61	32.28	31.88	48.16	20.17
Deep Learning Baselines								
Deep and Cross Network (DCN)	28.44 ± 0.44	28.73 ± 0.57	48.11 ± 0.41	51.97 ± 0.43	30.06 ± 0.62	29.24 ± 0.48	47.12 ± 0.23	19.70 ± 0.81
Deep MLP Bag-of-Codes (EmbeddingBag)	28.88 ± 0.21	29.13 ± 0.18	48.22 ± 0.29	52.04 ± 0.18	30.17 ± 0.98	29.40 ± 0.82	47.67 ± 0.72	20.18 ± 0.44
Star-GAT	26.97 ± 1.27	27.48 ± 1.23	47.19 ± 0.99	50.24 ± 0.68	29.44 ± 0.97	29.01 ± 0.91	48.25 ± 0.73	20.06 ± 0.76
Pure MIL Attention Pooling	29.25 ± 0.62	29.53 ± 0.43	48.37 ± 0.36	52.20 ± 0.63	31.11 ± 0.70	30.40 ± 0.31	48.11 ± 0.43	20.41 ± 0.53
Set Transformer (single-logit)	28.50 ± 0.68	29.02 ± 0.63	49.12 ± 0.37	52.04 ± 0.19	30.29 ± 0.62	29.92 ± 0.74	48.08 ± 0.43	20.51 ± 1.10
DeepSets (mean ⊕ max pool)	28.51 ± 0.33	28.84 ± 0.27	48.60 ± 0.60	51.92 ± 0.41	29.29 ± 0.49	29.11 ± 0.32	48.88 ± 0.43	**21.52 ± 0.26**
Our Model	**54.80 ± 1.40**	**49.42 ± 1.34**	**51.15 ± 0.88**	**61.97 ± 0.64**	**39.06 ± 3.27**	**37.39 ± 2.63**	**49.84 ± 1.38**	18.55 ± 0.19

Takeaway: MLCI shows the strongest dependence for mortality outcomes and length of stay, while also achieving the best ICU-transfer dependence on MIMIC-IV.

Experimental Results: Mutual Information (MI)

ICML International Conference On Machine Learning

Table 2. **Mutual Information between each model's unified risk score and clinical outcomes.** Higher is better. Bolded values highlight the best performer per outcome. *For readability, each column is scaled by the power of 10 shown in the header; divide by that factor to recover the original MI.*

Model	MIMIC-IV				MIMIC-III			
	MORT ($\times 10^3$)	**30M** ($\times 10^3$)	**LOS** ($\times 10^2$)	**ICU** ($\times 10^2$)	**MORT** ($\times 10^3$)	**30M** ($\times 10^3$)	**LOS** ($\times 10^2$)	**ICU** ($\times 10^3$)
Traditional Clinical Indices								
Charlson (CCI)	9.64	18.99	3.41	1.99	16.35	26.32	0.94	14.60
Elixhauser (ECI)	20.17	28.73	6.26	3.85	26.51	30.97	3.14	14.72
CCI+ECI (scalar)	19.94	28.66	6.39	4.86	30.99	36.11	3.40	16.52
Classical Machine Learning Baselines								
kNN	57.70	62.71	8.43	13.06	68.28	68.07	8.55	20.65
Multinomial Naive Bayes	56.44	57.74	7.90	13.37	57.28	54.45	7.27	22.29
Complement Naive Bayes	56.68	56.99	7.75	13.26	57.75	55.43	7.12	27.53
FusedLogits LR	53.55	54.33	13.75	17.60	56.87	53.66	13.25	28.19
Factorization Machine (FM score)	54.50	54.27	13.88	16.99	62.97	57.98	13.62	27.58
Gradient Boosted Trees (Score/Logit)	64.30	63.30	14.86	17.67	65.24	66.81	14.76	24.55
Deep Learning Baselines								
Deep and Cross Network (DCN)	65.33 ± 0.62	61.70 ± 0.24	14.48 ± 0.15	18.33 ± 0.33	65.46 ± 2.76	59.57 ± 1.60	13.47 ± 0.48	27.20 ± 0.64
Deep MLP Bag-of-Codes (EmbeddingBag)	65.88 ± 0.54	63.63 ± 1.19	14.73 ± 0.12	18.45 ± 0.06	66.20 ± 4.99	61.39 ± 3.21	13.83 ± 0.23	29.89 ± 2.13
Star-GAT	67.38 ± 2.48	65.32 ± 2.87	14.89 ± 0.27	18.89 ± 0.47	70.31 ± 2.27	63.74 ± 2.44	14.10 ± 0.54	29.28 ± 2.68
Pure MIL Attention Pooling	67.89 ± 0.94	65.16 ± 2.32	14.93 ± 0.04	18.71 ± 0.12	71.72 ± 1.57	64.64 ± 0.79	14.32 ± 0.37	26.14 ± 2.26
Set Transformer (single-logit)	68.35 ± 2.40	65.84 ± 3.19	15.16 ± 0.18	18.59 ± 0.25	75.15 ± 1.62	69.23 ± 1.95	14.54 ± 0.31	29.64 ± 3.14
DeepSets (mean ⊕ max pool)	69.92 ± 1.26	67.22 ± 1.60	15.36 ± 0.11	**19.24 ± 0.19**	73.69 ± 3.29	68.08 ± 1.88	15.17 ± 0.30	**32.42 ± 4.22**
Our Model	**74.22 ± 0.24**	**72.19 ± 0.93**	**15.42 ± 0.18**	19.01 ± 0.29	**84.52 ± 10.89**	**77.77 ± 7.53**	**16.45 ± 0.92**	26.42 ± 3.77

Takeaway: Mutual information shows the same overall trend.

Theory Diagnostics

We test the theory on MIMIC-IV and MIMIC-III using two diagnostics.

Q1: Does a shared severity signal appear in real clinical outcomes?

Rank-one alignment: Measures how much of the multi-outcome label structure is explained by one shared direction.

Table 3. Rank-one alignment and objective values on learned scores. $J(s)$ is the full multi-task nHSIC objective evaluated on the learned score vector; $J_1(s)$ is the rank-one projected objective.

Dataset	$\sigma_{1:4}$ of $\widetilde{W}$	$\sigma_1^2/\|\widetilde{W}\|_F^2$	$J(s)$	$J_1(s)$
MIMIC-IV	(1.40, 1.05, 0.81, 0.53)	0.49	0.622	0.397
MIMIC-III	(1.34, 1.14, 0.84, 0.47)	0.45	0.453	0.152

Finding #1: The stacked outcome labels show moderate rank-one structure, indicating a meaningful shared severity signal across outcomes.

Q2: Can the shared direction v recover the optimal high-severity cutoff?

Threshold agreement: We compare $\boldsymbol{j_v}$, the cutoff selected by the shared direction v, with $\boldsymbol{j_J}$, the best cutoff for the full multi-outcome objective.

Table 4. Two-level monotone threshold diagnostics. j_J is the best split for the full multi-task threshold objective within the threshold family; j_v maximizes $\rho_j^2(v)$, equivalently selecting the best threshold for the rank-one projected objective.

Data	j_J	j_v	Tail size	$\rho_\star^2(v)$
MIMIC-IV	4862	4862	137/5000 (2.74%)	0.554
MIMIC-III	11265	11265	643/11909 (5.40%)	0.220

Finding #2: The cutoff selected by the shared direction matches the best full-objective cutoff in both datasets, showing that v identifies a compact high-severity group.

Limitations of MLCI

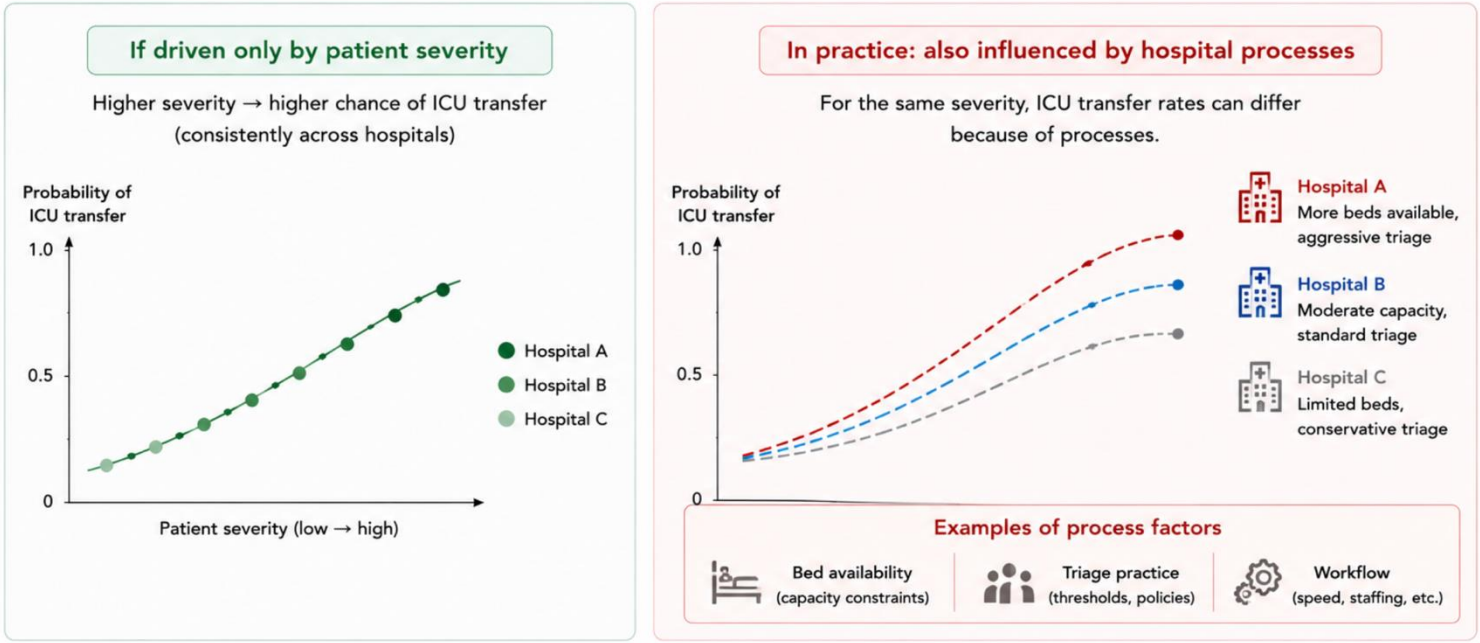
MLCI has two main limitations:

1. Shared severity assumption

MLCI assumes outcomes share a common severity signal. This assumption may be weaker for clinical outcomes driven by task-specific factors.

2. Some outcomes reflect hospital processes

For example, ICU transfer can depend on bed availability, triage practice, and workflow, rather than only patient severity.



The University of Iowa

comp|epi
computational epidemiology research

ICML
International Conference
On Machine Learning

Thank you for your time!

This project is funded by the United States National Science Foundation (NSF).

Connect
with me!