Trustworthy By Construction

AHLI Health AI Summer Camp 2026

Walter Gerych

The Methodology Development Pipeline

Observe failure

↓

Identify mechanism

↓

Analyze assumptions

↓

Build theory

↓

Apply theory to build robust model

One Foot in Application & One Foot In Abstraction

Outline

I will be illustrating my approach for methodology development using two “case studies”, before moving on to general lessons:

- Part 1: When Missing Data Isn't Random
- Part 2: When “Debiasing” Creates New Bias
- Part 3: General Lessons for Building Methodologies

Part 1: When Missing Data Isn't Random

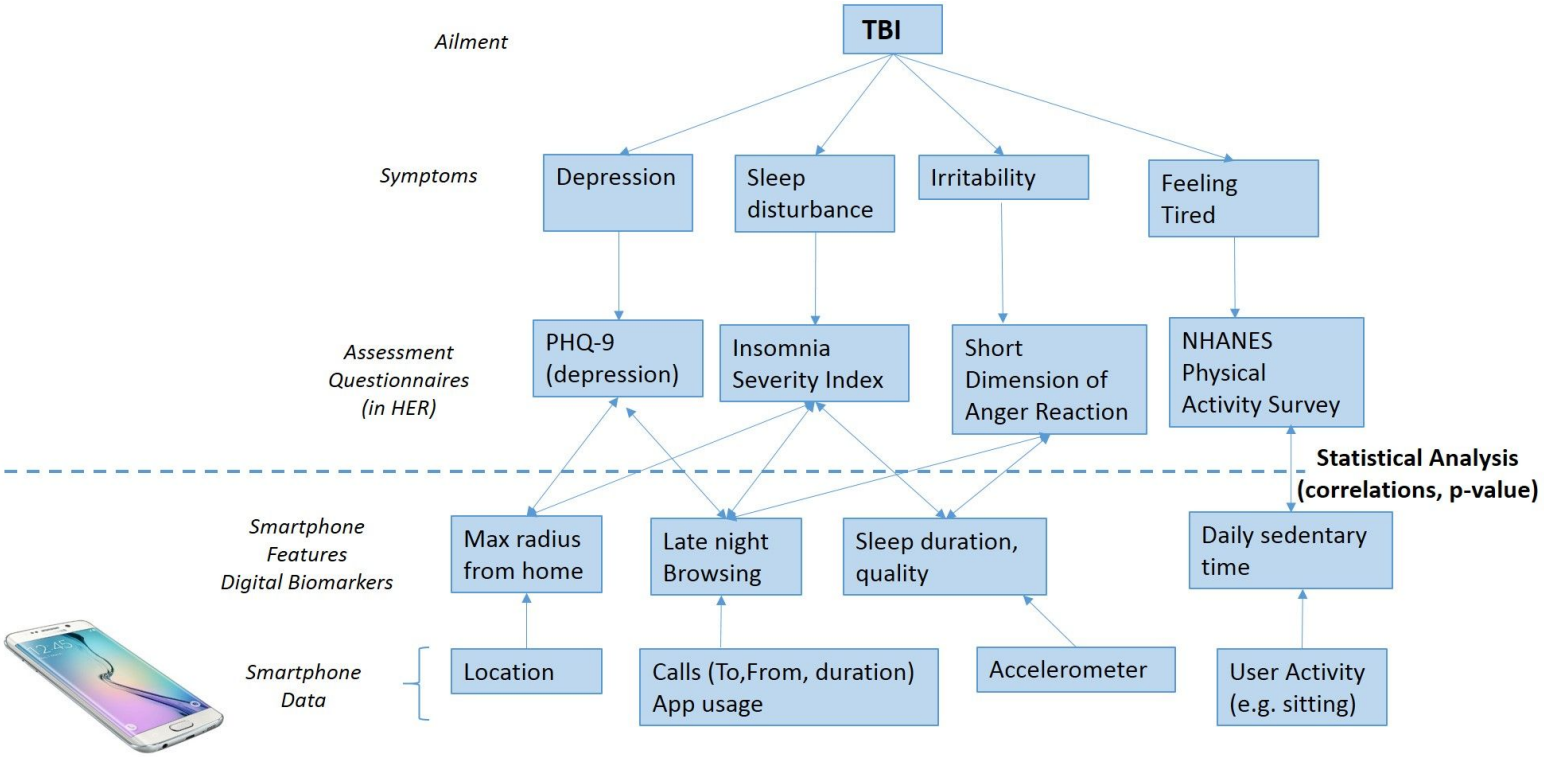
Based on the following works:

Gerych, Walter, Tom Hartvigsen, Luke Buquicchio, Abdulaziz Alajaji, Kavin Chandrasekaran, Hamid Mansoor, Elke Rundensteiner, and Emmanuel Agu. "Positive unlabeled learning with a sequential selection bias." SDM 2022.

Gerych, Walter, Thomas Hartvigsen, Luke Buquicchio, Emmanuel Agu, and Elke Rundensteiner. "Recovering the propensity score from biased positive unlabeled data." AAAI, 2022.

Nagaraj, Sujay, Walter Gerych, Sana Tonekaboni, Anna Goldenberg, Berk Ustun, and Thomas Hartvigsen. "Learning under temporal label noise." ICLR, 2025.

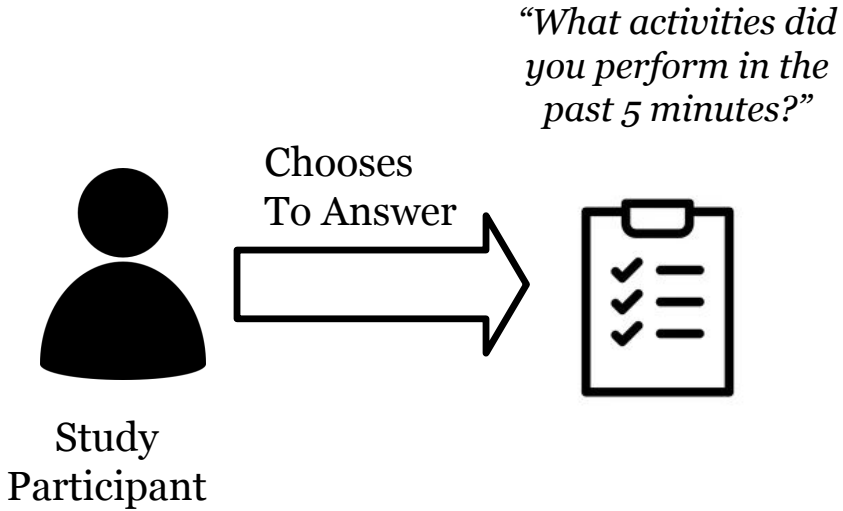
The Data: Smartphone Sensor Data For Digital Biomarkers



https://wash.wpi.edu/

Noisy Data Collection

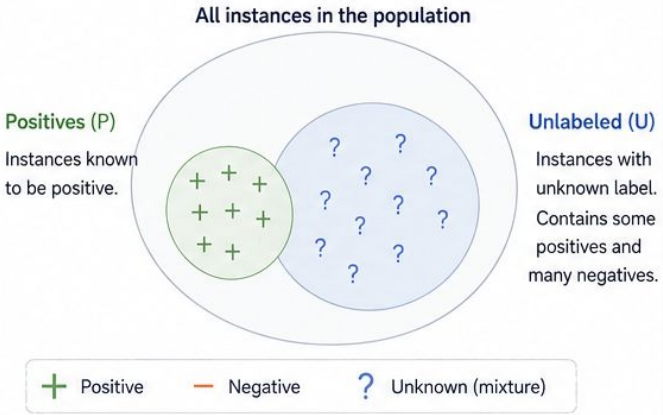
- We collected “in-the-wild” data
- Study participants self-reported their own activities over 2-week study
- Large chunks of the time series were totally unlabeled
- Our data was “positive unlabeled”



Positive-Unlabeled Data

We have reliable positive labels, but the unlabeled data contains a mix of positives and negatives.

1. What is PU data?



Key point
U is not "all negatives". It is a mixture of positives and negatives.
We only know for sure the positives.

2. Examples

COVID-19 diagnosis (early data)
P: patients with confirmed COVID-19 by PCR
U: patients tested but not confirmed (contains some infected, many not infected)

Breast cancer screening
P: patients with biopsy-confirmed breast cancer
U: patients with negative or unknown biopsy results (contains some cancers)

Sepsis prediction in ICU
P: ICU stays with confirmed sepsis
U: ICU stays without sepsis label (contain some sepsis cases, many not sepsis)

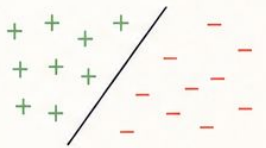
Stroke detection from imaging
P: scans with radiologist-confirmed stroke
U: scans without stroke annotation (contains some strokes)

3. How learning works (intuition)

1. Learn to distinguish positives from the unlabeled set.

2. Estimate how many positives are hidden inside U (class prior $\pi = P(Y=1)$).

3. Correct for the selection bias and recover a classifier for positive vs. negative.



Common PU Assumptions

PU (SCAR) assumption

The probability of an instance being included in P does not depend on its features, given that it is positive.

$P(s = 1 \mid x, y = 1) = P(s = 1 \mid y = 1)$

SAR assumption

The labeling probability can depend on the features.

$P(s = 1 \mid x, y = 1)$ can vary with x.

Positivity assumption

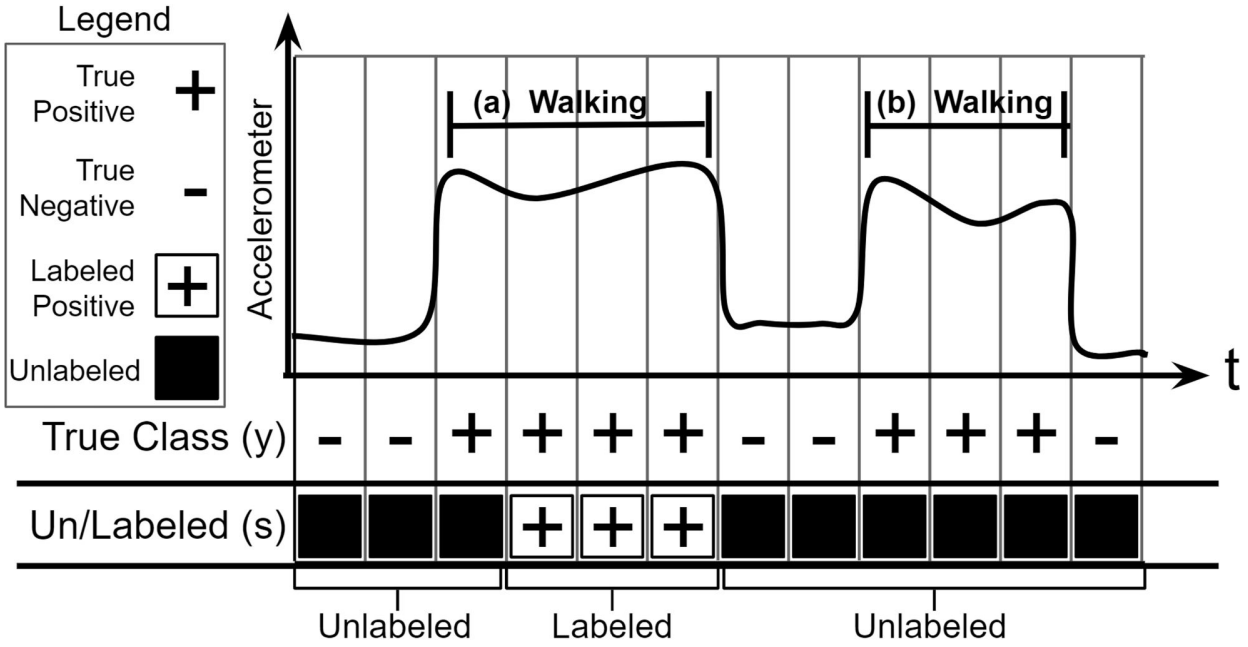
Every positive instance has a non-zero chance to be labeled as positive.

$P(s = 1 \mid x, y = 1) > 0$ for all x

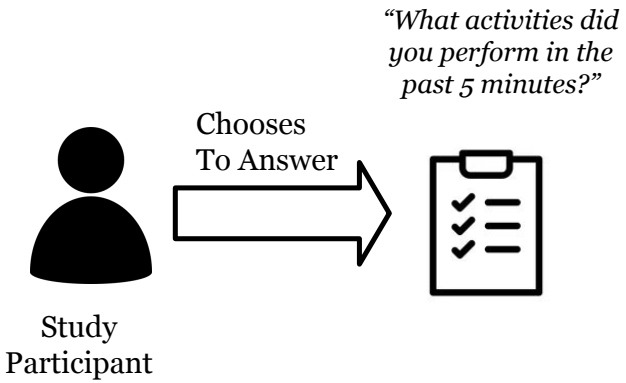
$s = 1$ if an instance is labeled (observed in P); $s = 0$ if unlabeled (in U)
$y = 1$ for positive; $y = 0$ for negative; x are features

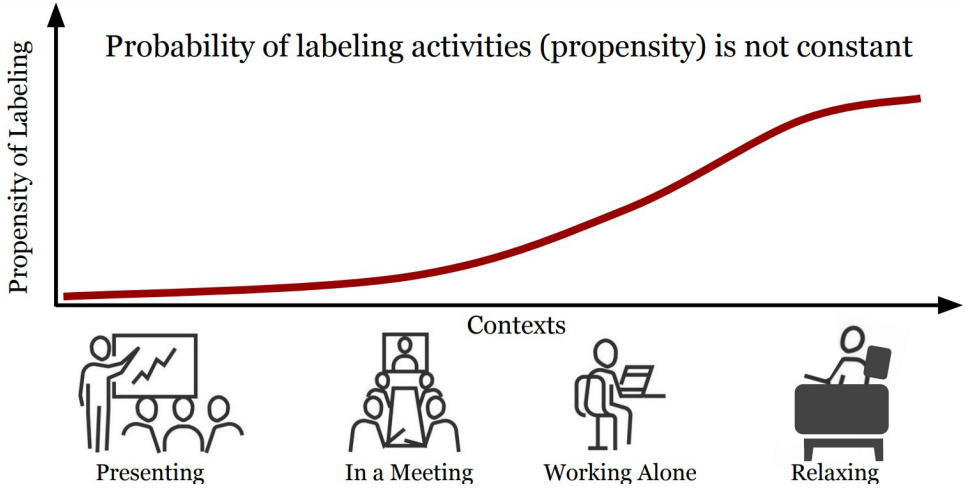
But Our Missingness Wasn't "Random"!

We observed a clear *temporal* aspect for the labeling patterns:

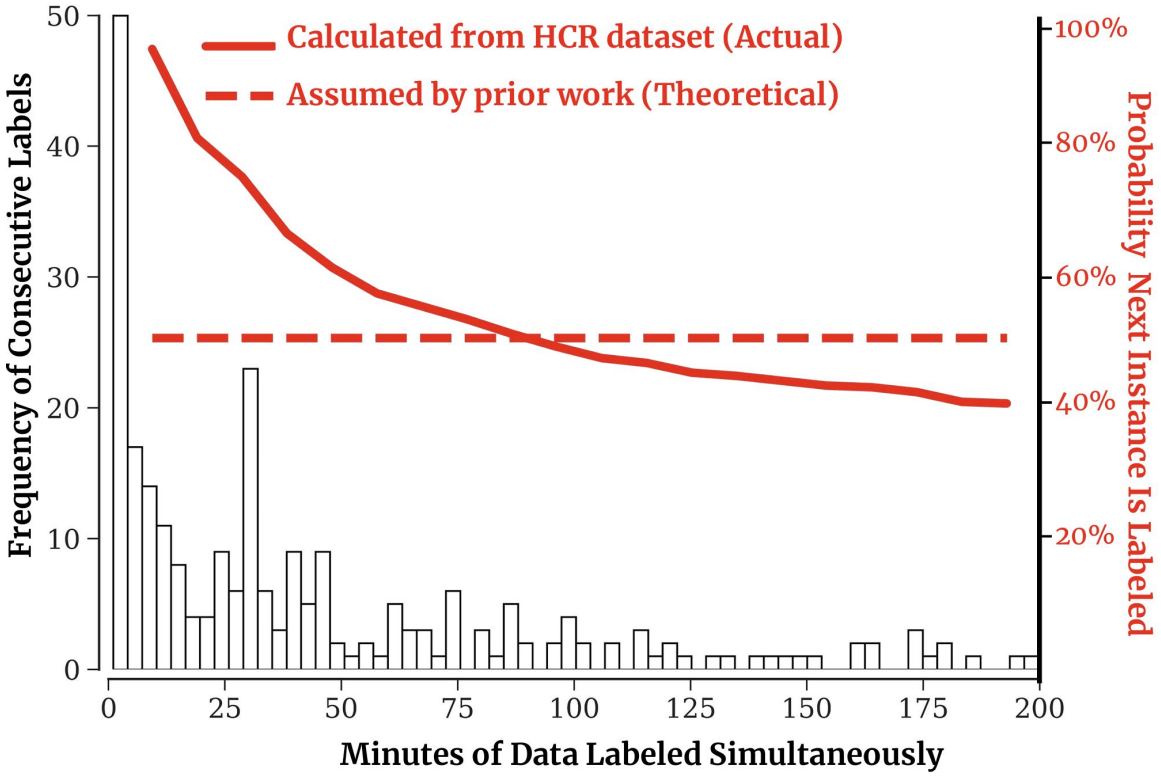


What's The Cause Of "Burst" Labeling?





Existing Methods Couldn't Model Our Data



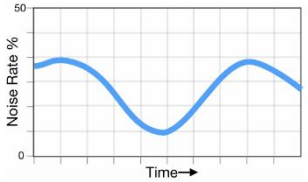
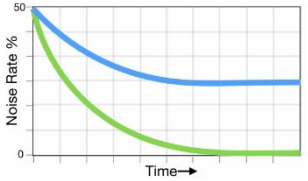
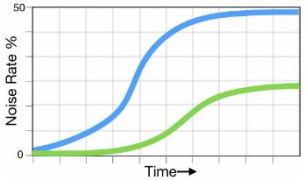
This Type of Labeling Wasn't Specific To Our Dataset

- Human Activity Recognition:
  - Wearable device studies often ask participants to annotate their activities (e.g., exercise) over time. Participants often mislabel their activities due to recall bias, time of day, or labeling-at-random for monetized studies

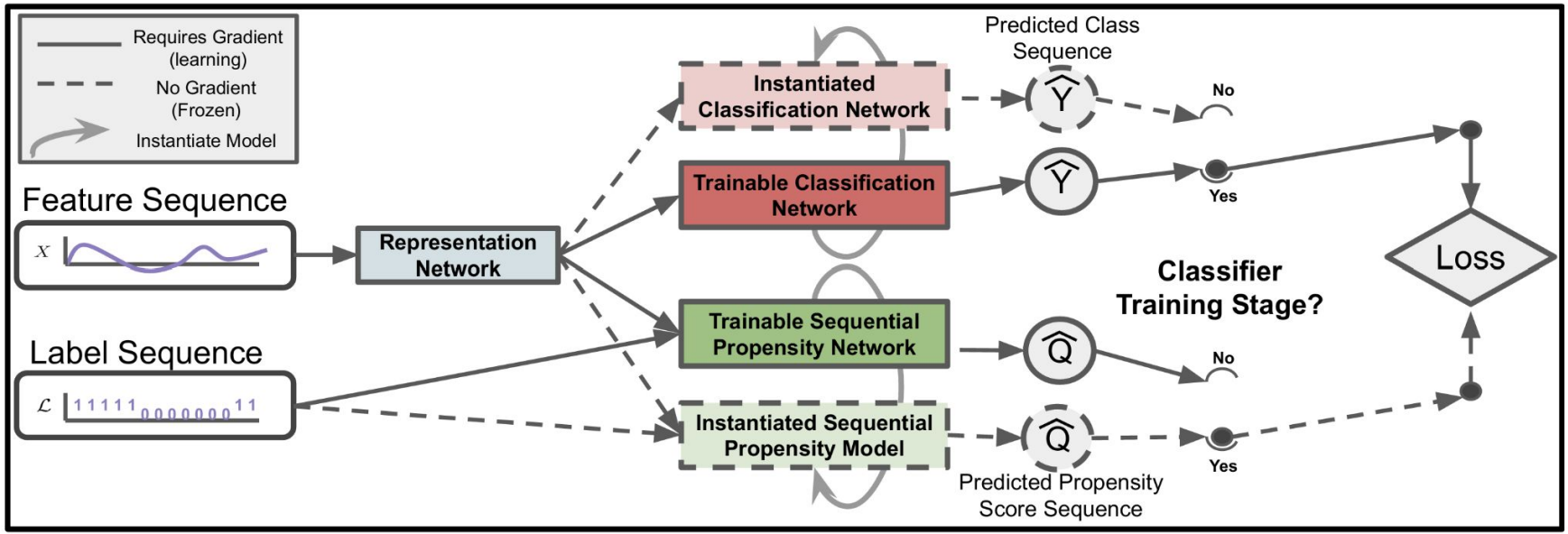
- Self-Reported Outcomes for Mental Health:
  - Mental health studies often collect self-reported survey data (e.g. depression) over long periods of time. Such self-reporting is known to be biased, as participants are more or less likely to report certain outcomes.

- Clinical Measurement Error:
  - Clinical prediction models often predict outcomes (e.g., mortality) derived from clinician notes in electronic health records. These labels may capture noisier annotations during busier times; e.g., when a patient is deteriorating.

Generalizing Temporal Label Noise

Pattern	Depiction	Noise Model Q_{ij}	Parameters	Applications
Periodic		$\frac{1}{2} + \frac{1}{2}\sin(\alpha_{ij}t + \phi_{ij})$	$\omega_{ij} = (\alpha_{ij}, \phi_{ij})$ α controls frequency ϕ controls shift	Annotation reliability **varies over time of day** – e.g., due to changes in annotator attentiveness over the day [16].
Decay		$\alpha_{ij}\exp(-\beta_{ij}t)$	$\omega_{ij} = (\alpha_{ij}, \lambda_{ij})$ α controls initial noise β controls decay rate	Annotation reliability **improves rapidly** (e.g., diagnostic accuracy of COVID-19 rapidly improved at the onset of the pandemic [51]) or **improves and plateaus** (e.g., irreducible uncertainty in labels [54])
Growth		$\frac{\alpha_{ij}}{1+\exp(-\beta_{ij}(t-\gamma_{ij}))}$	$\omega_{ij} = (\alpha_{ij}, \beta_{ij}, \gamma_{ij})$ α controls limit β controls growth rate γ controls inflection point	Annotation reliability **decreases abruptly** – e.g., due to rapid adoption of improved clinical guidelines [66]. Or, it **decreases gradually** – e.g., due to underdiagnosis during the pandemic due to healthcare disruptions [18]

Our Initial “Ad Hoc” Solution



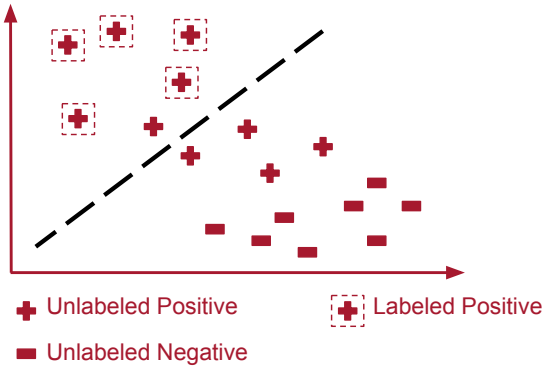
- We first proposed an EM-like methodology for jointly modeling a time-series classifier along with a temporal noise correction model

Our Initial “Ad Hoc” Solution

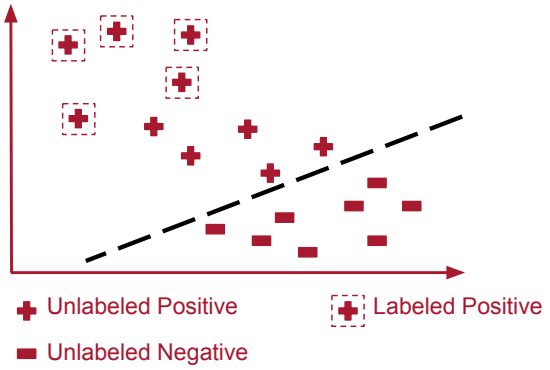
- Our first proposed method had decent results, but had some flaws
- We did not have guarantees that the temporal noise model would learn the true noise function (not very trustworthy!)
- We didn’t have a clear idea of when the model would work well, and when it would not!
- **Initial methodology creates new methodology questions!**
  - **This leads to *theory***

When Can We Accurately Model The Propensity Score?

Propensity Score **Unknown** → **Biased** Classifier



Propensity Score **Known** → **Unbiased** Classifier



It is not known when the propensity score is **identifiable**:
When is it possible to build an accurate data-driven model of the propensity score?

Common (Unbiased) PU Assumptions

Local Certainty/Separable Classes

- Bayes Error of 0 between positive and negative distributions

Positive Subdomain

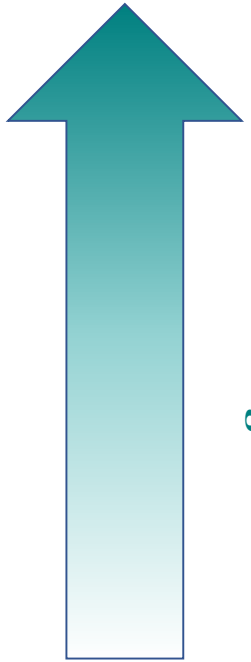
- There is some region A of the feature space determined by partial attribute assignment such that the Bayes error is 0

Positive Function

- There is some region A of the feature space determined by an arbitrary function for which the Bayes error is 0

Irreducibility

- The negative distribution is not a mixture containing the positive distribution



We Proved That The Labeling Mechanism Is Often *Not* Identifiable!

Theorem 1 *Let propensity score e be an arbitrary function of x, $e : \mathcal{X} \to (0, 1]$. Let the PU assumption hold (y is unobserved, ℓ and x are observed). Then, e is non-identifiable under the Positive Subdomain, Positive Function, and Irreducibly scenarios.*

Positive Subdomain

- There is some region A of the feature space determined by partial attribute assignment such that the Bayes error is 0

Positive Function

- There is some region A of the feature space determined by an arbitrary function for which the Bayes error is 0

Irreducibility

- The negative distribution is not a mixture containing the positive distribution

Where We Could Identify The Propensity Function

We also identified settings where the labeling mechanism could be learned from data:

Local Certainty/Separable Classes

- Bayes Error of 0 between positive and negative distributions

Propensity Follows Likelihood

- How likely a point is to be labeled tracks how likely the point is to be in the positive class

Temporal Minimum Volume Simplex

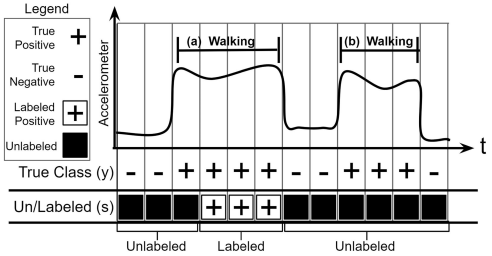
- At each point in time, the noise process is the tightest explanation for what we observe at that moment

Algorithms for learning the labeling mechanism in these settings were naturally obvious from the proofs of their existence!

Zooming Out For A Minute...

We went from...

...observation about our real-world, domain-specific data...



...to general theories about the when we can (and can't) model labeling mechanisms...

Theorem 1 *Let propensity score e be an arbitrary function of x, $e : \mathcal{X} \to (0, 1]$. Let the PU assumption hold (y is unobserved, ℓ and x are observed). Then, e is non-identifiable under the Positive Subdomain, Positive Function, and Irreducibly scenarios.*

...to a better methodology that addressed our core, real-world data labeling issues

			Static		*Temporal*		
Dataset	Metric	Ignore	Anchor	VolMinNet	Plug-In	Discontinuous	Continuous
`moving` [59]	Test Error	$29.4 \pm 1.7\%$	$20.9 \pm 2.6\%$	$14.9 \pm 2.7\%$	$20.0 \pm 1.8\%$	$15.9 \pm 2.7\%$	**$4.2 \pm 2.2\%$**
$n = 192, d = 14, T = 50$	Approx. Error	–	$42.4 \pm 3.4\%$	$35.3 \pm 0.8\%$	$36.8 \pm 1.7\%$	$32.6 \pm 0.5\%$	**$10.3 \pm 3.9\%$**
`senior` [44]	Test Error	$22.7 \pm 1.7\%$	$20.7 \pm .01\%$	$19.0 \pm 0.7\%$	$18.8 \pm 1.1\%$	$13.6 \pm 1.2\%$	**$11.0 \pm 0.3\%$**
$n = 444, d = 6, T = 100$	Approx. Error	–	$35.9 \pm 2.5\%$	$36.3 \pm 0.4\%$	$26.9 \pm 1.2\%$	$21.7 \pm 0.2\%$	**$6.4 \pm 0.8\%$**
`blinking` [60]	Test Error	$34.1 \pm 2.0\%$	$34.1 \pm 2.3\%$	$29.6 \pm 2.2\%$	$29.6 \pm 2.8\%$	$29.9 \pm 3.0\%$	**$29.6 \pm 2.3\%$**
$n = 299, d = 14, T = 50$	Approx. Error	–	$35.3 \pm 0.8\%$	$35.2 \pm 0.7\%$	$19.6 \pm 1.1\%$	$26.6 \pm 0.9\%$	**$14.9 \pm 2.3\%$**
`sleeping` [22]	Test Error	$28.7 \pm 0.8\%$	$24.9 \pm 1.1\%$	$26.8 \pm 1.4\%$	$20.4 \pm 1.8\%$	$19.6 \pm 0.8\%$	**$16.3 \pm 0.4\%$**
$n = 964, d = 7, T = 100$	Approx. Error	–	$34.3 \pm 1.8\%$	$41.8 \pm 0.1\%$	$19.1 \pm 3.5\%$	$22.4 \pm 0.2\%$	**$4.9 \pm 0.5\%$**

The Methodology Development Pipeline

Observe failure

↓

Identify mechanism

↓

Analyze assumptions

↓

Build theory

↓

Apply theory to build robust model

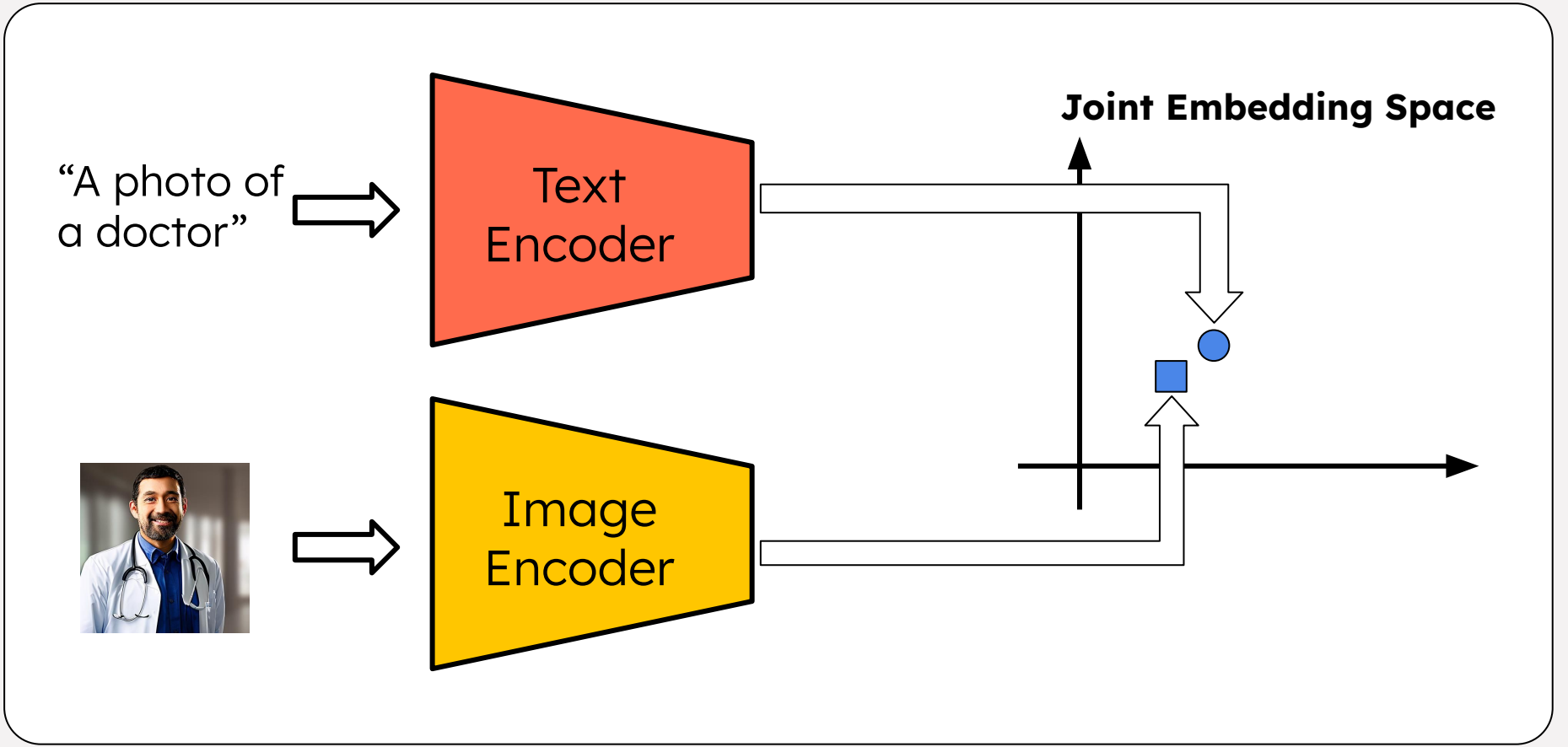
Part 2: When “Debiasing” Creates New Bias

Based on the following works:

Gerych, Walter, Haoran Zhang, Kimia Hamidieh, Eileen Pan, Maanas Sharma, Thomas Hartvigsen, and Marzyeh Ghassemi. "Bendvlm: Test-time debiasing of vision-language embeddings." NeurIPS, 2024.

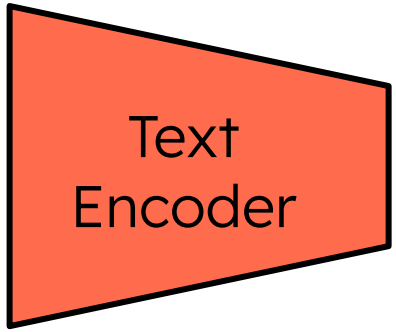
Gerych, Walter, Cassandra Parent, Quinn Perian, Rafiya Javed, Justin Solomon, and Marzyeh Ghassemi. "WRING Out The Bias: A Rotation-Based Alternative To Projection Debiasing." ICLR, 2026.

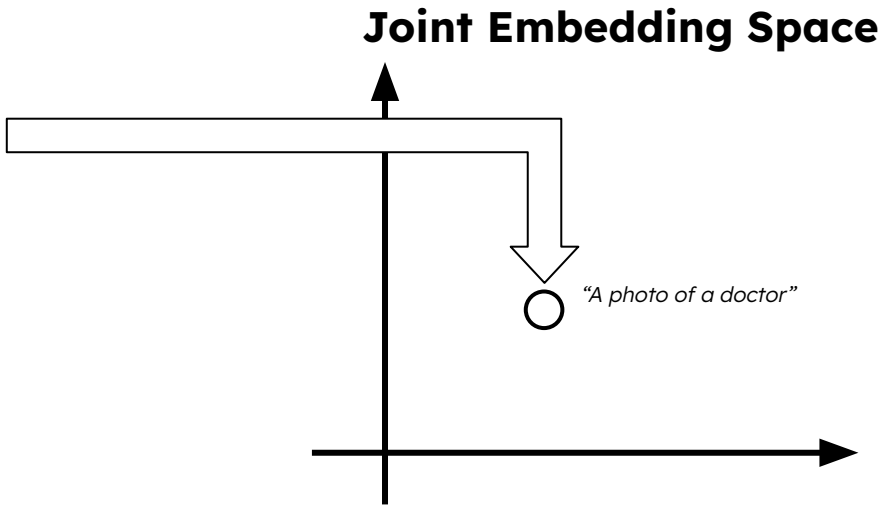
Vision-Language Embedding Models

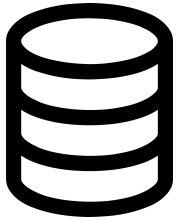


Vision-Language Embedding Models

Query:
“A photo of a doctor”

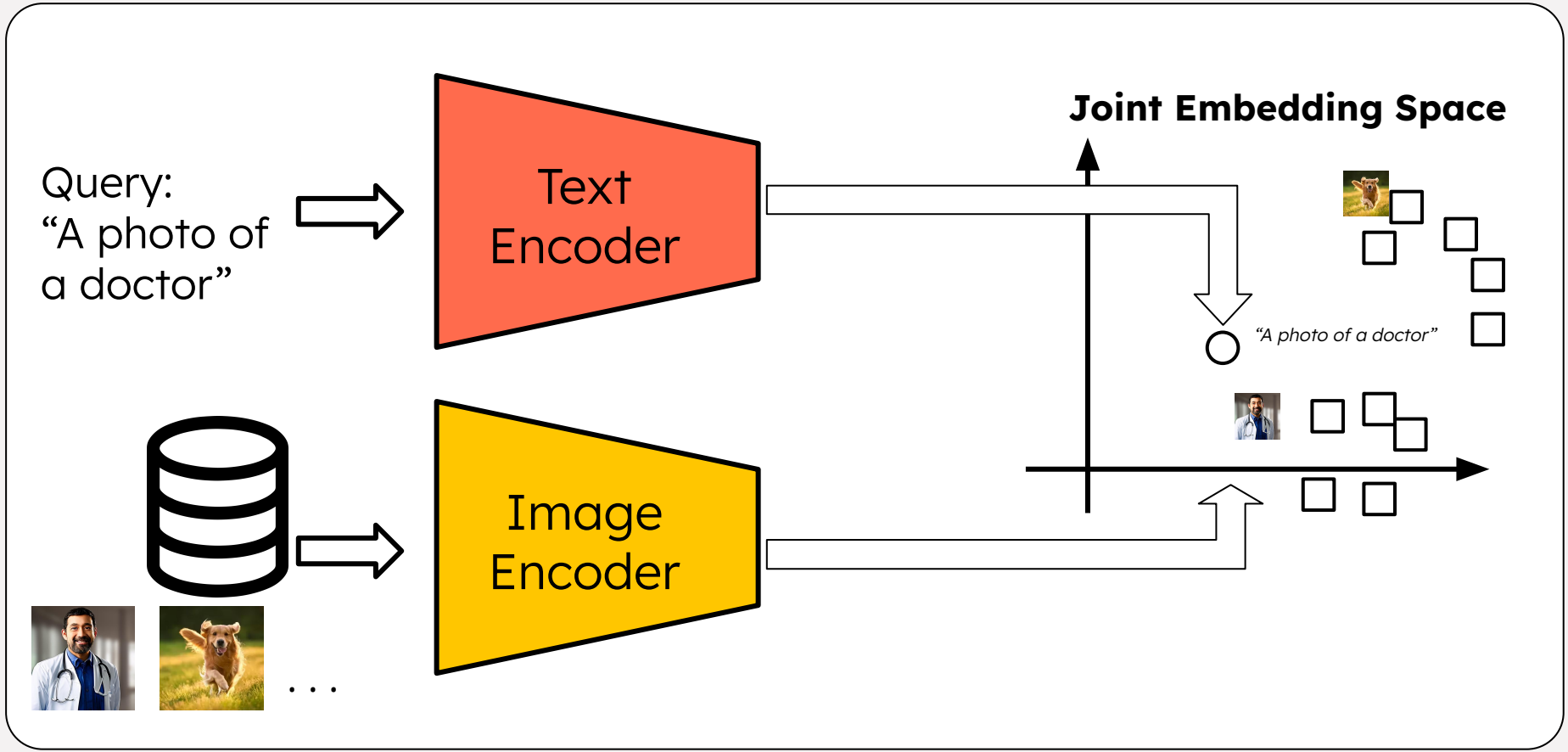




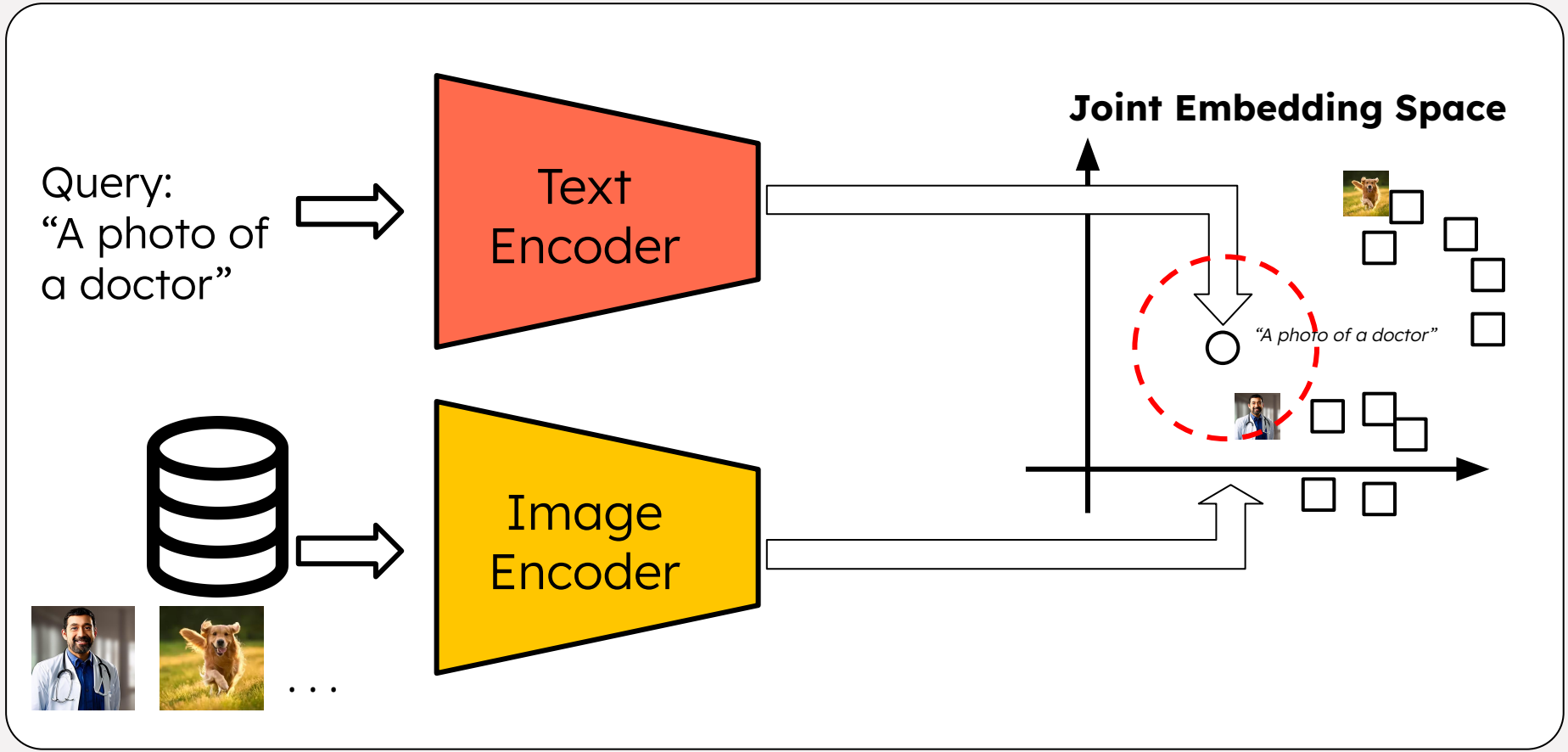


. . .

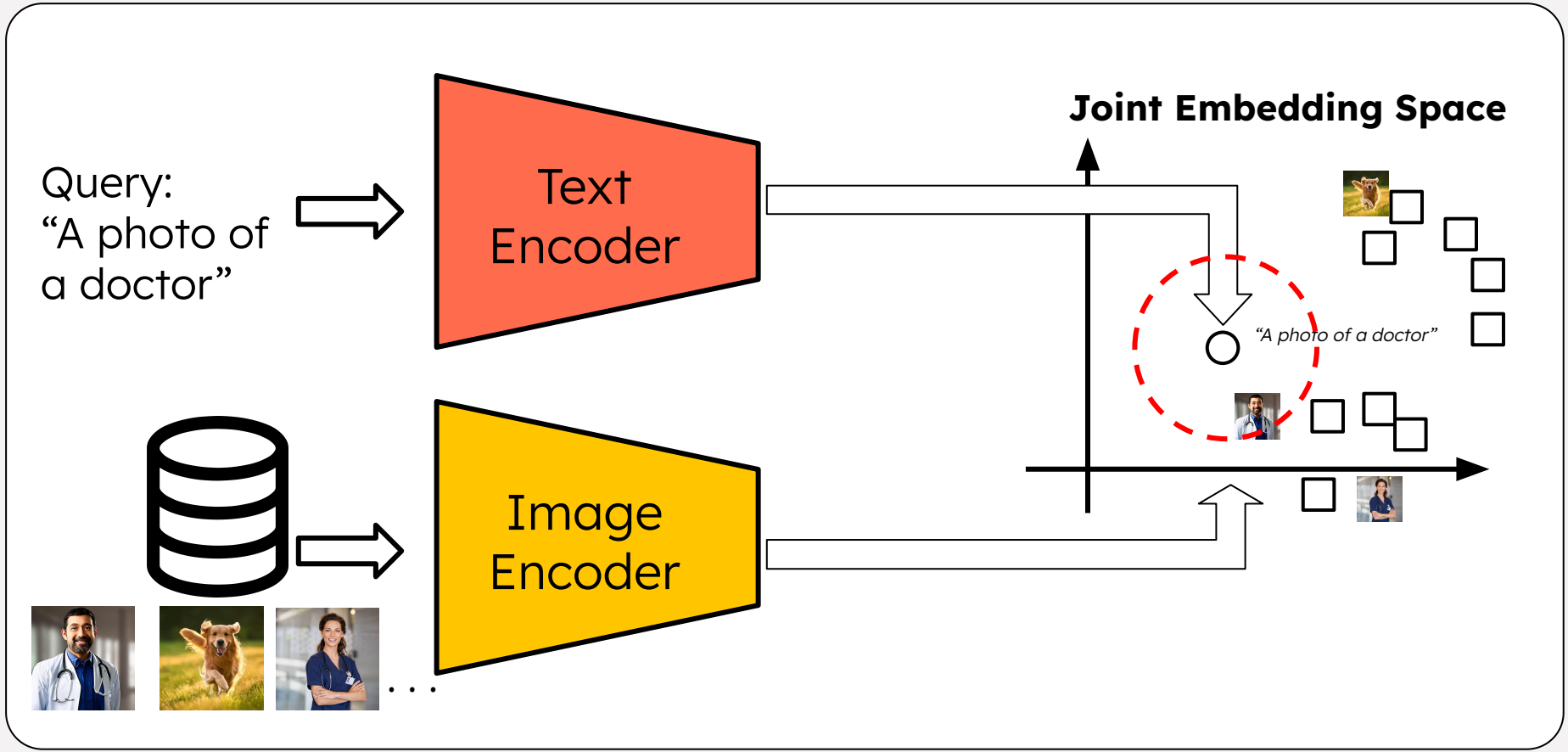
Vision-Language Embedding Models



Vision-Language Embedding Models



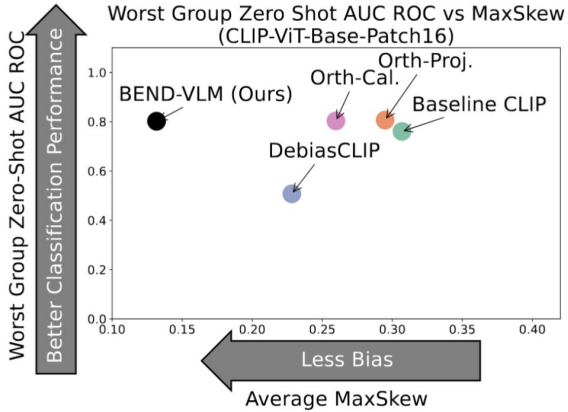
Vision-Language Embedding Models

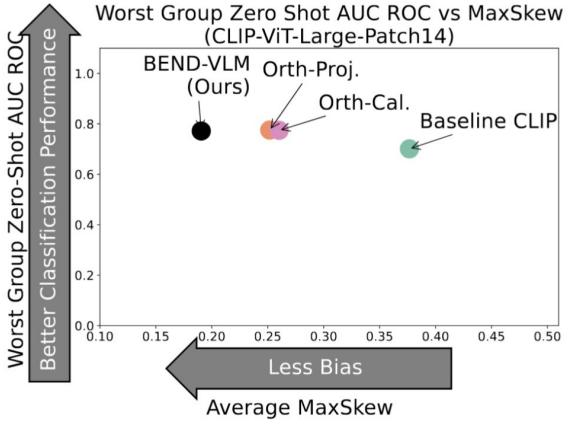


VLM Debiasing

We had recently developed a (nonlinear) method for debiasing VLMs

And it worked pretty well!





But we noticed a flaw…

Bias Amplification

But we noticed a flaw: Bias for the target (e.g. gender) went down, but bias for “unconsidered” concepts (e.g. race) often increased!

Method	KL Divergence ↓	MaxSkew ↓
Baseline CLIP	0.606 ± 0.043	0.155 ± 0.016
Orth-Proj.	0.826 ± 0.020	0.211 ± 0.014
Orth-Cal.	0.877 ± 0.021	0.226 ± 0.005
Bend-VLM	0.837 ± 0.035	0.193 ± 0.024

“Whac-A-Mole”

At first, we wrote this off as just another example of the “whac-a-mole” dilemma:

(a) We construct UrbanCars, a new dataset with multiple shortcuts, facilitating the study of multi-shortcut learning under the *controlled setting*.

(b) We discover the new watermark shortcut emerged from a *natural image* dataset—ImageNet, and create ImageNet-W test set for ImageNet.

Figure 1. Our benchmark results on both datasets reveal the overlooked Whac-A-Mole dilemma in shortcut mitigation, *i.e.*, mitigating one shortcut amplifies the reliance on other shortcuts.

Li, Zhiheng, et al. "A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others." CVPR 2023.

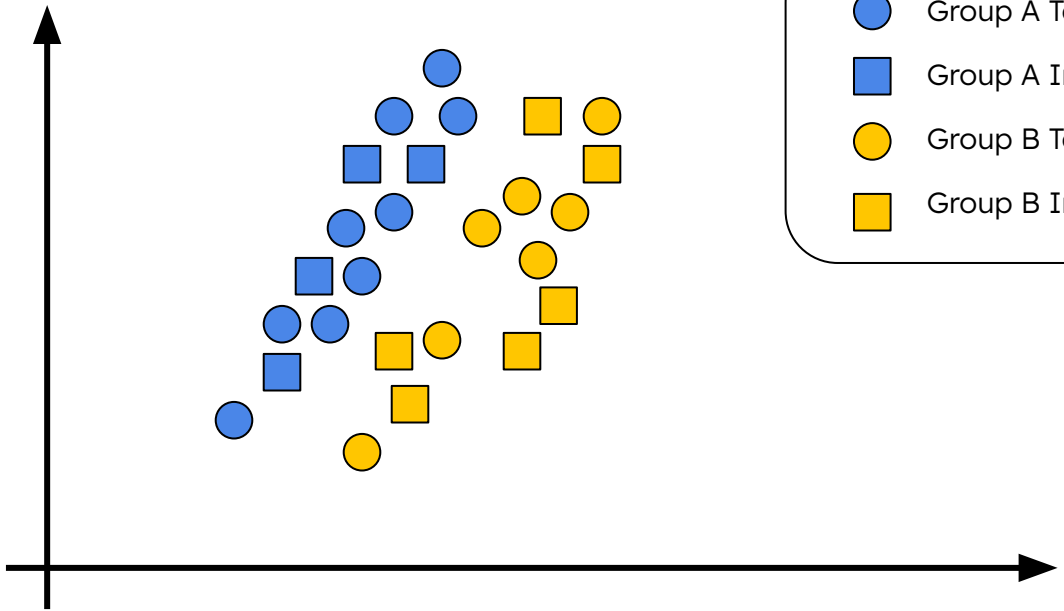
Bias Amplification

But we noticed a weird pattern when we performed an ablation experiment:

Method	KL Divergence ↓	MaxSkew ↓
Baseline CLIP	0.606 ± 0.043	0.155 ± 0.016
Orth-Proj.	0.826 ± 0.020	0.211 ± 0.014
Orth-Cal.	0.877 ± 0.021	0.226 ± 0.005
Bend-VLM (Without Step 1)	0.594 ± 0.074	0.146 ± 0.029
Bend-VLM (Without Step 2)	0.873 ± 0.024	0.223 ± 0.006
Bend-VLM (Full Method)	0.837 ± 0.035	0.193 ± 0.024

The “whac-a-mole” effect was coming from a particular step in our method; a *projection* operation

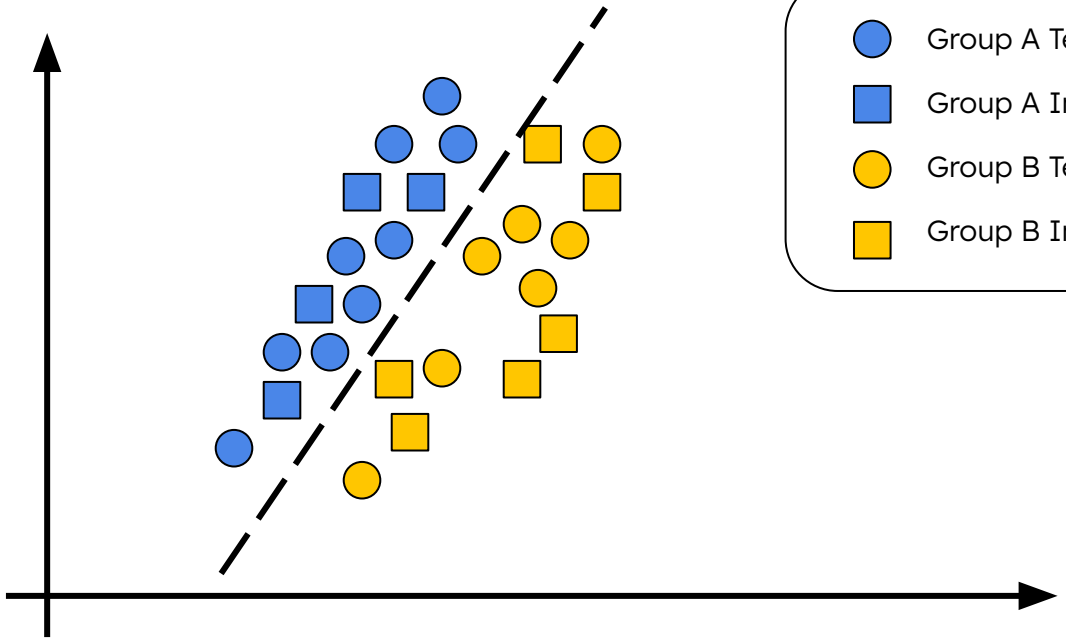
Linear Projection Debiasing

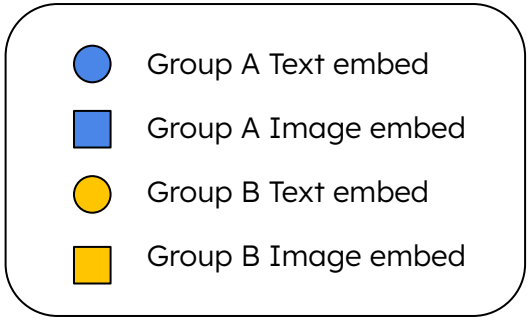


Group A Text embed
Group A Image embed
Group B Text embed
Group B Image embed

E.g.
Group A = Male,
Group B = Female,
Concept = Gender

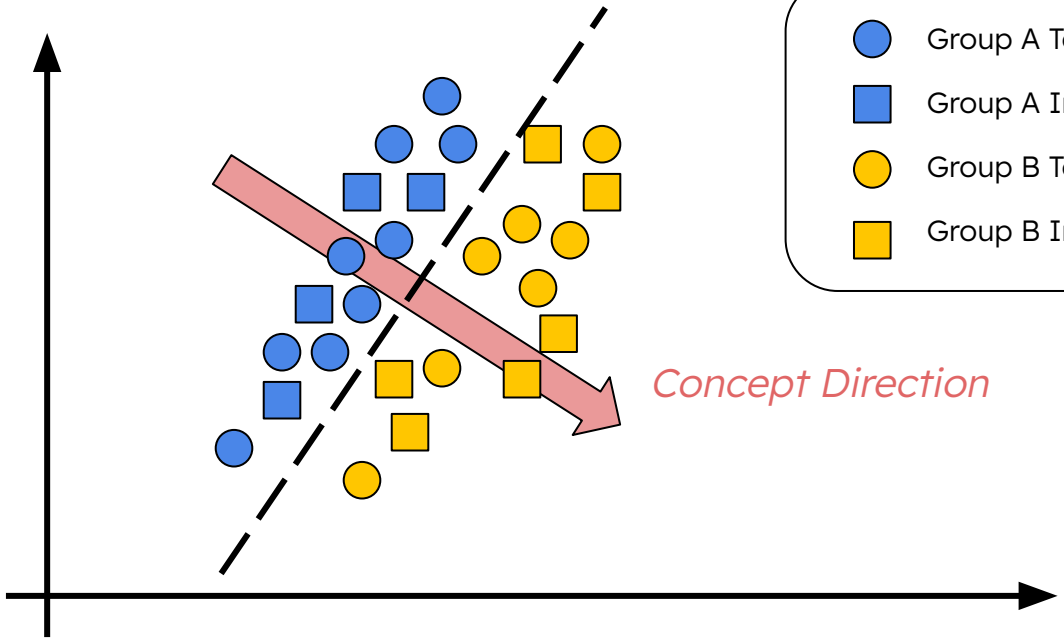
Linear Projection Debiasing





E.g.
Group A = Male,
Group B = Female,
Concept = Gender

Linear Projection Debiasing



Group A Text embed

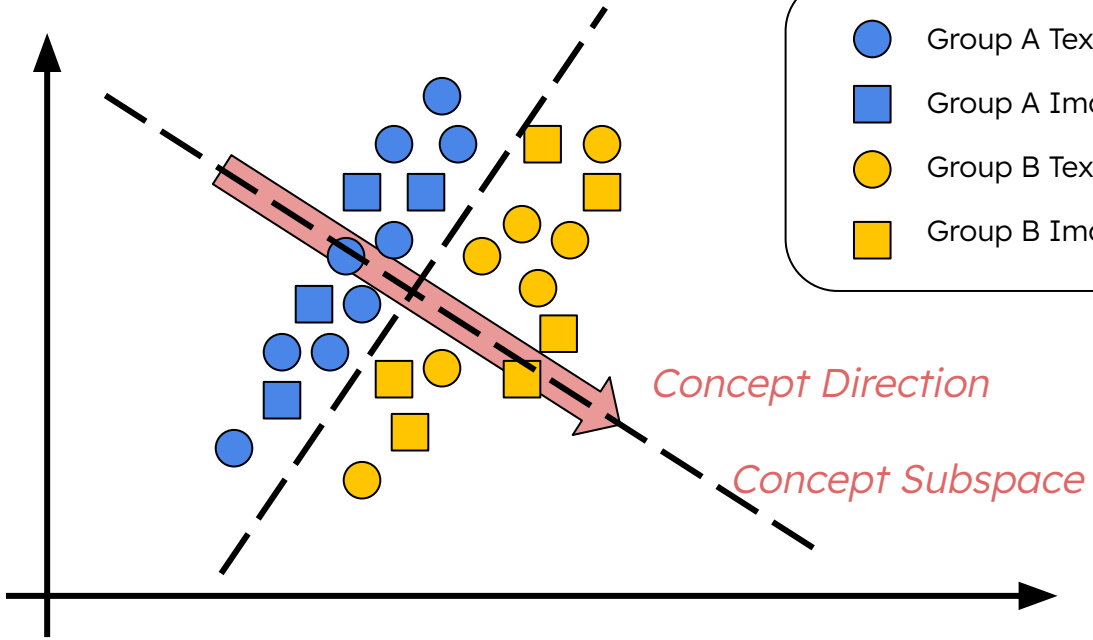
Group A Image embed

Group B Text embed

Group B Image embed

E.g.
Group A = Male,
Group B = Female,
Concept = Gender

Linear Projection Debiasing



Group A Text embed

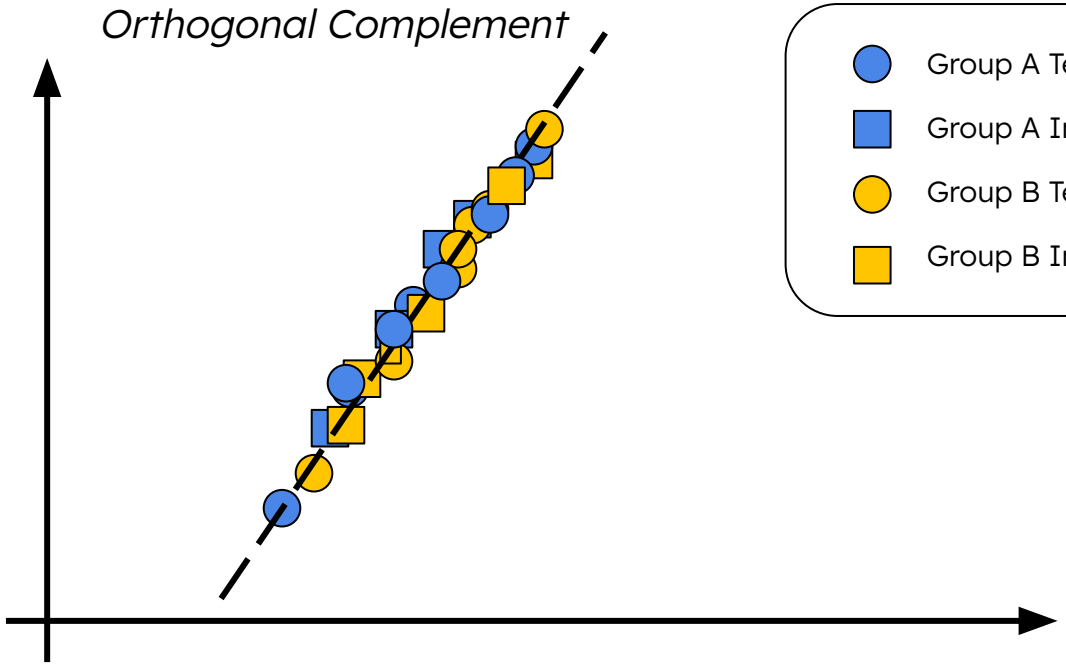
Group A Image embed

Group B Text embed

Group B Image embed

E.g.
Group A = Male,
Group B = Female,
Concept = Gender

Linear Projection Debiasing



Group A Text embed

Group A Image embed

Group B Text embed

Group B Image embed

E.g.
Group A = Male,
Group B = Female,
Concept = Gender

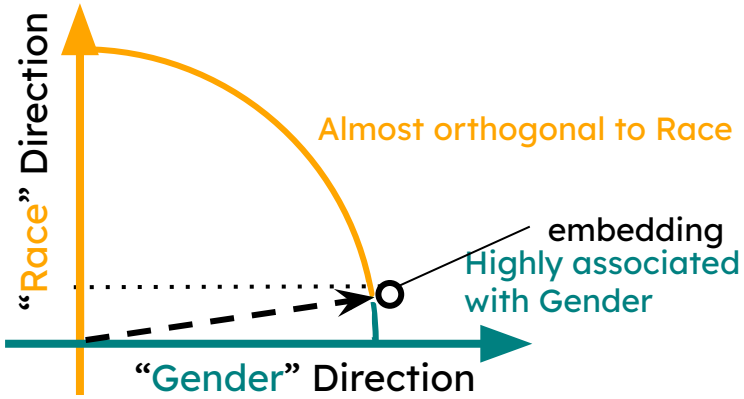
Bias Amplification

So, projection leads to whac-a-mole. We could stop here.

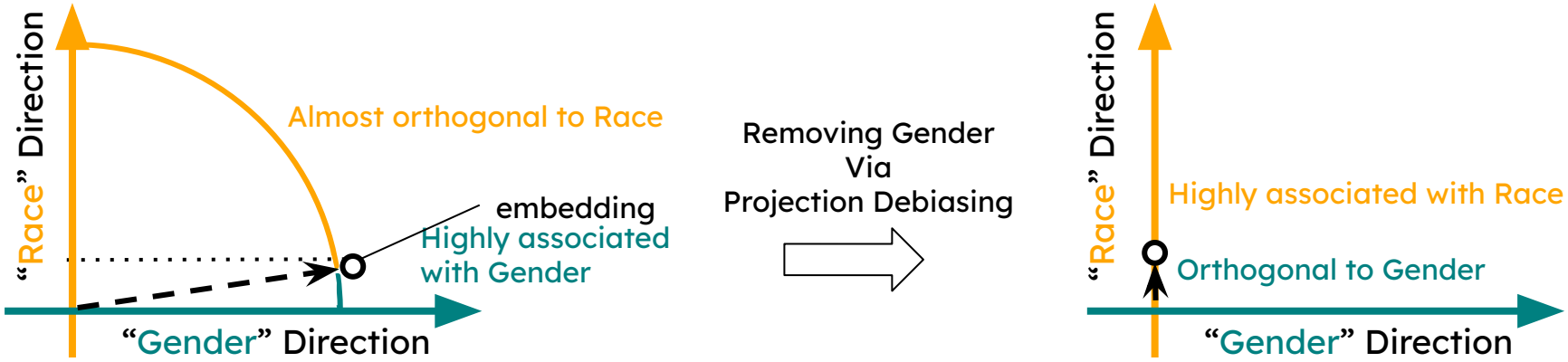
Method	KL Divergence ↓	MaxSkew ↓
Baseline CLIP	0.606 ± 0.043	0.155 ± 0.016
Orth-Proj.	0.826 ± 0.020	0.211 ± 0.014
Orth-Cal.	0.877 ± 0.021	0.226 ± 0.005
Bend-VLM (Without Step 1)	0.594 ± 0.074	0.146 ± 0.029
Bend-VLM (Without Step 2)	0.873 ± 0.024	0.223 ± 0.006
Bend-VLM (Full Method)	0.837 ± 0.035	0.193 ± 0.024

…but *why* does project do this?
We couldn't get the question out of our heads!

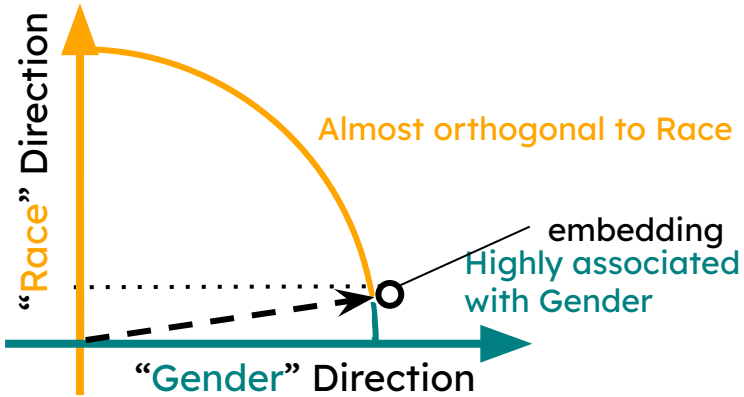
We Found The Answer Using A Little Linear Algebra



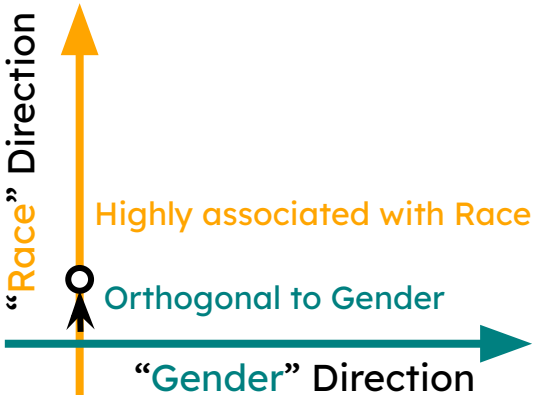
We Found The Answer Using A Little Linear Algebra



We Found The Answer Using A Little Linear Algebra



Removing Gender
Via
Projection Debiasing

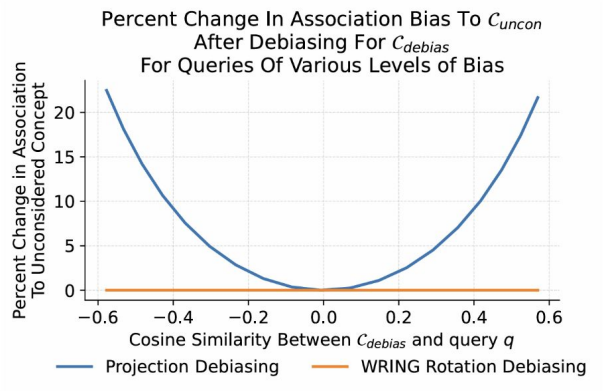


$$\underbrace{\text{bias}(\boldsymbol{v}_{\text{PROJECTION},C}, \boldsymbol{d_1}, \boldsymbol{d_2})}_{\text{bias after Projection}} = \underbrace{\frac{\|\boldsymbol{v}\|}{\|\boldsymbol{v} - \boldsymbol{P_C v}\|}}_{\text{bias amplification}} \cdot \text{bias}(\boldsymbol{v}, \boldsymbol{d_1}, \boldsymbol{d_2}) + \underbrace{\frac{\Delta_{\boldsymbol{P_C v}}}{\|\boldsymbol{v} - \boldsymbol{P_C v}\|}}_{\text{bias altering}},$$

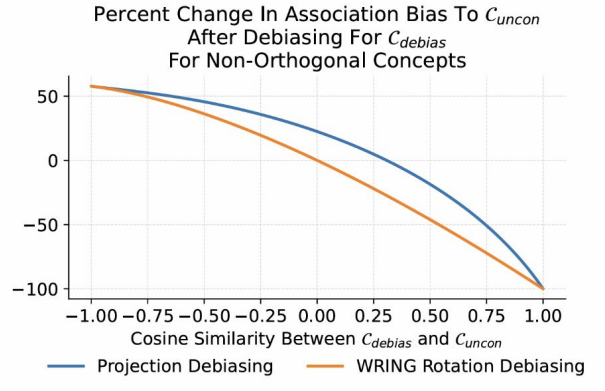
$$\Delta_{\boldsymbol{P_C v}} = \frac{\|d_1\| \langle \boldsymbol{P_C v}, \boldsymbol{d_2} \rangle - \|d_2\| \langle \boldsymbol{P_C v}, \boldsymbol{d_1} \rangle}{\|\boldsymbol{d_1}\| \|\boldsymbol{d_2}\|}.$$

We Discovered A Better Alternative Using Synthetic Data

- If projection causes bias amplification for off-target concepts, maybe we could replace it with another operation
- We started playing around with synthetic data and trying different operations
  - Using synthetic data allowed us complete control over our data assumptions
- We found that a rotation operation had nice properties:



(a) Comparing projection and rotation debiasing under the orthogonality assumption.



(b) Comparing projection and rotation debiasing when orthogonality is not satisfied.

We Discovered A Better Alternative Using Synthetic Data

- We knew that rotation had nice properties by observing its effects in synthetic data experiments We were then able to “reverse-engineer” formal statements about rotation based on these observations!

WRING Improves The Limitations of Projection: Let $\boldsymbol{v}_{\text{WRING},C}$ be the result of debiasing $\boldsymbol{v}$ with WRING. Let $\boldsymbol{d_1}, \boldsymbol{d_2} \in \text{col}(\boldsymbol{A}_D) \neq \text{col}(\boldsymbol{A}_C)$ be two embeddings in the subspace for concept $D \neq C$. The change in bias between $\boldsymbol{v}$ and $\boldsymbol{d_1}, \boldsymbol{d_2}$ is given by:

$$\underbrace{\text{bias}(\boldsymbol{v}_{\text{WRING},C}, \boldsymbol{d_1}, \boldsymbol{d_2})}_{\text{bias after WRING}} = \text{bias}(\boldsymbol{v}, \boldsymbol{d_1}, \boldsymbol{d_2}) + \frac{\|\boldsymbol{v} - \boldsymbol{P}_C\boldsymbol{v}\|}{\|\boldsymbol{v}\|} \underbrace{\frac{\Delta_{\boldsymbol{P}_C\boldsymbol{v}}}{\|\boldsymbol{v} - \boldsymbol{P}_C\boldsymbol{v}\|}}_{\text{bias altering}} - \underbrace{\Delta_{\boldsymbol{w}}}_{\text{dampening}} \tag{3}$$

where $\left(\Delta_{\boldsymbol{w}} = \langle \hat{\boldsymbol{w}}, \boldsymbol{d}_j \rangle - \langle \hat{\boldsymbol{w}}, \boldsymbol{d}_i \rangle\right) / \left(\|\boldsymbol{v}\| \, \|\boldsymbol{d_1}\| \, \|\boldsymbol{d_2}\| \right)$.

No bias amplification term. Unlike the change in bias after projection (Equation 1), the $\text{bias}(\boldsymbol{v}, \boldsymbol{d_i}, \boldsymbol{d_j})$ term is not multiplied by any bias amplification term.

Mitigated bias altering term. Equation 3 shares the *bias altering term* with Equation 1. However, in Equation 3 this term is mitigated by scaling it by a multiplicative factor that is less than 1. A nice property of WRING is that when $\Delta_{\boldsymbol{P}_C\boldsymbol{v}} > 0$, this term *immediately* helps to reduce bias without the need to overcome the amplification factor on the first term.

Dampening term. Equation 3 includes the bias dampening term $\Delta_{\boldsymbol{w}}$, which has a sign opposite to the bias amplification term. It acts as a mitigating factor and further helps to reducethe bias altering term.

No amplification when subspaces are orthogonal. When $\text{col}\, \boldsymbol{A}_D \perp \text{col}\, \boldsymbol{A}_C$ (the subspaces for C and D are orthogonal), the bias altering term and the dampening term are both 0. This means that WRING does not amplify the bias *at all* for orthogonal subspaces, unlike projection which always increases the bias for these spaces.

We Discovered A Better Alternative Using Synthetic Data

- Our theoretical findings were then supported by analysis on “real-world” datasets

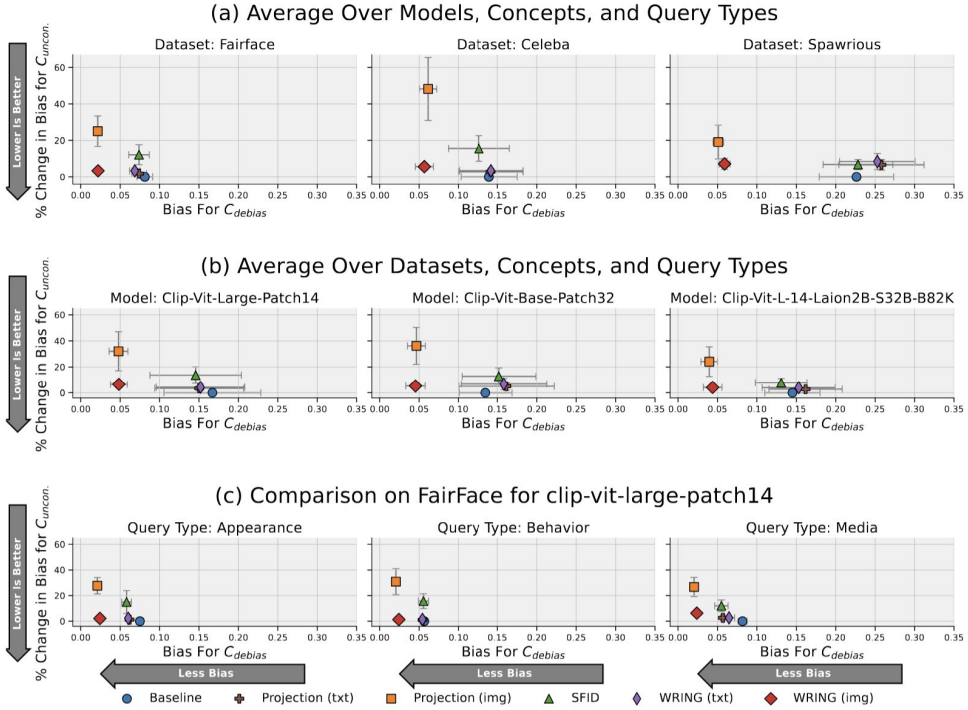


Figure 3: Comparison of debiasing approaches across models and datasets.

One takeaway from this methodology:

Understanding what’s happening within model representations (mathematically / geometrically) can be crucial for trustworthiness!

The Methodology Development Pipeline

Observe failure (Our method exhibited whac-a-mole dilemma)

↓

Identify mechanism (Projection causes bias amplification)

↓

Analyze assumptions (Projection doesn't maintain orthogonal relations)

↓

Build theory (Quantify how/why bias is amplified)

↓

Apply theory to build robust model (Rotation *does* preserve orth. relations)

Part 3: General Lessons for Building Methodologies

General Lessons for Building Methodologies

- **Study failures:** interesting and impactful work often comes from figuring out why something doesn't work!
  - New methods are needed when existing methods break!
  - e.g. temporal labeling patterns broke default labeling assumptions; the go-to debiasing operation of Projection leads to whac-a-mole

- **Search for mechanisms:** Always ask "*Why* did this happen?"
  - e.g. what causes temporal labeling; why does projection lead to whac-a-mole?

- **Don't be afraid to abstract the problem:** Figuring out a solution based on the underlying (general) mechanism can lead to better practical solutions
  - "People in my user study provide labels when they're not busy" -> Sequential labeling bias
  - The "whac-a-mole dilemma" being an empirical trend -> "projection inevitably causes it due to X, Y and Z geometric properties"

General Lessons for Building Methodologies

- **Keep one foot in each world:** Good methodologies often live at the intersection of application and abstraction!
    - If you're too focused on the specific application setting, you might not see underlying mechanisms
    - If you aren't grounded in your real task and real data, then you'll come up with a method that is (at best) intellectually interesting but likely won't be a good solution to the problems you should actually care about

General Lessons for Building Methodologies

- **Trustworthiness is often about assumptions:** Most failures happen because assumptions are violated
  - Methodology research often means making your assumptions *explicit* and modeling them *directly*

- **Synthetic data is a powerful tool:** Real-world data points you towards *what* is happening and *what* problems need to be solved. Synthetic data can show you *why* something is happening!
  - There are innumerable factors and degrees of freedom behind real-world data
    - It can be incredibly difficult to construct real counterfactual data, or to ablate data properties using real-world data
    - Synthetic data gives you full control and lets you test assumption one-by-one