Quantifying and Exploiting Latent Structure in Machine Learning: Concepts, Confounders, and Co-Hierarchies

by

Jitian Zhao

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 06/02/2025

The dissertation is approved by the following members of the Final Oral Committee:

Karl Rohe, Professor, Statistics
Frederic Sala, Assistant Professor, Computer Science
Kris Sankaran, Assistant Professor, Statistics
Keith Levin, Assistant Professor, Statistics
Yinqiu He, Assistant Professor, Statistics

© Copyright by Jitian Zhao 2025
All Rights Reserved

ACKNOWLEDGMENTS

This thesis has come a long way, and it would not have been possible without the support and encouragement of my advisors, collaborators, friends, and family.

First and foremost, I am deeply grateful to my advisors, Professor Karl Rohe and Professor Frederic Sala. I joined Karl's lab at the start of the pandemic, when everything felt uncertain. Karl's patience, openness, and flexibility gave me the freedom to freely explore my research interests and helped me grow from an inexperienced student into a confident researcher. I will always remember the afternoons we spent together, sketching and discussing those wild ideas. Fred is extraordinarily kind and knowledgeable. He offers practical guidance, works side by side with students during deadlines, and brings clear advice and optimism that carried me through moments of doubt. I am fortunate to have learned so much from both of them.

I am also grateful to the members of my thesis committee: Professor Kris Sankaran, Professor Keith Levin and Professor Yinqiu He, for their insightful feedback, invaluable time and discussions. I would like to extend my gratitude to the faculty members who guided me through classes and discussions, particularly Professor Zhengjun Zhang, Professor Yiqiao Zhong, and Professor Nicolas Garcia Trillos, from whom I am fortunate to learn fundamentals of statistical learning and mathematical statistics. A special thanks to Dr. Sebastian Raschka, who first introduced me to research and large language models, which later became the main focus of my work.

To my amazing labmates, thank you for your support, insight, and all the fun moments. From Rohe Lab: Sijia Fang, Fan Chen, Alex Hayes, Auden Krauska and Muzhe Zeng. From Sprocket Lab: Harit Vishwakarma, Nick Roberts, Tzu-Heng Huang, Changho Shin, Sonia Cromp, Dyah Adila,

Namburi GNVV Satya Sai Srinath, and everyone else in the lab. You made research not just meaningful, but also enjoyable.

Life in Madison was made so much better by the friendship of Sixu Li, Shenghong Dai, Yibing Wei, Junyi Wei, Xinran Miao, Jiaxin Hu, Shuyang Chen, Zhuoyan Xu, and Xufeng Cai. Thank you for all the laughs, chats, and shared experiences that made this journey special. To my childhood friends, Yuting Feng, Yating Zhang, and Jia Zhou. Thank you for always being there with hugs and support, even from afar. And to my furry friends, Tofu and Latte, thank you for giving me love and comfort through the hardest moments.

To my partner, Chenghui Li, thank you for being there every step of the way. Your encouragement and support have meant everything to me, and I feel so lucky to have shared this journey with you.

Finally, to my parents, Jihua Yu and Wenjian Zhao, thank you for your unconditional love and belief in me. I hope this milestone reflects the love and courage you gave me. And to my grandpa, Runqiao Yu, who taught me how to read and write when I was a kid, I wish you could be here to see this. I carry your memory with me.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The central theme of this work is about identification, quantification, and exploitation of latent structure inherent within complex, high-dimensional data. Across diverse machine learning scenarios, latent variables often encode meaningful patterns that are critical to both interpretability and model performance. This work explores such latent structures through three distinct yet conceptually interconnected projects.

Part I investigates latent conceptual structures within embeddings produced by deep neural networks, specifically Contrastive Language–Image Pre-training (CLIP) embeddings (Radford et al., 2021). By developing a rigorous statistical framework, we quantify rotation-sensitive structures to ensure that identified concepts represent robust, interpretable patterns rather than artifacts specific to certain methodologies. This approach enhances interpretability and helps mitigate reliance on spurious data correlations, as empirically demonstrated through improved worst-group accuracy on challenging benchmarks.

Part II addresses latent structures manifesting as biases and confounding factors in model evaluation frameworks, particularly when leveraging multiple Large Language Model (LLM) judges. Here, we employ probabilistic graphical models, explicitly capturing and disentangling latent correlations and confounders among judges. Our novel decomposition approach combines sparse-plus-low-rank matrix decomposition with tensor methods, resulting in a principled, statistically grounded aggregation methodology. This framework significantly reduces evaluation biases and aggregation errors, thereby yielding more reliable and interpretable evaluations.

Finally, part III explores latent hierarchical structures in bipartite graphs, aiming to uncover systematic co-clustering patterns that relate entities from two distinct domains. Instead of treating entities separately, we pro-

pose a unified hierarchical approach termed a *co-hierarchy*, which encapsulates latent structural dependencies via spectral decomposition techniques. This co-hierarchical framework reveals intricate relationships across hierarchical levels, enhancing interpretability and predictive understanding in applications such as recommendation systems and document classification.

Collectively, these projects illustrate a unified goal: extracting meaningful latent structures from data without explicit label supervision. While each work employs distinct statistical methods—ranging from hypothesis testing and spectral decomposition to graphical modeling—all emphasize rigorous theoretical guarantees and practical effectiveness in leveraging latent structures.

While these connections underscore a cohesive shared theme, each part of the dissertation corresponds to a separate manuscript, designed to be self-contained. Each manuscript introduces unique notation, provides independent theoretical results, and can be read independently.

Part I

Quantifying Structure in CLIP Embeddings: A Statistical Framework for Concept Interpretation

ABSTRACT

Concept-based approaches, which aim to identify human-understandable concepts within a model's internal representations, are a promising way to interpret embeddings from deep neural network models like CLIP. While these approaches help explain model behavior, current methods lack statistical rigor, making it hard to validate identified concepts and compare different techniques. To address this challenge, we introduce a hypothesis testing framework that quantifies rotation-sensitive structures within the CLIP embedding space. Once such structures are identified, we propose a post-hoc concept decomposition method. Unlike existing approaches, it offers theoretical guarantees that discovered concepts represent robust, reproducible patterns (rather than method-specific artifacts) and outperforms other techniques in terms of reconstruction error. Empirically, we show that our concept-based decomposition algorithm effectively balances reconstruction accuracy with concept interpretability and helps mitigate spurious cues in data. Applied to a popular spurious correlation dataset, our method yields a 22.6% increase in worst-group accuracy after removing spurious background concepts.

Note to Reader

This manuscript is collaborated with co-authors Chenghui Li, Frederic Sala, and Karl Rohe. It is currently submitted and under review. The notation used here is self-contained and can be read independently of other parts of the paper.

1 INTRODUCTION

CLIP Radford et al. (2021) is a powerful tool useful for a wide range of visual applications. Interpreting its high-dimensional embeddings is challenging due to the complex and entangled nature of the learned representations. Recent works address this via *concept-based decomposition*, aiming to identify interpretable semantic patterns within model components, embeddings, and neurons (Gandelsman et al., 2024a,b; Balasubramanian et al., 2024).

Existing approaches broadly fall into two categories. The first category, exemplified by SpLiCE (Bhalla et al., 2024), decomposes CLIP embeddings into sparse, human-interpretable concepts such as words. While this sparse decomposition improves interpretability, it introduces reconstruction errors, meaning that a substantial portion of the original embedding's information is lost. This negatively impacts downstream tasks such as zero-shot classification, where preserving semantic information is crucial. The second category uses Singular Value Decomposition (SVD) to decompose embeddings into linear combinations of concept vectors (Fel et al., 2024; Graziani et al., 2023; Zhang et al., 2021). These methods maintain high reconstruction fidelity (i.e., the reconstructed embedding closely approximates the original with minimal error by filtering out noise). However, they often struggle with concept interpretability, as singular vector directions are not inherently aligned with human-interpretable concepts, making their meaning largely dependent on human intuition. This highlights a trade-off: methods that prioritize interpretability often sacrifice reconstruction fidelity, while those that preserve fidelity tend to lack meaningful concept alignment.

Beyond this seeming interpretability-reconstruction fidelity trade-off, ***an even deeper issue remains***: existing methods are ad-hoc rather than the result of a rigorous statistical framework. In real-world settings, em-

beddings are inherently noisy, and without statistical guarantees, it is unclear whether the concepts extracted by a given method capture meaningful structure or merely reflect artifacts of noise. A key challenge is that noise and meaningful structure can both produce seemingly interpretable components, leading to silent failures where methods extract spurious "concepts" that do not correspond to real semantic attributes.

To address both challenges, we first propose a hypothesis testing framework to detect rotation-sensitive structure in the embedding subspace. Our key insight is that semantic concepts manifest as directional patterns in embedding space that are *sensitive to rotation, unlike random noise* which remains statistically unchanged under rotation. By testing for rotation-sensitive patterns, our method distinguishes noise from true underlying structure, ensuring that extracted concepts reflect meaningful, stable properties of the data rather than arbitrary artifacts.

Building on the theoretical insights of Varimax rotation (Rohe and Zeng, 2020), we develop a post-hoc method that requires no additional training or human annotation. Our approach ***achieves both the interpretability benefits of sparse decomposition and the high reconstruction fidelity of SVD-based methods while offering statistical guarantees of concept recoverability***.

The remainder of this paper is organized as follows:

- In Section 2, we present a hypothesis testing framework to quantify rotation-sensitive concept structure in the embedding space. We provide the detailed test procedure, theoretical guarantees for test statistics, as well as empirical results.

- In Section 3, we introduce a post-hoc concept-based decomposition method accompanied with an automatic concept interpretation algorithm.

- In Section 4, we formalize a statistical model that connects concepts

with embeddings and prove concept identifiability under certain assumptions. We show that concept decomposition methods with a fixed, misspecified concept vocabulary can suffer from reconstruction errors.

- In Section 5, we show through qualitative analysis that our method learns interpretable concepts and maintains high reconstruction fidelity, as evidenced by a sparsity-performance trade-off analysis. We also show that our method is effective in identifying and removing spurious concepts.

2 HYPOTHESIS TEST FOR ROTATION SENSITIVE CONCEPTS

We develop a hypothesis testing framework to detect concepts in embedding spaces through rotational properties. We first characterize meaningful concepts through rotational sensitivity (Sec. 2.1), validating our intuition on synthetic and real datasets. We then develop a resampling procedure (Sec. 2.2), test statistics (Sec. 2.3), and test procedure swith experimental results in (Sec. 2.4).

2.1 Characterizing Meaningful Structure via Rotational Sensitivity

We first explain why a rotation-based approach is well-suited for statistically modeling concepts in embeddings. Neural networks process embeddings through inner products with weight vectors, which measure how closely the embedding aligns with each weight vector's direction, making embeddings inherently directional objects. When examining embedding spaces for meaningful structure, we are essentially asking whether the distribution of embeddings shows preferences for certain directions over others. A completely structureless embedding space samples points uniformly from all directions, making it *rotationally invariant*. In contrast, meaningful concepts manifest as preferred directions in the embedding distribution, breaking this invariance.

To accurately characterize these rotational preferences, we focus on the rotational properties of singular vectors rather than raw embeddings. This choice is crucial because heterogeneous scaling in different directions (i.e., elliptical patterns in data) can create apparent rotation sensitivity in the raw embeddings even when no meaningful structure exists. Singular vectors, being normalized to unit variance, allow us to identify true directional preferences while controlling for such scaling effects.

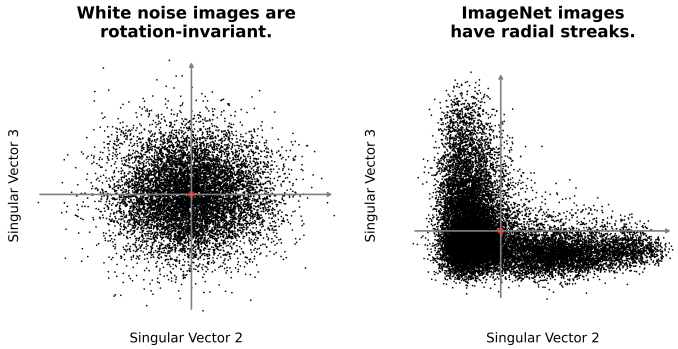


Figure 2.1: Visualization of singular vector loadings from CLIP embeddings from ViT-B/32 backbone model, where loadings represent how much each singular vector contributes to an image's representation. Left: Projection onto 2nd and 3rd singular vectors (after Varimax rotation) of embeddings from white noise images, showing rotation-invariant structure. Right: Same projection for ImageNet validation images, revealing distinct radial streak patterns that indicate rotation-sensitive structure. Each point represents one image, with first singular vector excluded to remove mean effects.

We now formalize these ideas, starting with a formal definition of rotational invariance:

Definition 2.1.1 (Probabilistic Rotational Invariance)**.** A probability distribution with density function f on $\mathbb{R}^d$ is said to be *rotationally invariant* if for any rotation matrix $R \in \mathbb{R}^{d\times d}$ (i.e., $R^\top R = I_d$ and $\det(R) = 1$), the distribution of x is the same as the distribution of xR, $f(x) = f(xR)$ for all $x \in \mathbb{R}^d$ and all rotation matrices R.

Intuitively, this definition formalizes when a distribution looks the same in all directions; there are no preferred directions or patterns in the data. To illustrate this definition, we first examine classical examples of rotation invariance, both at the data distribution level and the concept level:

Example 1 (Standard Multivariate Normal is Rotationally Invariant). The multivariate Gaussian distribution $\mathcal{N}(0, I_d)$ is rotationally invariant.

While Example 1 demonstrates rotation invariance at the data level, we are particularly interested in the rotational properties of concept-specific patterns, which we capture through singular vectors:

Example 2 (Singular Vectors of Gaussian Noise are Rotationally Invariant). Let $A \in \mathbb{R}^{n \times d}$ be a matrix with entries A_{ij} drawn i.i.d. from $\mathcal{N}(0, 1)$. Then the left singular vector matrix $U \in \mathbb{R}^{n \times k}$ and right singular vector matrix $V \in \mathbb{R}^{d \times k}$ are both rotationally invariant.

This serves as our null model, representing the absence of meaningful concept structure. In contrast, embeddings that encode meaningful concepts exhibit rotation sensitivity. For example,

Example 3 (Gaussian Mixture Model Shows Rotation Sensitivity). Consider data drawn from a Gaussian mixture model: $x \sim \frac{1}{2}\mathcal{N}(\mu, I_d) + \frac{1}{2}\mathcal{N}(-\mu, I_d)$, where $\mu = (1, 0, 0, \ldots, 0)^\top \in \mathbb{R}^d$. This distribution is not rotationally invariant.

We show its concept structure is also rotation-sensitive and defer the details in Example 4.

We can further validate these examples using real CLIP embeddings in Figure 2.1. The left panel shows embeddings of white noise images, displaying uniform distribution from all directions (i.e., rotation invariance) similar to Ex. 2. The right panel shows ImageNet embeddings, revealing clear directional structure through radial streaks, analogous to the structured distribution in Ex. 3.

To formalize these ideas, let $A \in \mathbb{R}^{n \times d}$ be the data matrix. We obtain its truncated singular value decomposition, where $U \in \mathbb{R}^{n \times k}$ contains the first k left singular vectors of A. Each column of U represents a principal direction of variation in the data. Our hypothesis test specifically examines the rotational properties of left singular vectors (matrix U).

2.2 Sampling from the Rotation invariant Distribution

Rotation sensitivity only manifests when an embedding prefers certain directions. To mimic the absence of such structure, we generate a null sample by *independently* rotating each row of the embedding while preserving its length. We describe the details of this method in Algorithm 8, which is deferred to Appendix. Comparing any test statistic to the distribution obtained from these rotated replicas yields a Monte-Carlo p-value for the presence of rotation-sensitive patterns.

The following proposition establishes the theoretical guarantees for our resampling procedure:

Proposition 2.2.1 (Statistical Properties of Resampling)**.** *Let* $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$ *be i.i.d. samples from a probability distribution with density* f*. For any measurable test statistic* $T : \mathbb{R}^{d\times n} \to \mathbb{R}$*, define:*

$$T_1 = T(x_1, \ldots, x_n), \quad x_i^{rot} = R^i x_i, \quad R^i \overset{iid}{\sim} \mathit{Uniform}(SO_d), \quad T^* = T(x_1^{rot}, \ldots, x_n^{rot}).$$

If f *is rotationally invariant, then* T^* *and* T_1 *have the same distribution and are conditionally independent given the set of norms* $\{\|x_i\|_2\}_{i=1}^n$.

This proposition guarantees that under the null hypothesis of rotational invariance, resampled test statistic (T^*) follows the same distribution as the original statistic (T_1).

2.3 Test Statistics

To detect and quantify rotation-sensitive structure in embedding spaces, we propose two complementary test statistics that capture different aspects of rotation sensitivity: distributional non-Gaussianity through kurtosis and achievable sparsity under rotation through Varimax objective function.

Kurtosis-based Statistic. Our first test statistic measures the non-Gaussian patterns in U, as meaningful concepts typically deviate from normal distributions. We define:

$$TS_1(U) = \frac{1}{k}\sum_{i=1}^{k} |\text{kurtosis}(U_{\cdot i})|, \quad \text{where kurtosis}(X) = \frac{\mathbb{E}[(X-\mu)^4]}{(\mathbb{E}[(X-\mu)^2])^2} - 3,$$

with $\mu = \mathbb{E}[X]$. Under the rotation invariance null hypothesis, we expect this statistic to be close to zero.

Theorem 2.3.1. *Under the null model of Example 2, an equivalent rescaled version of* $TS_1(U)$ *follows a standard normal distribution.*

This normalization provides an efficient computational path for hypothesis testing under Gaussian assumptions. We defer the detailed proof to Appendix.

Varimax-based Statistic. Our second test statistic optimizes over rotations to detect patterns that may be hidden in the original coordinate system. We define:

$$\nu(U,R) = \sum_{\ell=1}^{k} \frac{1}{n}\sum_{i=1}^{n} \left(|UR_{i\ell}|^4 - \left(\frac{1}{n}\sum_{q=1}^{n} |UR_{q\ell}|^2\right)^2\right), \qquad TS_2(U) = \max_{R\in SO_k} \nu(U,R), \tag{2.1}$$

where ν is the Varimax objective function, and SO_k is the special orthogonal group of $k \times k$ rotation matrices. This statistic measures the maximum achievable sparsity under rotation, making it particularly sensitive to structured patterns that may be hidden in the original coordinate system.

To apply these test statistics, we use a bootstrap approach. We first generate samples from the null distribution by applying random rotations to the original data matrix. We then compute both test statistics on these null samples to form their empirical distributions. The p-values are calculated by comparing the observed test statistics against these null distributions.

Algorithm 1 Hypothesis Test for Rotation-Sensitive Concepts

Input: Matrix $U \in \mathbb{R}^{n \times k}$, resamples N_{resample}
Output: Varimax p-value p_v, kurtosis p-value p_{kur}

for $i = 1$ to N_{resample} **do**
  $U_i^{\text{rot}} \leftarrow$ Rotation Invariant Matrix(U) ▷ Algorithm 8
  $Z_i^{\text{rot}} \leftarrow U_i^{\text{rot}} \times \arg\max_{R \in SO_k} v(U_i^{\text{rot}}, R)$ ▷ Apply Varimax rotation in eq: equation 2.1
  Compute statistics $TS_1(Z_i^{\text{rot}})$, $TS_2(Z_i^{\text{rot}})$
end for
$\hat{Z} \leftarrow U \times \arg\max_{R \in SO_k} v(U, R)$
Compute $TS_1(\hat{Z})$, $TS_2(\hat{Z})$
Compute p-values:
$p_{\text{kur}} \leftarrow \frac{\sum \mathbb{1}[TS_1(\hat{Z}) > TS_1(Z_i^{\text{rot}})]}{N_{\text{resample}}}$
$p_v \leftarrow \frac{\sum \mathbb{1}[TS_2(\hat{Z}) > TS_2(Z_i^{\text{rot}})]}{N_{\text{resample}}}$
return p_{kur}, p_v

2.4 Test Procedure and Results

We present a hypothesis testing procedure to detect rotation-sensitive structure in embedding spaces. Under the null hypothesis of rotation invariance, we expect large p-values indicating no meaningful structure (e.g., $\geqslant 0.05$) , while significantly small p-values (e.g., $\leqslant 0.05$) suggest the presence of rotation-sensitive structure. Algorithm 1 outlines our testing approach.

We evaluate our method using CLIP ViT-L/14 embeddings on three datasets: ImageNet validation set images, white noise images (Gaussian noise fed into the CLIP model), and pure white noise embeddings. As shown in Figure A.2, our test statistics reveal strong evidence of rotation-sensitive structure in CLIP embeddings of ImageNet images, with both test statistics showing significant separation between their bootstrap null distributions and observed values. Control experiments on white noise *confirm our method's validity*. Both white noise embeddings (p-values: 0.55

for Kurtosis, 0.6 for Varimax) and white noise images (p-values: 0.62 for Kurtosis, 0.81 for Varimax) yield non-significant p-values, confirming that random noise contains no meaningful structure.

3 IDENTIFYING INTERPRETABLE CONCEPTS

We present our main algorithm to identify rotation-sensitive concept structure in the embedding space, as motivated by the notions in Section 2. Our approach decomposes the embedding matrix into two components: a sparse loading matrix that captures how individual images relate to concepts, and an orthogonal concept dictionary that maintains independence between discovered concepts. This achieves both interpretability and high reconstruction fidelity.

3.1 Method Overview

As shown in Figure 3.1, our pipeline processes image and text inputs through a pretrained CLIP model to obtain embeddings. Given image embeddings $A \in \mathbb{R}^{n \times d}$ and text embeddings $T \in \mathbb{R}^{M \times d}$, we learn k orthogonal concepts represented through three key matrices: concept dictionary matrix $\hat{Y}$, image loadings $\hat{Z}_{img}$, and text loadings $\hat{Z}_{txt}$. The image and text loadings indicate how strongly each image or text embedding aligns with the learned concept - higher values represent a stronger association with a particular concept in the dictionary.

Algorithm 2 details our decomposition method, which has three key steps. First, we normalize the embedding matrix for better spectral estimation (see details in Appendix A.3). Second, we perform SVD to identify the principal directions in the embedding space. However, these raw SVD components, while capturing the underlying structure, are not automatically aligned with human-interpretable concepts. This motivates our third step: Varimax rotation, which maximizes the variance of squared loadings for each concept, naturally pushing individual loadings toward either high values or zero and thus promoting sparsity. This sparsification is crucial for interpretability—it associates each data point primarily with its most

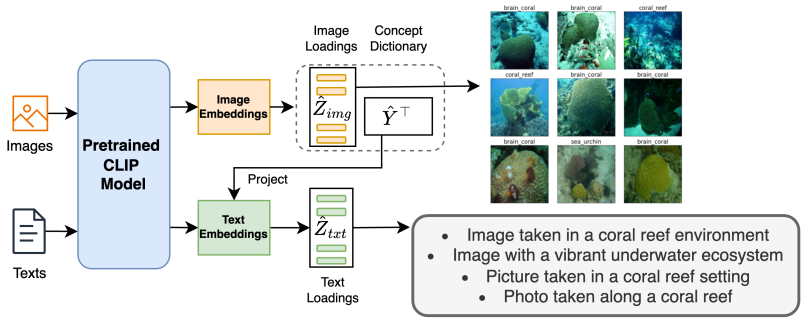


Figure 3.1: Pipeline overview. CLIP embeddings from images and texts are processed to extract interpretable concepts. Image embeddings are factorized into sparse loadings ($\hat{Z}_{img}$) and a concept dictionary ($\hat{Y}$), while text embeddings are projected to obtain loadings ($\hat{Z}_{txt}$). Bottom: Example of a discovered concept with its representative images and text descriptions.

Algorithm 2 Concept-based Embedding Decomposition

Input: Embedding matrix $A \in \mathbb{R}^{n \times d}$, number of concepts k
Output: Concept matrix $\hat{Y} \in \mathbb{R}^{d \times k}$, image loadings $\hat{Z} \in \mathbb{R}^{n \times k}$
$\tilde{A} \leftarrow \text{Normalization}(A)$
$U, D, V^\top \leftarrow \text{SVD}(\tilde{A})$
$R \leftarrow \arg\max_{R \in SO_k} \nu(UD, R)$ ▷ Optimizes objective in Eq. equation 2.1
$\hat{Z} \leftarrow UDR$
$\hat{Y} \leftarrow VR$
return $\hat{Z}, \hat{Y}$

relevant concepts and makes concepts more semantically distinct by connecting them only to related examples. The rotation step effectively aligns the rotation-sensitive structure we detected (Section 2) with interpretable axes in the embedding space. We empirically validate the necessity of this rotation in Section 5, where we show that rotated sparse concepts exhibit clearer semantics compared to raw SVD components.

Algorithm 3 Automatic Concept Interpretation

Input: Concept matrix $\hat{Y} \in \mathbb{R}^{d \times k}$, text descriptions $\{M_i\}_{i=1}^{M}$, text embeddings $T \in \mathbb{R}^{M \times d}$, descriptions per concept r
Output: Text descriptions $\{\mathcal{T}_j\}_{j=1}^{k}$ for each concept
$L \leftarrow T\hat{Y}$ ▷ Project text embeddings to concept space
for each concept $j = 1, \ldots, k$ **do**
    $\mathcal{I}_j \leftarrow$ indices of top r values in $L_{\cdot j}$
    $\mathcal{T}_j \leftarrow \{M_i : i \in \mathcal{I}_j\}$
end for
return $\{\mathcal{T}_j\}_{j=1}^{k}$

3.2 Concept Interpretation

We propose two methods to interpret each concept from the decomposition. The first uses the image loading matrix $\hat{Z}$ to identify representative examples for concepts. For the j-th concept, we examine its corresponding column $\hat{Z}_{\cdot j}$ and select the r images with highest loading scores. These typically share common semantic features, allowing us to derive an interpretable theme for the concept.

Our second method (Alg. 3), provides automatic concept interpretation through text descriptions without human intervention. This approach requires a pool of text descriptions for the image dataset, which can be obtained through LLMs or visual-language models (e.g., LLaVA (Liu et al., 2023)). The algorithm projects these text descriptions onto our learned concept space and identifies the most relevant descriptions for each concept. In our experiments, we use the curated text description set from Gandelsman et al. (2024b), which provides general descriptions of ImageNet classes.

4 THEORETICAL RESULTS: IDENTIFICATION AND RECOVERY BOUNDS

In this section, we establish theoretical guarantees for our concept decomposition method. We first prove that our method can reliably identify meaningful concepts under certain statistical assumptions, extending previous work on Varimax rotation identification. We then analyze fundamental limitations of fixed-concept approaches, demonstrating why adaptive concept learning is necessary.

4.1 Concept Identification

Our identification guarantees build on a key insight: when embedding data exhibits sufficient statistical structure, there exists a unique rotation (up to permutation) that aligns with interpretable concepts. We formalize this through the following assumptions:

Assumption 4.1.1 (The identification assumptions for Varimax)**.** The matrix $Z \in \mathbb{R}^{n \times k}$ satisfies the identification assumptions for Varimax if all of the following conditions hold on the rows $Z_i \in \mathbb{R}^k$ for $i = 1, \ldots, n$: (i) the vectors $Z_1, Z_2, \ldots, Z_n$ are i.i.d., (ii) each vector Z_i has k independent random variables (not necessarily identically distributed), (iii) the elements of Z_i have kurtosis $\kappa > 3$.

The independence conditions (i) and (ii) ensure structural consistency across samples, while (iii) requires sufficient non-Gaussianity in the data. We relax the equal variance assumption from Rohe and Zeng (2020), allowing different concepts to have different strengths of expression. This is crucial. In the vintage sparse-PCA model the data admit the factorization $X = ZBY^\top$, where Z, Y are Varimax-rotated eigenvectors, and B is the diagonal matrix of eigenvalue, left and right multiplied by rotation

matrices. Because B absorbs all scaling, Z and Y can be rescaled without losing orthogonality. Our model instead factors the data as $X = ZY'$ for clear interpretation purpose, where Z is data loading on each concept, and Y is the concept dictionary. Hence, any attempt to transfer scale from Z to Y would break the orthogonality of Y, which makes concept dictionary harder to interpret.

Theorem 4.1.2 (Varimax rotation identification)**.** *Suppose that* $Z \in \mathbb{R}^{n\times k}$ *satisfies Assumption 4.1.1. Define* $\tilde{Z} = Z - \mathbb{E}(Z)$*. For any rotation matrix* $\tilde{R} \in \mathcal{O}(k)$*,*

$$\arg\max_{R\in\mathcal{O}(k)} \mathbb{E}\left(\nu(R, Z\tilde{R}^{\top})\right) = \{\tilde{R}P : P \in \mathcal{P}(k)\},$$

where $\mathcal{P}(k) = \{P \in \mathcal{O}(k) : P_{ij} \in \{-1, 0, 1\}\}$*, is the full set of matrices that allow for column reordering and sign changes, and* ν *is defined in equation 2.1.*

Under our assumptions, this shows the Varimax objective identifies the correct concept rotation up to permutation.

4.2 Reconstruction Error Bounds for Fixed-Concept Methods

When the concept matrix is fixed, reconstruction errors arise from potential misalignment between predefined concepts and the ground-truth concept structure. To formalize this limitation, we denote the ground-truth latent concept matrix as $\boldsymbol{C}^* \in \mathbb{R}^{d\times k}$ and the fixed concept matrix (such as in SpLiCE) as $\boldsymbol{C}_W \in \mathbb{R}^{d\times m}$ where $k \leqslant d < m$. We assume $\boldsymbol{C}_W$ may fail to capture some information present in $\boldsymbol{C}^*$, which we quantify through the following condition: $\min_{P\in\mathbb{R}^{m\times k}} \|\boldsymbol{C}_W P - \boldsymbol{C}^*\|_F \geqslant \delta$, where P is an arbitrary projection matrix, $\delta > 0$ represents the minimum possible misalignment between the fixed and true concepts, and $\|\cdot\|_F$ represents the Frobenius norm.

Theorem 4.2.1 (Fixed concept-decomposition method reconstruction error lower bound)**.** *Given the misspecification condition above, consider* $A \in \mathbb{R}^{n \times d}$ *such that* $A = Z^* \boldsymbol{C}^{*\top}$ *with positive* k*-th singular value, i.e.* $\sigma_k(Z^*) > 0$*, then we have* $\min_{Z \in \mathbb{R}^{n \times m}} \|A - Z\boldsymbol{C}_W^\top\|_F \geqslant \sigma_k(Z^*)\delta$*, where* $\sigma_k(Z) = \sqrt{\sigma_k(Z^\top Z)}$ *is the absolute* k*-th largest singular value of* Z*.*

This theorem quantifies the risk of fixing the concept decomposition matrix in SpLiCE: when the predefined concept vocabulary cannot be aligned with the true concepts, reconstruction error is unavoidable. Our proposed method avoids this limitation by learning concepts from the data.

5 EXPERIMENT RESULTS

We evaluate our method through qualitative and quantitative analyses. We assess the interpretability of learned concepts via visualizations and textual alignment, analyze the trade-off between sparsity and reconstruction fidelity, and demonstrate the effectiveness of our method in removing spurious correlations across multiple datasets.

5.1 Qualitative Evaluation of Discovered Concepts

We evaluate the effectiveness of our concept decomposition method visually and textually .

Setup. We apply our method to CLIP ViT-B/32 embeddings of the ImageNet validation set, using the curated text description set from Gandelsman et al. (2024a) that provides class-specific descriptions generated via ChatGPT.

Results. Figure 5.1 compares concept clusters discovered by our Varimax-rotated decomposition (left column) against those from raw SVD (right column). For two representative concepts, we display the top nine images by loading score ($\hat{Z}_{\cdot j}$) alongside their automatically retrieved text descriptions from Algorithm 3. The top row shows a concept manually selected to demonstrate the effectiveness of our method, while the bottom row concept displays a randomly selected concept from a pool of 50. ***Our method consistently yields semantically coherent clusters*** (***e.g. butterfly feeding scenes, screws***) ***with concise, focused descriptions.*** In contrast, raw SVD clusters mixed themes such as furniture and animals, screws and knitwears, accompanied with broader, less specific descriptions. These differences demonstrate that Varimax rotation effectively isolates mean-

ingful concept directions in the CLIP embedding space, resulting in far more structured and interpretable concept representations than standard SVD decomposition.

5.2 Sparsity-performance Trade-off

We analyze how the number of concepts in our decomposition affects reconstruction fidelity, measured by cosine similarity between original and reconstructed embeddings.

Setup. We evaluate our method on the ImageNet validation set using two CLIP models: ViT-B/32 (512 dimensions) and ViT-L/14 (768 dimensions). We compare against SpLiCE (Bhalla et al., 2024) as a baseline.

Results. Figure 5.2 shows that reconstruction fidelity improves with increasing number of concepts k for both models. ViT-L/14 consistently shows lower cosine similarity compared to ViT-B/32 at equal k, reflecting the challenge of capturing its richer 768-dimensional embedding space with the same concept budget. Our method ***achieves substantially higher reconstruction fidelity*** compared to SpLiCE when using comparable numbers of concepts.

The quality of reconstruction is crucial as it indicates how well our decomposition preserves the semantic information encoded in the original embeddings. While concept decomposition inherently involves a trade-off between interpretability and information preservation, our method offers flexible control through the number of concepts k, allowing users to balance these competing objectives. Unlike SpLiCE, which prioritizes interpretability at the cost of significant information loss, ***our approach maintains interpretable concepts while better preserving the original embedding structure***. This preservation of semantic information is essential for downstream applications and validates that our discovered concepts capture meaningful aspects of the data.

Figure 5.1: Comparison of concept clusters obtained by our Varimax-rotated decomposition (left column) and raw SVD (right column) on CLIP image embeddings. Each cell shows the top nine images for a given concept, annotated with their retrieved text descriptions. Our method produces tight, semantically coherent clusters and precise labels, whereas raw SVD yields mixed-semantics groups and more generic descriptions.

Model	Waterbirds			iWildCam			CelebA		
	Avg	WG†	Gap‡	Acc	mF1	MRec	Avg	WG†	Gap‡
ZS	84.8	38.1	46.7	6.23	0.001	0.002	81.2	74.2	7.0
SVD-recon.	85.5	39.0	46.5	3.83	0.001	0.001	78.1	74.9	**3.2**
Spurious Removed	**89.6**	**60.7**	**28.9**	**18.8**	**0.003**	**0.006**	**82.6**	**75.1**	7.5

Table 5.1: Performance comparison across datasets. WG†: worst-group accuracy (higher better), Gap‡: accuracy gap (lower better), mF1: micro-F1, MRec: macro-recall. Best results in **bold**. Purple and green highlights indicate best worst-group accuracy and smallest accuracy gap.

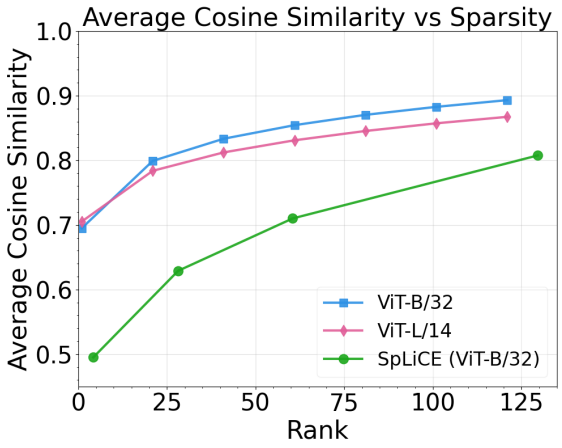


Figure 5.2: Reconstruction quality versus number of concepts. Higher cosine similarity indicates better preservation of the original embedding structure. Our method offers better interpretability-fidelity trade-off than SpLiCE.

5.3 Removing Spurious Correlations

We evaluate our method's ability to identify and remove spurious correlations across three datasets: Waterbirds (Sagawa et al., 2019), WILDS-iWildCam (Beery et al., 2020), and CelebA (Liu et al., 2015). Dataset details are provided in Appendix A.3.

Setup. For all experiments, we use CLIP ViT-B/32 embeddings (512 dimensions) and decompose them into $k = 50$ concepts using Algorithm 2. Spurious concepts are identified using dataset-specific strategies (detailed

in Appendix A.3), which analyze correlation between concept embeddings and text descriptions emphasizing different attributes (e.g., target vs. background features). We defer the details of how to remove spurious concepts to appendix.

We compare three embeddings: the full, original CLIP image embedding; the concept-based reconstruction, where embeddings are reconstructed from all learned concepts; and the spurious-removed reconstruction (Ours), where embeddings are reconstructed after removing spurious concepts. For classification, we follow the standard zero-shot classification setup.

Results. Our method ***consistently improves zero-shot prediction performance after removing spurious concepts*** (Table 5.1). On Waterbirds, removing spurious background concepts improves worst-group accuracy by 22.6% and reduces the accuracy gap by 17.8%. On iWildCam, the prediction accuracy triples from 6.23% to 18.8%; and on CelebA, we achieve highest average and worst-group accuracy while using only 5% of the original embedding dimensions. The SVD-reconstructed embeddings maintain similar average accuracy to the original embeddings for Waterbirds dataset, suggesting our method preserves task-relevant information while removing noise.

6 CONCLUSION

We introduced a hypothesis testing framework to quantify rotation-sensitive structures in the embedding spaces and proposed a concept-decomposition method that achieves both high reconstruction fidelity and clear interpretability. We validated it through theoretical and empirical analyses. Applied to challenging distribution shift benchmarks, our method consistently demonstrated significant improvements after identifying and removing spurious concepts.

Limitations. Our approach assumes linearity in the embedding decomposition, which may overlook complex non-linear structures potentially present in the embedding space. In addition, we note that the interpretability of discovered concepts partially depends on the quality and scope of the text descriptions available, potentially introducing biases or limiting generalization. Finally, while our hypothesis testing procedure is robust to rotationally invariant noise, it does not explicitly handle structured, non-rotational noise patterns, leaving room for further refinement in more nuanced settings.

Part II

From Many Voices to One: Statistically Principled Aggregation of LLM Judges

ABSTRACT

LLM-as-a-judge—often with multiple judges—is now the standard paradigm for scalable model evaluation. This strategy is known to suffer from biases, spurious correlations, confounding factors, etc., and many heuristic approaches have been proposed to tackle these. We address this problem from the point of view of probabilistic graphical models, enabling us to ***capture the challenges involved in using multiple judges in a principled way***. By considering Markov random fields (MRF) with multiple latent factors, we can model undesired correlations between judges, a latent unknown true notion of quality, and one or more additional latent distractors (for example, generation length). The key technical challenge is to identify and learn a higher-rank latent variable MRF, which we solve via a new approach that mixes sparse plus low-rank and tensor decompositions. This enables us to better understand the quality and behavior of judges, leading to improved evaluation capabilities. In addition, we show how to augment our approach via programmatic judges that can be cheaply constructed and added to standard model-based judges. Empirically, our framework, CARE (Confounder-Aware Aggregation for Reliable Evaluation), demonstrates consistent gains on diverse public benchmarks, reducing aggregation error by up to 25.15% and showing robust integration of programmatic judges. Additionally, CARE offers superior performance and efficiency compared to individual-judge intervention strategies. These results underscore CARE's ability to effectively model correlations and mitigate biases, leading to more accurate and robust aggregation of LLM judge scores.

Note to Reader

This manuscript is collaborated with co-authors Changho Shin, Tzu-Heng Huang, Srinath Namburi and Frederic Sala, and is currently submitted and under review. The notation used here is self-contained and can be read independently of other parts of the paper.

7 INTRODUCTION

Large language models (LLMs) are the workhorse solution for automated evaluation of model generations. For example, using *LLM-as-a-judge* systems avoids incurring the cost and latency of expert annotation (Zheng et al., 2023). Given the ease of applying such tools, a common evaluation paradigm is to *ensemble multiple LLM judges* to form consensus evaluation scores (Hu et al., 2024). While attractive, these approaches are unreliable. Judges can be individually inaccurate and suffer from biases, e.g., relying on spurious factors like position or verbosity (Ye et al., 2025; Shi et al., 2024; Wang et al.). Additionally, judge models are highly correlated (due to being trained on the same data), so that incorporating more judges may add no additional evaluation signal—or worse, boost confidence in an incorrect assessment (Deutsch et al., 2022; Li et al., 2025).

Many heuristic techniques have been proposed to mitigate these issues. Single judge bias-reduction methods include answer-order shuffling (Chen et al., 2024), prompt calibration (Li et al., 2024a; Furniturewala et al., 2024; Guo et al., 2022), and fine-tuned evaluators (e.g., JudgeLM (Zhu et al., 2023), PandaLM (Wang et al.)). Ensembling methods aggregate judge scores via a simple majority vote or average (Li et al., 2024b) in the hope of reducing unreliability. Unfortunately, these approaches ***do not provide a general and principled way to improve LLM-as-a-judge frameworks***. Indeed, ad-hoc approaches target one spurious factor (e.g., generation length (Ye et al., 2025)) and leave others in place, or make implicit assumptions that are unlikely to hold (e.g., majority vote and unweighted averages assume access to independent and equally reliable judges).

These difficulties motivate the need for a *general* and *principled* approach to LLM-as-a-judge ensembles. We provide one through the lens of probabilistic graphical models—a classic framework that can be used for model-

ing and aggregating viewpoints. Concretely, we recast multi-judge evaluation as probabilistic inference in a ***higher-rank latent variable Markov Random Field*** (***MRF***). This enables us to model and deal with key challenges in LLM-as-a-judge ensembles:

- **No access to ground-truth scores**: One latent variable in the MRF represents a ground-truth quality for the generation being evaluated; we have *no access to it* and never observe it.
- **Unknown spurious factors**: Other latent MRF components model unknown and general distractors or spurious correlations that are associated with—but not causal—to generation quality. These might include generation length, verbosity, and other factors.
- **Complex correlations**: Judges may have correlations beyond their voting behavior, due to the use of shared data for training or shared base models. These correlations are flexibly modeled by MRF interactions between variables corresponding to judges.

Higher-rank latent variable MRFs provide a ***principled and general recipe to automated model-based evaluation***. The recipe is to learn the MRF (i.e., learn its parameters, including those for the latent variables, from observed data—LLM votes) then compute a posterior estimate of the latent ground-truth quality. However, learning such higher-rank latent MRFs is challenging. We must address **1)** how can we learn the model parameters despite never observing any latent variable, and **2)**: how can we identify which latent corresponds to a ground-truth quality score (rather than spurious factors)?

We tackle this technical challenge with a two-pronged approach. First, to address 1), we introduce a novel two-stage technique to learn higher-rank latent MRFs. It ***combines a sparse plus low-rank decomposition that partially recovers the model with a second tensor decomposition step to***

fix the remaining parameters. While each approach has been individually used to learn latent factor models in more limited settings, our new combined approach is substantially more general. Second, to handle 2), we introduce a variety of approaches that boost identifiability such as anchoring latent factor to human labels, enforcing balanced variable loadings, which enables us to distinguish between latent variables corresponding to ground-truth scores versus spurious factors or confounders.

In addition to our basic estimator, we develop an adaptive approach that *augments an existing set of judges with new, generated judges*. The augmented evaluators we focus on in particular are *programmatic judges*—programs that can perform evaluation that are themselves the output of LLMs. We find that such programmatic judges enable (1) boosting the signal for evaluation and (2) facilitate the expansion of the judge set, leading to improved accuracy and robustness.

Summary of Contributions.

1. We propose CARE, the first *confounder-aware aggregation* framework that explicitly models shared latent confounders among LLM judges, unifying single-judge debiasing with principled statistical fusion.

2. We prove identifiability and derive finite-sample error bounds, showing that our estimator can reliably aggregate judge scores even when confounders are non-trivial.

3. We characterize the inherent model misspecification error incurred by methods ignoring confounders, demonstrating CARE's advantage over independence-based competitors.

4. We demonstrate consistent gains on diverse public benchmarks, reducing aggregation error by up to **25.15%** and proving *more effective and efficient* than individual-judge intervention strategies.

5. We show that CARE *robustly integrates* programmatic judges and supports *progressive expansion* of the evaluator pool, *consistently outperforming baseline aggregation methods*.

By explicitly modeling confounders during aggregation, our framework offers a principled alternative to current heuristic pipelines and substantially enhances the reliability of LLM-as-a-judge.

8 BACKGROUND AND OVERVIEW

We start with brief background on automated evaluation and probabilistic graphical models.

LLM-as-a-judge. The goal of these techniques is to efficiently and cheaply evaluate model generations. Large language models can act as inexpensive, fast proxies for human raters by returning (i) *scalar quality scores* (e.g., 1–10 Likert or percentile ranks) (Zhu et al., 2023; Wang et al.; Shi et al., 2024), (ii) *pairwise preferences* that indicate which of two candidate answers is better—an output format popularized by RLHF pipelines (Ouyang et al., 2022; Bai et al., 2022), and (iii) *categorical labels* such as error type, topic tag, or correctness flags (Gilardi et al., 2023; Chen et al., 2024). As individual LLM judges are often biased, recent work (Verga et al., 2024a) deploys *multiple* LLM judges and aggregates their opinions—via majority vote, average pooling, or other techniques—to boost robustness and accuracy. Our framework builds on this line of work *but seeks a more principled approach to multi-judge aggregation that explicitly models shared confounders and correlated errors*.

Graphical Models and Latent-Variable MRFs. Graphical models represent conditional independence in multivariate distributions, with Markov Random Fields (MRFs) being particularly valuable due to their effective structure learning and efficient inference capabilities, enabling the discovery of meaningful dependency structures from data for probabilistic reasoning at scale. In our LLM-as-a-judge setting, we employ MRFs to jointly model judge scores (J), confounding factors (C), and latent quality variables, allowing us to capture intricate dependencies among LLM evaluations while maintaining efficient inference and learnability. When key influences are unobserved, such as the true quality signal, augmenting an MRF with latent nodes allows for the recovery of this hidden structure or "ground-truth" variables from noisy observations. This latent-variable

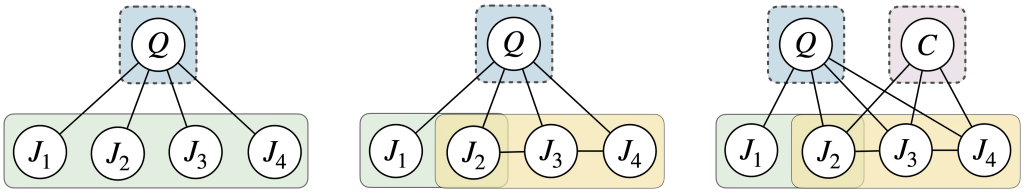
MRF perspective is crucial in our context, offering a principled method to estimate the latent, true-quality signal from observable judges' scores while accounting for correlated judging errors.

9 CARE: CONFOUNDER-AWARE AGGREGATION FOR RELIABLE EVALUATION

We introduce CARE (Confounder-Aware Aggregation for Reliable Evaluation), our graphical model-based aggregation framework that robustly estimates the true quality of LLM-as-a-judge assessments. Our framework explicitly models the influence of a latent true-quality variable and additional latent confounders on the observed scores provided by multiple judges.

9.1 Graphical Model Framework And Assumptions

For each prompt-response pair, we observe scores $J = (J_1, \ldots, J_p)^\top$ from p judges. We assume these observed scores depend on latent variables including one *true quality variable* Q and one or more *confounders* $C =$



(a) Naive: only true-quality (b) Connection-aware (c) Confounder-aware (Ours)

Figure 9.1: **Graphical models for aggregating judge scores under different structural assumptions.** **(a)** A naive model assumes scores reflect only a true latent quality (Q) and that all judges are equally reliable and represent independent views. **(b)** Connection-aware approach models intra-judge interactions $(J_2 - J_3 - J_4)$, but still assumes the presence of a single latent quality score. **(c)** Our Confounder-aware model *explicitly* introduces additional latent confounders (C) influencing judge scores.

$(C_1, \ldots, C_k)$, which we define as $H = (Q, C)$. Our graphical model encodes the conditional independence structure among the nodes in (J, Q, C): if there is no edge between a pair of nodes, they are independent conditioned on the other nodes. An example is shown on the right in Fig. 9.1. We assume this structure is *sparse*; i.e., there are not too many edges in the graph, and make this precise later on. We note the main difference between panel (b) and (c) in Fig. 9.1 is that by explicitly modeling confounding factors, we can interpret why two judges are correlated (e.g., both influenced by a shared position-bias node) rather than merely observing an unexplained edge as in (b).

This framework is quite general and is compatible with a variety of distributions. For example, we may take J, Q, C to involve discrete variables, Gaussians, or mixed models. We can take the model to be an MRF or alternatively a mixture model. Our approaches are compatible with a broad range of choices, with practitioners able to select the most suitable modeling assumptions for their settings.

Goals and Assumptions. Under the chosen modeling assumptions, our goal is to learn the distribution over J, Q, C. This involves handling ***three challenges***. First, **C1**: *we never observe the latents in* H—neither ground truth nor confounders. Second, **C2**: we *cannot assume any particular interaction* in the graph. Third, **C3**: even if we recover the model parameters, we must be able to distinguish between Q and the confounders C *to identify the model*. The latter is required to discover ***which latent is the ground-truth quality—and which is a confounder***. Once these obstacles are overcome, we seek to perform aggregation, e.g., compute a posterior $P(Q|J)$, the Bayesian estimate for the latent true quality conditioned on all observable judge scores.

In the following, we will work under the assumption that the judge scores J conditioned on the latents form a multivariate Gaussian distribution, i.e., $J \mid H \sim \mathcal{N}(\mu_H, \Sigma)$, where μ_H is the conditional mean of observable

variables. We defer other scenarios to the Appendix.

9.2 CARE Algorithm

The idea behind CARE is to examine two techniques, each of which is stymied by one of the obstacles **C2** or **C3** and to *delicately combine them in a novel way*. First, the sparsity of the conditional independence graph is encoded into an two-dimensional object that can be empirically estimated (e.g., the observable covariance matrix, or a cross-moment matrix). However, the presence of the latent variables (**C1**) obscures this structure—but a *sparse + low-rank decomposition* can reveal it Chandrasekaran et al. (2012). However, while we can decompose the resulting low-rank term via SVD in the hope of identifying the model, we can only do so *up to rotations*. Therefore we are blocked by **C3**.

Conversely, tensor product decompositions Anandkumar et al. (2014) exploit tensor rigidity to enable this decomposition to be uniquely identified. However, for these techniques the judges must be independent conditioned on the latents—and we cannot assume this by **C2**.

CARE (Algorithm 4) combines these approaches. First, it estimates the underlying graph structure from the observed judge scores via the sparse + low-rank decomposition, overcoming **C1** and **C2**. It then uses recovered sparse term to estimate the graph and discover subsets of judges with sufficient conditional independence. These sets are then used to construct a tensor that can be decomposed via standard approaches (e.g., tensor power method) to recover the model, mitigating **C3**.

This procedure is then followed by a symmetry-breaking step. This requires a weak assumption on the quality of the judges; in practice, even this assumption can be removed by employing simple heuristics to identify the true-quality factor among the latent factors. Finally, we aggregate judge scores into robust evaluations by weighting according to loadings from

Algorithm 4 CARE: Confounder-Aware Aggregation for Reliable Evaluation

Input: Score matrix $J \in \mathbb{R}^{n \times p}$, parameters (γ_n, τ), decomposition method $\mathcal{D} \in$ SVD, Tensor

Output: Estimated True Quality $\{\hat{q}^{(i)}\}_{i=1}^{n}$

1: **Graph Sparse Structure Estimation:** Compute appropriate observed matrix $f(J)$.

2: **Sparse + low-rank decomposition:**

$$(\hat{S}, \hat{L}) \leftarrow \arg\min_{S,L} \ \tfrac{1}{2}\|f(J) - S - L\|_F^2 + \gamma_n(\|S\|_1 + \tau\|L\|_*)$$

3: **Latent Factor Extraction:**

4: **if** $\mathcal{D} =$ SVD **then** ▷ Fully Gaussian scenario

5: Compute $U\Lambda U^\top \leftarrow \text{SVD}(\hat{L})$, where $U \in \mathbb{R}^{p \times h}$

6: **else if** $\mathcal{D} =$ Tensor **then** ▷ Binary-Gaussian mixture scenario

7: Partition judges into independent groups using $\hat{S}$

8: Form empirical third-order tensor from judge groups

9: Run tensor decomposition, obtain latent conditional means μ_{qc} and mixture proportions π_{qc}

10: **end if**

11: **Symmetry Breaking:** Identify the true-quality factor using heuristics described in 9.3

12: **Latent Quality Estimation:** Use the identified quality factor, compute $\hat{q}^{(i)}$ for each example, where $\hat{q}^{(i)} = P(Q = 1 \mid J_i)$ for mixture model or $\hat{q}^{(i)} = \mathbb{E}[Q \mid J]$ for fully Gaussian

the identified quality factor.

We study two special cases to build our intuition; more general settings are shown in the Appendix.

CARE For Gaussian Mixtures. We have binary latents (Q, C) with $\Pr\big(Q = q,\ C = c\big) = \pi_{qc}$, where the judges follow a Gaussian conditional distribution with mean $\mu_{qc} \in \mathbb{R}^p$ and covariance Σ:

$$J \,\big|\, (Q = q,\ C = c) \ \sim\ \mathcal{N}\big(\mu_{qc},\ \Sigma\big), \qquad (q, c) \in \{0, 1\}^2.$$

Here, performing the sparse + low-rank decomposition and obtaining $\hat{L}$ is insufficient: the eigen-decomposition of $\hat{L}$ does not directly yield identifiable latent-judge connections. We rely on third-order tensor statistics to identify conditional distributions explicitly:

$$\mathbb{E}(X_1 \otimes X_2 \otimes X_3 \mid Q, C) = \mathbb{E}(X_1 \mid Q, C) \otimes \mathbb{E}(X_2 \mid Q, C) \otimes \mathbb{E}(X_3 \mid Q, C),$$

where judges are partitioned into independent groups X_1, X_2, X_3 using the learned sparse structure $\hat{S}$. Performing a tensor decomposition yields the conditional means μ_{qc} and mixture proportions π_{qc}. Then, applying Bayes' rule allows estimation of latent variables given observed scores:

$$P(Q = 1|J) \propto \pi_{10}\mu_{10} + \pi_{11}\mu_{11}. \tag{9.1}$$

CARE for Fully Gaussian Models. Under the fully Gaussian assumption, latent variables H are continuous, and the inverse covariance matrix (the *precision* matrix) encodes independence:

$$\Sigma = \mathrm{Cov}\big[(J, H)^\top\big], \quad \Sigma^{-1} = K = \begin{pmatrix} K_{JJ} & K_{JH} \\ K_{HJ} & K_{HH} \end{pmatrix}, \quad S = K_{JJ}, \quad L = K_{JH}K_{HH}^{-1}K_{HJ}.$$

If assuming connections K_{JH} between latent variables and judges are orthogonal and no direct connections among latent variables (i.e. K_{HH} is diagonal), the low-rank matrix $\hat{L}$ admits eigen-decomposition $\hat{L} = U\Lambda U^\top$, where eigenvectors in U directly correspond to latent-judge edges (K_{JH}), and eigenvalues correspond to K_{HH}. Each eigenvector represents how one latent variable influences observable judges. With these edges recovered, the conditional mean of true quality Q can be estimated by $\mathbb{E}(Q \mid J) = K_{QQ}^{-1}K_{QJ}J$, a weighted linear combination of observed scores.

The fully Gaussian model prevents decomposing the low-rank term uniquely (due to rotational invariance). This holds regardless of whether we apply SVD or a tensor decomposition, leading to the special handling

in Algorithm 4. As a result, in this case, orthogonal and independent latent assumptions are needed for identifying the latent-judge connection. This works the best when each judge is connected to exactly one latent variable. If a judge depends on *both* the confounder C and the true quality Q with comparable weights, the recovered columns $\{\hat{\mu}_r\}$ are only identifiable up to an arbitrary rotation, causing estimation errors.

9.3 Heuristics for Identifiability and Robust Estimation

Any instantiation of CARE will require symmetry-breaking procedures for latent variable identifiability. For example, the fully Gaussian case needs a heuristic to identify the true-quality direction among latent factors, distinguishing Q from confounders C. In the binary-Gaussian mixture scenario, an additional step resolves ambiguity between latent states ($Q = 0$ vs. $Q = 1$). Doing so will require additional information that can come from modeling assumptions, the use of ground-truth samples, or heuristics. We detail some examples below:

Identifying True-Quality Factor for Joint-Gaussian Model. We introduce heuristics particularly aimed at distinguishing the true-quality latent variable from confounding latent variables. First, the *human-anchor criterion* leverages a small validation set containing human ratings. By including these human judgments in the graphical model, we anchor the latent quality variable to ground truth by selecting the latent factor exhibiting the strongest connection to the human evaluations. Second, we apply a *loading balance heuristic*, identifying the true-quality factor as one that loads broadly and with similar magnitude across all competent judges. Conversely, factors dominated by a few judges typically indicate shared confounding rather than true quality.

Identifying Latent States for Mixed Model. In scenarios such as the

tensor-based method, symmetry breaking additionally involves distinguishing latent states corresponding to different quality levels (e.g., $Q = 0$ versus $Q = 1$). In practice, we can use known labeled samples (such as high-quality examples) to anchor and identify latent-state configurations. By comparing different latent configurations with these known labeled samples, we select the latent-state assignment that best aligns with empirical observations, effectively removing latent state ambiguity.

We formalize the graphical model under joint Gaussian distribution and notation (Section 10.1), then discuss the identifiability of graph structure with exact and approximate recovery (Section 10.2) and quantify the sample complexity required for consistent recovery of our SVD-based algorithm (Section 10.3). Next, we present the model misspecification error when confounding factor is not correctly characterized (Section 10.4). Finally, we discuss sample complexity required for tensor-based algorithm under mixed Gaussian distribution (Section 10.4). All proofs are deferred to Appendix B.4.

10.1 Model and Notation

We discuss the model under joint Gaussian distribution where all variables follow the same definitions as in Section 9. Briefly, $J = (J_1, \dots, J_p)^\top$ stacks the p observable judge scores, and $H = (Q, C_1, \dots, C_k)^\top$ collects the $h = k + 1$ latent variables.

$$\Sigma = \mathrm{Cov}\big[(J, H)^\top\big], \qquad \Sigma^{-1} = K = \begin{pmatrix} K_{JJ} & K_{JH} \\ K_{HJ} & K_{HH} \end{pmatrix},$$

where the subscript J (resp. H) refers to observable (resp. latent) coordinates.

The observable block factorizes via the Schur complement:

$$(\Sigma_{JJ})^{-1} = S + L, \quad S = K_{JJ}, \quad L = K_{JH}\, K_{HH}^{-1}\, K_{HJ}.$$

Here Σ_o is the covariance matrix of observable variables, $S \in \mathbb{R}^{p \times p}$ is sparse and encodes direct conditional edges among judges, L is low-rank with $\mathrm{rank}(L) \leqslant h$ and captures dependencies mediated by the latent

variables. Entry $(K_{JH})_{i\ell}$ is the edge weight between judge i and latent factor ℓ.

10.2 Graph Structure Identifiability

While (S, L) can be recovered (e.g. via convex sparse-plus-low-rank regularization (Chandrasekaran et al., 2012), the finer structure of K_{JH} is usually not identifiable from L. For example, for arbitrary rotation matrix $R \in \mathbb{R}^{h\times h}$, $L = (K_{JH}K_{HH}^{-1/2}R)(R^\top K_{HH}^{-1/2}K_{HJ})$, this indicates one cannot distinguish $K_{JH}K_{HH}^{-1/2}$ from $K_{JH}K_{HH}^{-1/2}R$ without further constraints. Hence, we need to impose additional assumptions:

Assumption 10.2.1 (Latent–latent independence and eigen-gap)**.** $K_{HH} = \mathrm{diag}(d_1, \ldots, d_h)$ with $d_1 > d_2 > \cdots > d_h > 0$.

Assumption 10.2.2 (Orthogonal latent–observable connections)**.** The columns of K_{JH} are orthogonal, i.e. $K_{JH}^\top K_{JH}$ is diagonal. A special case is the *disjoint-support* model where each judge connects to exactly one latent factor.

Next, we provide an exact recovery result given the above assumptions.

Theorem 10.2.3 (Exact Recovery)**.** *Under Assumptions 1 and 2, columns in* K_{JH} *are identifiable up to column permutations and sign flips.*

Real-world data rarely satisfy the exact orthogonality in Assumption 10.2.2. To assess robustness, consider the following perturbed connection matrix:

$$\tilde{K}_{JH} = K_{JH} + E, \qquad \|E\|_2 \text{ small.}$$

The associated low-rank part is $\tilde{L} = \tilde{K}_{JH}K_{HH}^{-1}\tilde{K}_{HJ}$. Let the eigen-pairs of $L = K_{JH}K_{HH}^{-1}K_{HJ}$ and $\tilde{L}$ be $\{(\lambda_i, u_i)\}_{i=1}^h$ and $\{(\tilde{\lambda}_i, \tilde{u}_i)\}_{i=1}^h$, ordered so that $\lambda_1 > \cdots > \lambda_h > 0$, and denote the eigen-gap by

$$\delta_i = \min_{j\neq i} |\lambda_i - \lambda_j| > 0.$$

Theorem 10.2.4 (Stability under approximate orthogonality)**.** *For every* $i \in [h]$,

$$\|\hat{u}_i - u_i\|_2 \;\leqslant\; \frac{2\|K_{HH}^{-1}\|_2\,\|E\|_2}{\delta_i} \;+\; O\big(\|E\|_2^2\big).$$

This indicates that latent–observable directions remain identifiable (up to column permutations and sign flips) whenever the perturbation norm $\|E\|_2$ is small relative to the eigen-gap δ_i. We defer the proof to Appendix B.4.

10.3 Sample Complexity Bound

We now quantify how many i.i.d. samples are needed for the two–stage estimator in Algorithm 4 to recover the latent–observable directions $K_{JH} \in \mathbb{R}^{p\times h}$.

As detailed in Algorithm 4, our estimator for K_{JH} proceeds in two stages: first, a sparse + low-rank decomposition of sample precision matrix. Second, we extract the latent–observable directions by taking the rank-h eigen-decomposition $\hat{L}_n = \sum_{i=1}^{h} \hat{\lambda}_i\, \hat{u}_i \hat{u}_i^\top$ and setting $\hat{K}_{JH} := [\hat{u}_1, \ldots, \hat{u}_h]$.

Theorem 10.3.1 (Sample complexity for recovering K_{JH})**.** *Let*

$$L^* = K_{JH} K_{HH}^{-1} K_{HJ} \in \mathbb{R}^{p\times p}$$

have distinct eigenvalues $\lambda_1 > \cdots > \lambda_h$ *and define the (global) eigengap* $\delta := \min_{1\leqslant i<j\leqslant h}|\lambda_i - \lambda_j|$. *Assume the identifiability, incoherence, and curvature conditions of Chandrasekaran et al. (2012). Then for any* $\epsilon > 0$*, with probability at least* $1 - 2e^{-\epsilon}$*,*

$$\max_{i\leqslant h} \big\| \hat{u}_i - u_i \big\|_2 \;=\; O\Big(\frac{\sqrt{\epsilon}}{\sqrt{n}\,\xi(T)\,\delta}\Big),$$

where n *is the sample size,* $\hat{u}_i$ *and* u_i *are the* i*-th eigenvectors of* $\hat{L}_n$ *and* L^* *respectively.* $T = T(L^*)$ *is the tangent space of* L^**,* $\xi(T)$ *is the curvature constant from Chandrasekaran et al. (2012).*

We defer the proof to Appendix B.4. At a high-level, we adapt the identifiability, incoherence and curvature conditions from Theorem 4.1 of Chandrasekaran et al. (2012) and combine it with extended result of Davis-Khan's theorem (Yu et al., 2015).

This bound shows that the column-wise ℓ_2 error decays at the standard parametric rate $n^{-1/2}$, and is attenuated by both the manifold curvature $\xi(T)$ and the eigengap δ. Achieving an accuracy of at most $\alpha \in (0, 1)$ therefore requires

$$n = \tilde{O}\left(\frac{\epsilon}{\xi(T)^2 \delta^2 \alpha^2}\right)$$

samples, up to universal constants and log-factors.

10.4 Misspecification Error

Many label aggregation frameworks (e.g.,Bach et al. (2019); Fu et al. (2020); Shin et al. (2022)) assume a *single* latent variable that explains the observed labels. However, in setups like LLM-as-a-judge, the scores may be influenced by additional latent factors or confounders that also affect the observed annotations. Ignoring these *confounder* latents leads to model misspecification, which can bias the aggregated labels. We characterize this bias and analyze its impact on the estimated aggregation weights.

Let $L^* = \sum_{\ell=1}^{h} \frac{1}{d_\ell} \mathbf{k}_\ell \mathbf{k}_\ell^T$ be the true rank-h low-rank component of the observable precision matrix, derived from the latent-observable connection matrix $K_{JH} = [\mathbf{k}_1, \ldots, \mathbf{k}_h]$ and latent-latent precision $K_{HH} = \mathrm{diag}(d_1, \ldots, d_h)$. Let $\mathbf{u}_1^{\text{true}} = \mathbf{k}_1 / \|\mathbf{k}_1\|_2$ be the true direction of influence for the quality score latent variable Q (assuming $\mathbf{k}_1 \neq \mathbf{0}$).

Define $\mathbf{A} = \frac{1}{d_1}\mathbf{k}_1\mathbf{k}_1^T$. Its principal (and only non-zero) eigenvalue is $\lambda_1 = \frac{1}{d_1}\|\mathbf{k}_1\|_2^2$, and its spectral gap (to its other zero eigenvalues) is $\delta = \lambda_1$. Let $\mathbf{E} = \sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T$ be the confounding component, so $L^* = \mathbf{A} + \mathbf{E}$. Let $\mathbf{v}_1$ be the principal unit-norm eigenvector of L^*. When a rank-1 model is fitted, the estimated direction is $\hat{\mathbf{u}}_1^{pop} = \mathbf{v}_1$.

Theorem 10.4.1. *If* $\|\mathbf{E}\|_{op} \leqslant \delta/2$, *the* ℓ_2 *deviation of the estimated direction* $\mathbf{v}_1$ *from* $\mathbf{u}_1^{true}$ *is bounded by:*

$$\left\|\mathbf{v}_1 - s\mathbf{u}_1^{true}\right\|_2 \leqslant \frac{2\|\mathbf{E}\|_{op}}{\delta} = \frac{2\left\|\sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T\right\|_{op}}{\frac{1}{d_1}\|\mathbf{k}_1\|_2^2}$$

for a sign $s = \pm 1$ (*chosen so that* $s(\mathbf{u}_1^{true})^T\mathbf{v}_1 \geqslant 0$).

We provide the following theoretical guarantees for our Algorithm 4.
Identifiability of the Latent Structure. To ensure identifiability of the latent structure, we introduce assumptions on latent independence and orthogonality of latent-observable connections. Under these assumptions, we prove exact recovery of the latent directions, as well as stability under mild perturbations from orthogonality (see Appendix 10.2).
Sample Complexity Bound. We derive sample complexity bounds for consistent estimation of latent-observable connections, demonstrating how estimation accuracy depends on factors like eigengaps and manifold curvature (Appendix 10.3).
Model Misspecification Error. We analyze errors arising from model misspecification—specifically, the bias introduced when confounding latent factors are omitted—and provide explicit bounds on the resulting errors in estimated conditional means (Appendix 10.4).

The theorem quantifies the directional bias in the estimated influence of Q when confounders are ignored. This bias is proportional to the collective "strength" of confounders in the precision domain (numerator) and inversely proportional to Q's own "strength" (denominator). Fitting a

rank-1 model forces this bias, while a higher-rank model offers the capacity to separate these influences.

Corollary 10.4.2 (Error Bound for Estimated Conditional Mean of Q)**.** *Denote the true conditional mean of true quality score latent variable* Q *given the observable variables* $O = (J_1, ..., J_p)$ *be denoted by* $\mathbb{E}[Q|O]_{true}$*. Then,* $\mathbb{E}[Q|\mathbf{o}]_{true} = -\frac{||\mathbf{k}_1||_2}{d_1}(\mathbf{u}_1^{true})^T\mathbf{o}$*. Let an estimated conditional mean with the misspecified direction,* $\mathbb{E}[Q|\mathbf{o}]_{mis}$*, be formed using the misspecified direction* $\mathbf{v}_1$ *be* $\mathbb{E}[Q|\mathbf{o}]_{mis} = -\frac{||\mathbf{k}_1||_2}{d_1}(s \cdot \mathbf{v}_1)^T\mathbf{o}$*, where* $s = \pm 1$ *is chosen such that* $s \cdot (\mathbf{u}_1^{true})^T\mathbf{v}_1 \geqslant 0$*. Then, the absolute error in the estimated conditional mean due to the directional misspecification is bounded by:*

$$|\mathbb{E}[Q|\mathbf{o}]_{mis} - \mathbb{E}[Q|\mathbf{o}]_{true}| \leqslant \frac{2\left|\left|\sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T\right|\right|_{op}}{||\mathbf{k}_1||_2} ||\mathbf{o}||_2$$

This holds if the condition from the main theorem, $||\mathbf{E}||_{op} \leqslant \delta/2 = \frac{1}{2d_1}||\mathbf{k}_1||_2^2$*, is met, where* $\mathbf{E} = \sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T$*.*

This corollary shows that the error in the estimated conditional mean of Q (due to using the misspecified direction for Q's influence) scales with:

- The magnitude of the observable vector $\mathbf{o}$ (specifically, $||\mathbf{o}||_2$).
- The collective strength of the confounding latent variables in the precision domain ($\left|\left|\sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T\right|\right|_{op}$).
- Inversely with the ℓ_2-norm of the true connection weights of Q ($||\mathbf{k}_1||_2$).

Especially, we see that strong confounders widen the gap bound, whereas heavier connection weights to the true score shrink it. Put differently, ***misspecification hurts most when confounders are strong and the quality signal is weak.***

Sample Complexity for CARE tensor algorithm

Assumption 10.4.3 (Model and identifiability)**.** Let $J = (X_1^\top, X_2^\top, X_3^\top)^\top \in \mathbb{R}^p$ $(p = p_1 + p_2 + p_3)$ be one observations i.i.d generated as

$$(Q, C) \sim \text{Multinomial}(\{\pi_{qc}\}_{q,c\in\{0,1\}}), \qquad X_\ell \mid (Q = q, C = c) \;\sim\; \mathcal{N}\big(\mu_{qc}^{(\ell)},\, \Sigma\big),$$

with $\ell \in \{1, 2, 3\}$. Write $r \in [4] \leftrightarrow (q, c) \in \{0, 1\}^2$ and define $w_r := \pi_{qc}$, $a_r := \mu_{qc}^{(1)} \in \mathbb{R}^{p_1}$, $b_r := \mu_{qc}^{(2)} \in \mathbb{R}^{p_2}$, $c_r := \mu_{qc}^{(3)} \in \mathbb{R}^{p_3}$.

(A1) **Block-conditional independence.** $X_1 \perp X_2 \perp X_3 \mid (Q, C)$.

(A2) **Full-rank moment tensor.** The population third-order moment $M := \mathbb{E}[X_1 \otimes X_2 \otimes X_3] = \sum_{r=1}^{4} w_r\, a_r \otimes b_r \otimes c_r$ has rank 4, with $\pi_{\min} := \min_r \pi_r > 0$ and $\lambda_{\min} := \min_r \|a_r\|_2 \|b_r\|_2 \|c_r\|_2 > 0$.

(A3) **Non-degenerate covariance.** $\sigma_{\max}^2 := \|\Sigma\|_{\text{op}} < \infty$.

(A4) **Spectral gap.** The CP factors are uniquely defined up to scaling/sign and satisfy the eigenvalue-gap condition of Theorem 5.1 in Anandkumar et al. (2014). Denote that gap by $\delta > 0$.

(A5) **Correct graph partition.** There exist a graph partition such that judges between different groups are conditionally independent. Step A of Algorithm 9 returns the true groups $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$.

Theorem 10.4.4 (Sample complexity of CARE tensor step)**.** *Fix* $0 < \varepsilon < 1$ *and let the assumptions above hold. Run Algorithm 2 (CARE) on* n *i.i.d. samples to obtain* $\{\hat{\mu}_{qc}, \hat{\pi}_{qc}\}_{q,c\in\{0,1\}}$*. Under Assumption 10.4.3, there exist universal constants* $C_1, C_2 > 0$ *such that if*

$$n \;\geqslant\; C_1\, \frac{\sigma_{\max}^6}{\delta^2\, \pi_{\min}^2}\; p\, \log\!\big(p/\varepsilon\big),$$

then with probability at least $1 - \varepsilon$

$$\max_{q,c} \|\hat{\mu}_{qc} - \mu_{qc}\|_2 \leqslant C_1 \frac{\sigma_{\max}^3}{\delta} \sqrt{\frac{p \log(p/\varepsilon)}{n}},$$
$$\max_{q,c} |\hat{\pi}_{qc} - \pi_{qc}| \leqslant C_2 \sqrt{\frac{p \log(p/\varepsilon)}{n}}.$$

We defer the proof to B.4.

11 EXPERIMENTAL RESULTS

We evaluate the effectiveness of CARE across diverse experimental setups, encompassing synthetic, semi-synthetic, and real-world scenarios. Our goal is to validate the following key claims:

- **Improving aggregation of LLM judge:** CARE produces more accurate and robust aggregate scores from multiple LLM judges compared to existing methods. (Section 11.1)
- **Effective Integration of Program Judges:** CARE integrates programmatic judges, known to have high bias, by explicitly modeling their biases (Huang et al.) (Section 11.2).
- **Evolving Jury via Progressive Program Judge Expansion:** CARE effectively incorporates an expanding pool of judges, demonstrating consistent improvements in aggregation performance as judges are progressively added (Section 11.3).
- **Greater Robustness than Individual Intervention:** CARE is competitive against interventions at the individual judge level, which typically require extensive manual tuning (Section 11.4).
- **Demonstrating Robustness under Controlled Confounding Factors:** CARE remains accurate when evaluations are deliberately affected by controlled biases, as demonstrated by the semi-synthetic data from Chen et al. (2024) (Section 11.5).
- **Validating Theoretical Results in a Fully Controlled Setting:** We empirically validate our theoretical results through synthetic experiments (Section 11.6).

Datasets & Metrics. We use FeedbackQA (Li et al., 2022), UltraFeedback (Cui et al., 2023), and HelpSteer2 (Wang et al., 2024b) datasets for re-

sponse scoring. Each of these dataset has a ground-truth quality score label, specified as following: in FeedbackQA, each answer is rated by humans on a 1–5 helpfulness scale; in UltraFeedback, responses receive a 0–10 score aggregated from human annotators and GPT-4; and in HelpSteer2, "helpfulness" is rated on a 0–4 scale by human evaluators. Performance is benchmarked using Mean Absolute Error (MAE) to measure numerical accuracy and Kendall's τ rank correlation (Kendall, 1938) to evaluate ranking consistency, accommodating variations in judge scales and calibration.
Baselines. We compare CARE to following baseline aggregation methods: (i) majority voting (MV), (ii) simple averaging (AVG) (Li et al., 2024b), (iii) discrete-based weak supervision (WS) (Bach et al., 2019), and (iv) continuous-based weak supervision (UWS) (Shin et al., 2022).
LLM Judges. We consider the following LLMs as judges to score responses:

- `Llama-3.2-1B` (Grattafiori et al., 2024)
- `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024)
- `Mistral-7B-Instruct-v0.3` (Jiang et al., 2023)
- `Qwen3-0.6B` (Team, 2025)
- `Qwen3-1.7B` (Team, 2025)
- `Qwen3-4B` (Team, 2025)
- `Qwen3-8B` (Team, 2025)
- `Phi-4-mini-instruct` (Abouelenin et al., 2025)
- `gemma-3-1b-it` (Team et al., 2025)
- `gemma-3-4b-it` (Team et al., 2025)

11.1 Improving Aggregation of LLM judges

Setup. We compare aggregation methods using the 10 LLM judges listed above. To ensure consistency, we adapt the prompt template from Roucher (n.d.), modifying it to fit our experimental setup. The exact used prompt is provided in Appendix B.5.
Results. We present aggregation performance in Table 11.1. The CARE approach consistently outperforms baseline methods. Specifically, CARE

Table 11.1: Aggregation performance across different datasets, measured by MAE and Kendall's τ CARE outperforms baseline methods in most cases.

	FeedbackQA		HelpSteer2		UltraFeedback	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
MV	0.8812	0.3703	0.9951	0.1629	0.8522	0.2985
AVG	0.8492	0.4497	0.9822	0.1611	0.6860	0.3621
WS	0.8144	0.4401	1.3030	0.1511	1.1603	0.3306
UWS	0.9051	**0.4580**	0.9849	0.1697	0.6794	0.3669
CARE	**0.7866**	0.4542	**0.9742**	**0.1805**	**0.6379**	**0.3806**

Table 11.2: Performance on different datasets using both LLM and program judges. Program judges are beneficial in FeedbackQA but may introduce noise in HelpSteer2 and UltraFeedback. In both cases, CARE consistently outperforms other baselines.

	FeedbackQA		HelpSteer2		UltraFeedback	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
MV	0.8607	0.3815	1.0244	**0.1465**	0.8751	0.3179
AVG	0.8128	0.4671	1.1012	0.1268	1.0371	**0.3733**
UWS	0.8179	**0.4816**	0.9992	0.1040	0.9534	0.3047
CARE	**0.7582**	0.4796	**0.9800**	0.1398	**0.7351**	0.3520

achieves the lowest MAE on FeedbackQA (0.7866) and UltraFeedback (0.6379), outperforming the majority vote (MV) baseline by **10.74%** and **25.15%**, respectively. These gains highlight CARE's ability to model correlations among LLM judges and mitigate compounding biases.

11.2 Effective Integration of Program Judges

Setup. We integrate our LLM-based evaluators with ten program judges, each encoding their evaluation logic in program code and synthesized by OpenAI's GPT-4o (Hurst et al., 2024). These judges are designed to assess response quality through specific, individual criteria, such as *structure, readability, safety, relevance, and factuality*. While cost-effective to construct them, their deterministic nature may introduce systematic biases, potentially leading to noisy signals. Details of program judge generation

process are provided in Appendix B.5.

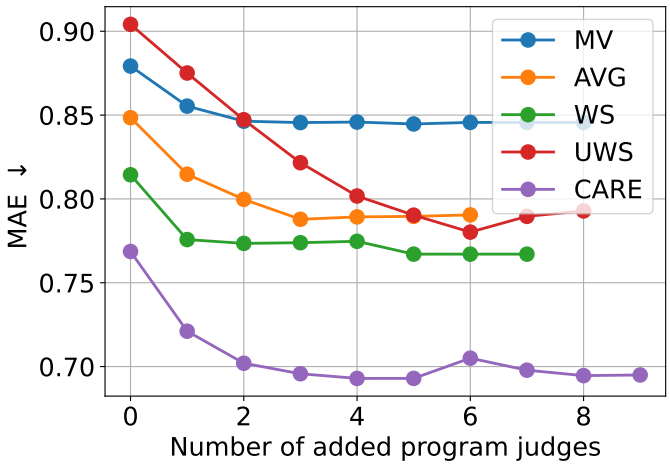


Figure 11.1: Progressive judge selection on the FeedbackQA dataset. CARE robustly integrates new judges and consistently outperforms baseline aggregation methods.

Results. Table 11.2 presents the integration results. Adding program judges enhance performance on FeedbackQA, where CARE achieves the lowest MAE (0.7582) and highest τ (0.4796), outperforming the MV baseline's MAE by **11.92%**. However, performance declines on HelpSteer2 and UltraFeedback, where CARE records MAEs of 0.9800 and 0.7351, respectively, still outperforming MV by **4.33%** and **15.99%**. Despite these variations, CARE consistently exceeds baselines on MAE across all datasets, demonstrating its effectiveness when encountering noisier signals for aggregation.

11.3 Progressive Judge Expansion

Setup. Next, we start with a fixed set of LLM judges and progressively add program judges from a pool of 23. At each step, we greedily select

Table 11.3: Comparison with aggregation methods using individually intervened LLM judges. While other baselines aggregate scores from debiased LLM judges, CARE operates directly on raw outputs.

	FeedbackQA		HelpSteer2		UltraFeedback	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
MV	0.8004	0.9640	0.9951	0.1629	0.8562	0.2799
AVG	0.8029	0.4412	0.9822	0.1611	0.6801	0.3704
WS	**0.7674**	0.4429	1.3030	0.1511	1.1516	0.3588
UWS	0.8117	0.4390	0.9849	0.1697	0.6683	0.3782
CARE	0.7866	**0.4542**	**0.9742**	**0.1805**	**0.6379**	**0.3806**

the program judge that yields the largest improvement in the validation of MAE. The process stops when no further reduction in validation MAE is observed. We evaluate aggregation methods as in previous sections, using FeedbackQA, where program judges were most beneficial.

Results. Figure 11.1 shows the experimental result. CARE achieves consistently lower error as more program judges are added, highlighting its ability to adaptively improve with additional supervision. This points to a promising direction for developing dynamic, expandable judge ensembles.

11.4 Comparison with Individual Intervention

Setup. An alternative to our confounder-aware approach is direct interventions at the individual judge level. Specifically, we compare CARE to prompt-based interventions proposed by Ye et al., which instruct LLM judges to account for known sources of bias. The intervened prompt used for this comparison is included in Appendix B.5.

Results. Table 11.3 presents the results. While bias-aware prompting improves performance in most cases, CARE remains the top performer in the majority of settings, and even when not, it is competitive with the best. This suggests that CARE can effectively mitigate biases without relying on careful prompt engineering.

Table 11.4: Robustness to artificially injected bias. CARE is particularly effective against stylistic biases such as beauty (rich content) and authority, but less effective for gender and fallacy biases, which may impact the actual quality of system answers.

	Beauty Bias		Fallacy Oversight Bias		Gender Bias		Authority Bias	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
MV	0.9190	0.3336	1.8971	-0.0284	1.7428	0.1272	0.8239	0.2977
AVG	0.5063	0.3943	**1.4007**	**0.1181**	**1.1355**	**0.2879**	0.3250	0.4288
WS	1.9225	0.3792	2.5588	0.0680	2.0217	0.2474	0.9296	0.4886
UWS	0.5080	0.4383	1.3826	0.0491	1.1646	0.2576	0.2705	0.5799
CARE	**0.3749**	**0.5334**	1.8996	0.0116	1.5985	0.2311	**0.2466**	**0.6327**

11.5 Robustness to Confounding Factors

Setup. We evaluate robustness using the dataset from Chen et al. (2024), in which LLM responses are systematically altered to introduce specific biases via targeted GPT-4 prompts. The dataset includes four types of injected bias: beauty, fallacy oversight, gender, and authority. LLM judges are prompted to assign scores from 1 to 10 for each response. Robustness is assessed by comparing aggregated scores before and after bias injection, using mean absolute error (MAE) and Kendall's τ. Lower MAE and higher Kendall's τ indicate better robustness under perturbation.

Results. Table 11.4 shows that CARE exhibits strong robustness to stylistic biases—such as beauty and authority—maintaining consistent rankings and score levels. In contrast, its robustness diminishes when facing biases that alter the factual or semantic content, including logical fallacies and gender-related framing.

11.6 Synthetic Experiments

We evaluate the performance of CARE-Tensor using simulated binary-Gaussian mixture data. Dataset details deferred to Appendix.

Sample Complexity Result. We investigate how the sample size n influences estimation accuracy. We estimate conditional means $\hat{\mu}_{qc}$ and latent

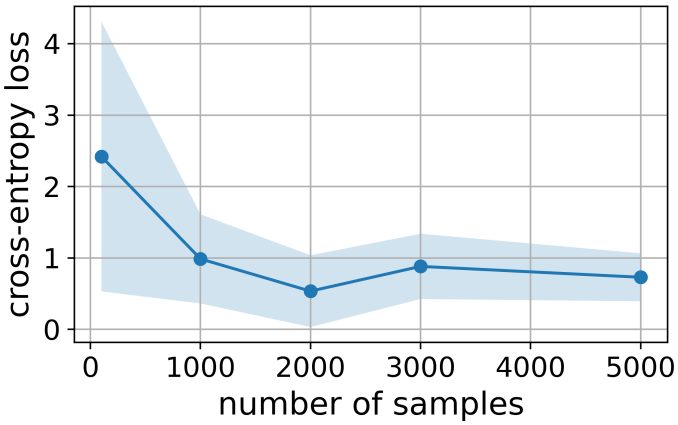


Figure 11.2: Averaged cross-entropy loss of our algorithm versus the number of samples. Markers denote average over three random seeds, and the shaded band denotes one standard deviation.

state proportions $\hat{\pi}_{qc}$ using Algorithm 9. Subsequently, we compute the posterior probabilities $P(Q = 1 \mid J)$ via the Bayesian formulation in Eq. 9.1. We measure the performance using cross-entropy loss. Lower entropy loss yields more accurate prediction. We observe a clear decreasing trend in cross-entropy loss as sample size increases.

Tensor Decomposition vs SVD. We illustrate the advantage of tensor decomposition over classical eigen-decomposition (SVD) in addressing rotation ambiguity with higher-order moments. We quantify performance using mean squared error (MSE) between true conditional means μ_{qc} and estimated means $\hat{\mu}_{qc}$. Detailed methodologies for SVD estimation are deferred to the appendix.

Evaluating across 10 random seeds, we find substantial performance differences: CARE-Tensor achieves significantly lower estimation errors with MSE ($\mathbf{0.51 \pm 0.41}$) compared to the eigen-decomposition baseline (SVD) with MSE ($\mathbf{1.18 \pm 0.74}$). This shows tensor decomposition accurately recovers conditional means without affected by rotation ambiguity.

12 RELATED WORK

We discuss related work in bias in LLM-as-a-judge, label aggregation, and highlight our contribution. An extended discussion on related work can be found in Appendix B.2.

Bias in LLM-as-a-judge. Large language models (LLMs) used as automated evaluators exhibit systematic preferences such as positional, verbosity, authority, and self-enhancement biases (Ye et al., 2025; Zhu et al., 2023). To mitigate these issues, prior work has explored prompt-based interventions (Shi et al., 2024; Jiao et al., 2024; Ye et al., 2025) and fine-tuned evaluators such as JudgeLM and PandaLM, which aims to align model judgments more closely with human preferences (Zhu et al., 2023; Wang et al.; Li et al., 2024d). ***While effective locally, these techniques debias each single LLM judge and do not address the downstream problem of aggregating multiple, potentially correlated, LLM scores.***

Label Aggregation. Classic aggregation models such as Dawid–Skene (Dawid and Skene, 1979), GLAD (Whitehill et al., 2009), and MACE (Hovy et al., 2013) infer latent truth by modeling annotator-specific error rates. Weak-supervision frameworks generalize this idea to programmatic sources (Bach et al., 2019; Fu et al., 2020; Shin et al., 2022). Recently, Hu et al. (2024) introduce GED, a framework that ensembles and denoises preference graphs from multiple weak LLM evaluators to produce consistent and reliable model rankings. Wang et al. (2025) analyzed various inference methods for decoding LLM-as-a-judge by looking at the judge probability distributions and computing statistics such as mean and mode (i.e greedy decoding) and studied how pre- vs post-aggregation of judge outputs affect the judge scores. ***However, existing methods do not account for shared confounding factors that systematically influence annotators***

or LLMs alike.

Our Contribution. We propose the first *confounder-aware aggregation* method for the LLM-as-a-judge setting. Unlike prior work that assumes independent annotator noise around a latent true score, our approach explicitly models shared latent confounders—such as verbosity or formality—that may jointly affect all judges. This bridges the gap between single-judge bias mitigation and statistical aggregation, enabling more reliable consensus scores in the presence of correlated judgment errors.

13 CONCLUSION

We introduce CARE, a confounder-aware aggregation framework that formulates multi-judge scoring as inference in a higher-rank latent-variable model and delivers three main contributions. (**i**) It explicitly models shared confounders, providing an aggregation scheme tailored to LLM-judge scenarios. (**ii**) It offers statistically principled estimators—sparse-plus-low-rank covariance recovery and tensor method—with provable identifiability. (**iii**) On three public benchmarks, CARE lowers MAE and raises Kendall's τ by up to 15%. Taken together, these advances enable principled, scalable, and low-cost evaluation pipelines for LLMs.

Limitations. Our theory assumes sufficient sparsity and approximate factor orthogonality; strong collinearity among latent variables, or latent components exhibiting similar spectral strengths may still hinder identifiability. In addition, selecting the "quality" factor currently relies on a simple loading-balance heuristic that can be unstable when confounders dominate, and our experiments are confined to English, text-only, scalar ratings—generalization to multilingual or multimodal settings remains future work.

Part III

A hierarchical co-clustering algorithm for bipartite graphs via spectral decomposition

14 INTRODUCTION

Hierarchical structures naturally emerge in numerous complex systems, ranging from biological classifications and organizational charts to document-term relationships and e-commerce user-item interactions. Traditionally, these hierarchical models have focused on unipartite settings—organizing entities of the same type into clusters or hierarchies. However, real-world scenarios frequently involve bipartite relationships, representing interactions between distinct sets of entities such as users and items, or authors and keywords.

Such bipartite structures can sometimes suggest underlying joint hierarchical organizations, which we refer to as *co-hierarchies*. In these co-hierarchies, each side potentially follows a tree-structured organization, and there might be mutual consistency between the hierarchies. Identifying and recovering these co-hierarchical structures from bipartite data has the potential to enhance interpretability, improve recommendation quality, and facilitate latent structure discovery. Nevertheless, existing hierarchical methods struggle with capturing these nuanced bipartite hierarchies due to intrinsic identifiability challenges.

Motivation for exploring co-hierarchical structures extends beyond purely bipartite data contexts. Consider the widely used biplot in Principal Component Analysis (PCA), which simultaneously visualizes observations and features. This joint representation potentially offers richer insights by concurrently analyzing both entity types involved.

Similarly, clustering approaches that incorporate nodes with distinct behavioral patterns can uncover more meaningful community structures. For instance, Rohe et al. (2012) explored clustering in directed graphs, explicitly acknowledging that nodes might exhibit different patterns in sending versus receiving connections. Their approach identified crucial asymmetries, highlighting the importance of jointly analyzing different

interaction behaviors within the same clustering framework. Although our work differs in focusing explicitly on interactions between two distinct sets of entities rather than directed interactions within a single set, the underlying motivation remains consistent: jointly modeling entities with fundamentally different interaction behaviors yields deeper structural insights.

Addressing this critical gap, we introduce a formal framework specifically tailored for modeling and recovering hierarchical structures in bipartite contexts. Our approach begins by formalizing the notation of a *co-hierarchy* for bipartite graphs and show its connection to bipartite Degree Corrected Stochastic Blockmodel (DCSBM). We then identify the ambiguity that arise when only cross-type relationship are observed, highlighting why naive tree-reconstruction procedures can fail. Finally, we identify a natural yet rich subclass-*perfect red-blue cherry trees*-for which, we design a simple yet effective recovery algorithm and prove both exact-recovery guarantees and robustness to perturbations.

Specifically, our main contributions are:

1. We introduce a general definition of co-hierarchies, relating hierarchical structures directly to bipartite degree-corrected stochastic block models (DCSBMs).

2. We define and study a practically motivated subclass, perfect red-blue cherry trees, which ensures strong identifiability.

3. We propose an efficient algorithm tailored to these perfect trees, accompanied by rigorous theoretical guarantees on exact recovery and perturbation robustness.

4. We provide practical diagnostics and illustrate our method's empirical effectiveness.

The remainder of this paper proceeds as follows: Section 15 reviews related literature. Section 16 introduces the co-hierarchy model and its connection to Degree Corrected Stochastic Block Model. In Section 17, we identify identifiability issues, define the canonical perfect cherry-tree representations, and present an algorithm to recover the canonical tree. Section 18 evaluates empirical performance on Newsgroup-20 dataset. Section 19 provides theoretical guarantee for our co-hierarchy estimation algorithm.

15 RELATED WORK

Our work bridges several distinct areas within hierarchical and bipartite data analysis:

Flat co-clustering and bipartite community detection. Classical co-clustering approaches, such as spectral methods and stochastic block models (SBMs), partition bipartite data into latent groups without explicit hierarchical structure. Seminal work by Dhillon (2001) introduced spectral co-clustering using bipartite graph partitioning, while subsequent approaches have improved computational efficiency and flexibility (Chakrabarti et al., 2004; Chen et al., 2023).

Unipartite hierarchical clustering methods. Hierarchical clustering algorithms like Ward's method, average linkage, and Neighbor-Joining (NJ) extensively model hierarchical structures in unipartite settings, with substantial theoretical guarantees available (e.g., Atteson's theorem on NJ). However, adapting these approaches directly to bipartite data is non-trivial due to distinct structural assumptions.

Hierarchical methods for bipartite data. Recent efforts have extended hierarchical clustering to bipartite scenarios. Initial efforts by Li and Li (2010) introduced hierarchical co-clustering, which was further developed to handle incremental updates (Pensa et al., 2014) and applied to various domains like music data organization (Li et al., 2012) and entity exploration in linked data (Zheng et al., 2018). However, existing hierarchical approaches focus primarily on algorithmic aspects and practical applications, lacking a rigorous model and theoretical guarantees for understanding when and how hierarchical structure can be recovered from bipartite data.

In summary, while substantial literature exists for hierarchical and bipartite analysis independently, no current method simultaneously recovers interpretable hierarchical structures from bipartite data with rigorous

theoretical guarantees. Our framework fills this gap by explicitly linking bipartite DCSBMs with hierarchical recovery under clear identifiability conditions, ensuring both interpretability and theoretical soundness.

16 DEGREE-CORRECTED STOCHASTIC BLOCK MODEL VIA LATENT TREE STRUCTURE

We begin by briefly reviewing the concept of Tree-Stochastic Graphs (TSGs) in the unipartite setting, as they provide foundational insights into the structure and identifiability of hierarchical models. Subsequently, we introduce our main focus: co-hierarchies for bipartite data.

16.1 Unipartite Background: $\mathbb{T}$-Stochastic Graphs (TSGs)

A unipartite $\mathbb{T}$-Stochastic Graph (TSG) is defined over a set of nodes structured hierarchically as leaves of a rooted tree. Each internal node corresponds to a cluster of leaf nodes, reflecting hierarchical groupings. A $\mathbb{T}$-Stochastic Graph generates a random graph represented by the adjacency matrix $A \in \mathbb{R}^{n\times n}$ where edge probabilities are determined by distances $d_{\mathbb{T}}$ in a latent hierarchy (i.e. a tree) $\mathbb{T}$:

$$\mathbb{E}(A_{ij}) := \lambda_{ij} = \exp(-d_{\mathbb{T}}(i,j)),$$

where λ_{ij} represents expected connecting probability between nodes i and j, $d_{\mathbb{T}}(i,j)$ represent the distance between nodes i and j in the tree $\mathbb{T}$.

TSGs directly correspond to Degree-Corrected Stochastic Block Models (DCSBMs), thereby connecting hierarchical clustering methods to probabilistic generative models. However, existing TSG frameworks primarily consider unipartite graphs, leaving bipartite scenarios unexplored.

16.2 Co-hierarchy: Bipartite $\mathbb{T}$-Stochastic Graphs

We now generalize the TSG concept to bipartite settings, introducing the notion of "co-hierarchy," represented by red-blue hierarchical trees. At a high level, to make a co-hierarchy, we are going to assign a color (red or blue) to every leaf node in $\mathbb{T}$. The two colors correspond to the two types of nodes in the bipartite graph.

Here is a more careful explanation. A unipartite graph $G = (V, E)$ has a node set V with n elements and an edge set E. A bipartite graph $G = (V_R, V_B, E)$ has two node sets V_R with n_1 elements and V_B with n_2 elements. In the incidence matrix $A \in \mathbb{R}^{n_1 \times n_2}$ for the bipartite graph G, the rows are indexed by the node set V_R and the columns are indexed by the node set V_B.

In the unipartite case, the leaf nodes of $\mathbb{T}$ are indexed by the node set V. In the bipartite case, the leaf nodes of $\mathbb{T}$ are indexed by the union of V_R and V_B. To discuss the differences we will add colors to the nodes in $\mathbb{T}$; make all of the V_R nodes *red* and make all of the V_B nodes *blue*. The color of the internal nodes is left ambiguous for now. We refer to the resulting graph as a red-blue tree.

Definition 16.2.1 (Co-hierarchy: Red-Blue Tree)**.** A *red-blue tree* $\mathbb{T}$ is a tree whose leaf nodes can be partitioned into two disjoint subsets of size n_1 and n_2:

$$\{r_1, r_2, \ldots, r_{n_1}\} \quad \text{(red leaf nodes)}, \quad \{b_1, b_2, \ldots, b_{n_2}\} \quad \text{(blue leaf nodes)}.$$

The red nodes correspond to the row nodes in the bipartite graph, while the blue nodes correspond to the column nodes.

To formally connect co-hierarchies with bipartite graphs, we define:

Definition 16.2.2 (Bipartite T-stochastic Graph)**.** A bipartite T-stochastic graph associated with red-blue tree $\mathbb{T}$ is a random bipartite graph $G = (V_R, V_B, E)$ with incidence matrix A, where the edge probability between $r_i \in V_R$ and $b_j \in V_B$ is:

$$P(A_{ij} = 1) = exp(-d_{\mathbb{T}}(r_i, b_j))$$

This model directly generalizes DCSBMs to hierarchical bipartite settings, maintaining interpretability while capturing richer structure through the hierarchical interactions. Moreover, co-hierarchies naturally address a critical challenge known as the "partial-distance" problem: observing connections only between distinct partitions (red-blue edges) obscures complete pairwise distances within each set, complicating direct hierarchical recovery.

In practice, we observe the incidence matrix A, and our goal is to estimate the underlying hierarchical tree structure $\mathbb{T}$. In certain cases, the entire hierarchical structure can be recovered exactly; in more general settings, however, inherent ambiguities may restrict us to partial reconstruction. Specifically, multiple distinct hierarchies can yield identical bipartite observations, making direct recovery challenging. In the following sections, we first clarify sources of ambiguity and then introduce a structured subclass of hierarchical models—*perfect red-blue cherry trees*—for which we develop a robust algorithm capable of exact hierarchical reconstruction.

17.1 Identifiability and Canonical Representation

As established in Section 16, a bipartite graph $G = (V_R, V_B, E)$ only captures edges between distinct vertex sets V_R and V_B. As a result, many different trees induce identical partial distance matrices, rendering them indistinguishable to any algorithm. We first separate those aspects of the trees that are intrinsically unidentifiable from those that can be recovered.

We proceed in two steps. First, we list several examples of ambiguity. Let $\mathbb{T}$ be an arbitrary red–blue tree and D its red–blue distance matrix. The following three phenomena leave D unchanged:

1. **Same-color sibling permutations.** Re-ordering a block of red (or blue) siblings under a common parent does not affect any red–blue path length.

2. **Linear chains of same-color nodes.** Compressing or expanding a path that alternates color once and then continues with the same color leaves all cross-type distances intact (Proposition 17.1.2).

3. **Twig-balance shift.** Adding a constant c to every blue twig while subtracting c from every red twig preserves every entry of D (Proposition 17.1.1).

Next, we enforce a canonical representation that removes these three ambiguity while leaving the observable distances untouched:

1. Sibling merge collapses same-color siblings into one representative leaf.

2. Path reduction eliminates redundant chains of identical color.

3. Twig balance fixes the free parameter in Proposition 17.1.1 by equalizing total red and blue twig length.

After these operations each equivalence class of trees maps to a single canonical tree. These transformations are necessary, omitting any of the three transformations leads to rank-deficient $\exp(-D)$ and provably misleads the reconstruction (see Figure 17.2 and the proof of Proposition 17.1.2).

Yet the canonical transformations we listed above are necessary but not sufficient for *uniqueness*. Different but observationally equivalent canonical trees can still share the same partial distances. In Section 4.2 we demonstrate this phenomenon and introduce additional assumptions—the *perfect red–blue cherry* condition—that removes the remaining ambiguity and yields a fully identifiable tree class.

Proposition 17.1.1 (Partial Distance Preservation Under Edge Length Transformation)**.** *Let $\mathbb{T}$ be a co-hierarchy with partial distance matrix D. For any constant $c \in \mathbb{R}$, define $\mathbb{T}_c$ by:*

1. *Increasing all blue twig lengths by c*
2. *Decreasing all red twig lengths by c*

Then $\mathbb{T}_c$ has the same partial distance matrix as $\mathbb{T}$.

Figure 17.1 illustrates this transformation, showing how modifying twig lengths in opposite directions preserves all observable distances between nodes of different colors. This invariance property motivates the development of our canonical form, which we now introduce.

Leveraging this invariance, we define the canonical form of a red-blue tree through three transformations:

- **Sibling Merge:** Merge siblings of the same color into a single node.

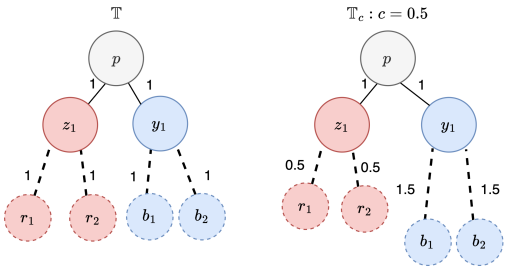


Figure 17.1: Edge length transformation preserves partial distances. The transformation adds constant c to blue twigs and subtracts c from red twigs, leaving all red-blue distances unchanged.

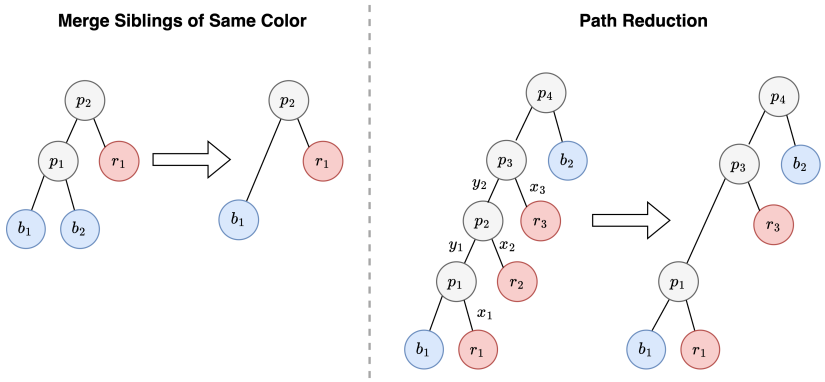


Figure 17.2: Illustration of the canonical mapping. Left: Tree with same-color siblings mapped to canonical tree. Right: Tree with consecutive merges with same color mapped to canonical tree. Edge lengths are used in proof for Proposition 17.1.2

.

- **Path Reduction:** Remove redundant consecutive merges involving nodes of the same color.
- **Twig Balance:** Adjust twig lengths to equalize the total lengths of red and blue twigs.

These operations eliminate structural redundancies and ambiguities, ensuring a full-rank exponential transformation $\exp(-D)$ (Proposition 17.1.2).

Figure 17.2 illustrates these operations.

The following proposition establishes their necessity:

Proposition 17.1.2 (Rank Preservation)**.** *The path reduction operation of the canonical mapping* (*operation* 2) *is necessary for* $\exp(-\mathrm{D})$ *to have full rank. Specifically, if a tree structure permits a path reduction operation but the operation is not performed, then* $\exp(-\mathrm{D})$ *is rank deficient.*

17.2 Perfect Red-Blue Cherry Trees are Canonical Trees

While the canonical map just defined removes ambiguities due to sibling permutations and chain symmetries, it alone does not guarantee a unique reconstruction: balanced "twig" subtrees can still swap without affecting pairwise distances. To eliminate this remaining freedom—and both simplify the model and enable a practical recovery algorithm-we further restrict our canonical representation to *perfect red-blue cherry trees*, characterized by each internal node at the lowest hierarchical level joining exactly one red and one blue leaf. This structure guarantees strong identifiability, simplifies recovery, and aligns naturally with the canonical form transformations described earlier.

Given the canonical form's structural clarity, we propose a straightforward recovery algorithm (Algorithm 5). The proposed algorithm has two steps, first we apply the Hungarian algorithm to optimally pair red and blue leaves, forming cherries, and calculate distances among parent nodes using these cherries. Next, we reconstruct the internal tree structure from these parent distances via Neighbor Joining (NJ). This two-step approach efficiently recovers the exact underlying tree structure in ideal conditions.

Algorithm 5 Red-Blue Tree Reconstruction

Input: Distance matrix $D \in \mathbb{R}^{k\times k}$ for leaves $R = \{r_1, \dots, r_k\}$ and $B = \{b_1, \dots, b_k\}$.

Output: Binary tree $\widehat{T}$ with k red–blue cherries.

1: Define the cost function

$$f(D, \Pi) = \sum_{i=1}^{k} D_{i,\pi(i)} = \langle D, \Pi \rangle,$$

for any permutation matrix $\Pi \in \mathcal{P}_k$. Determine the optimal assignment

$$\Pi^* = \arg\min_{\Pi \in \mathcal{P}_k} f(D, \Pi).$$

This yields the pairing $(r_i,\ b_{\pi^*(i)})$ for $i = 1, \dots, k$.

2: Define a parent p_i for each pair (r_i, b_i) and the inter-parent distances by

$$d(p_i, p_j) = \tfrac{1}{2}\Big[d(r_i, b_j) + d(r_j, b_i) - d(r_i, b_i) - d(r_j, b_j)\Big],$$

for $i, j = 1, \dots, k$, forming the matrix $D_p = [d(p_i, p_j)] \in \mathbb{R}^{k\times k}$.

3: Apply the Neighbor Joining algorithm to D_p to obtain a tree $\hat{T}_p$ on $\{p_i\}$.

4: Replace each p_i in $\hat{T}_p$ with its corresponding pair (r_i, b_i) to form $\widehat{T}$.

5: **Return:** $\widehat{T}$.

17.3 End-to-End Co-hierarchy Recovery Algorithm

Pipeline outline. Given the bipartite incidence matrix $A \in \mathbb{R}^{n_1\times n_2}$ (rows = red objects, columns = blue objects) and a target co-hierarchy size k, our recovery procedure proceeds in two stages:

1. *Block-structure estimation.* We first uncover the k lowest-level red and blue "cherries"—i.e. leaf pairs that share a common parent—together

with their interaction strengths. This is accomplished by the *Vintage Sparse PCA* subroutine (Algorithm 6), a varimax-rotated, sparsity-promoting SVD that simultaneously returns

$$\hat{Z},\ \hat{Y}\ \text{(soft leaf–to–cherry memberships)}, \quad \hat{B}\ \text{(red–blue interactions)}.$$

The factorization $\tilde{A} \approx \hat{Z}\hat{B}\hat{Y}^{\top}$ is provably identifiable under mild conditions.

2. *Tree reconstruction.* Treating each estimated cherry as a single composite leaf, we convert interaction weights to additive distances via $\hat{D} = -\log(\hat{B})$. On this $k \times k$ distance matrix we invoke a red-blue tree recovery algorithm tailored to red-blue inputs. The result is a fully specified red-blue tree $\hat{\mathbb{T}}$ whose leaves refine back to the original nodes by the memberships in $\hat{Z}, \hat{Y}$ (Algorithm 7).

Algorithms. Our method is summarized in Algorithms 6–7, providing a two-step, *end-to-end* solution that transforms raw incidence data into an interpretable and canonical red-blue co-hierarchy (Figure 17.3).

Algorithm 6 (Vintage Sparse PCA) identifies coherent red and blue communities (cherries) by estimating block membership matrices $\hat{Z}, \hat{Y}$ and a block interaction matrix $\hat{B}$. Algorithm 7 then reconstructs a co-hierarchy by processing the estimated block interactions into a red-blue tree and appending original vertices to their respective block nodes using $\hat{Z}, \hat{Y}$.

Complexity and Robustness. Algorithm 6 operates in $O(k \cdot \mathrm{nnz}(A))$ time, linear in the number of non-zero entries of A for fixed k. The final neighbor-joining (NJ) step requires $O(k^3)$ time. Section 19 provides detailed perturbation bounds, guaranteeing exact topology recovery under measurement noise of order $\tilde{O}(\eta/k)$, with η being the shortest internal edge of the ground-truth tree.

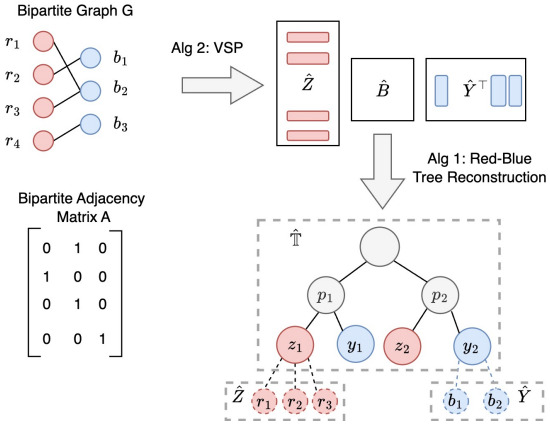


Figure 17.3: Two-step co-hierarchy recovery algorithm. Algorithm 6 estimates block memberships $(\hat{Z}, \hat{Y})$ and interactions $(\hat{B})$. Algorithm 5 then reconstructs a tree from block interactions, attaching original vertices accordingly.

Algorithm 6 Vintage Sparse PCA (vsp)

Input: Incidence matrix $A \in \mathbb{R}^{n_1 \times n_2}$, number of cherries k
Output: Memberships $\hat{Z}, \hat{Y}$, interaction matrix $\hat{B}$
Graph normalisation: $\tilde{A} \leftarrow D_R^{-1/2} A D_B^{-1/2}$
Compute rank-k SVD: $(U, V) \leftarrow \mathrm{SVD}(\tilde{A}, k)$
Varimax rotations: $(R_U, R_V) \leftarrow \mathrm{varimax}(U, V)$
$\hat{Z} \leftarrow \sqrt{n_1}\, U R_U, \quad \hat{Y} \leftarrow \sqrt{n_2}\, V R_V$
$\hat{B} \leftarrow \hat{Z}^\top \tilde{A} \hat{Y}$
return $\hat{Z}, \hat{Y}, \hat{B}$

Algorithm 7 End-to-End Co-hierarchy Recovery

Input: Incidence matrix A, number of cherries k
Output: Recovered co-hierarchy $\hat{\mathbb{T}}$
$(\hat{Z}, \hat{Y}, \hat{B}) \leftarrow \mathrm{vsp}(A, k)$ ▷ Algorithm 6
Convert weights to distances: $\hat{D} \leftarrow -\log(\hat{B})$
$\hat{\mathbb{T}} \leftarrow \mathtt{Algorithm5}(\hat{D})$ ▷ Red-Blue Tree Reconstruction Algorithm
return $\hat{\mathbb{T}}$

18 EXPERIMENT RESULTS

Setup. We ran our co-hierarchy estimator on the classic 20-Newsgroups collection. After standard English stop-word removal, lower-casing, and pruning words that appear in > 90 % or < 20 posts. The processed corpus contains 18846 documents and a vocabulary of 3694 unique terms. We interpret the TF–IDF-weighted document–term matrix as the adjacency matrix of a bipartite graph and apply the Vintage Sparse PCA + Red-Blue Neighbor-Joining pipeline (see Algorithm 7). Setting the latent dimension to $k = 10$ yields the consensus tree reproduced in Figure 18.1.

Result. Figure 18.1 shows the recovered consensus tree. Ten coherent meta-clusters emerge, grouping semantically related documents and their characteristic vocabulary. Table 18.1 summarizes each cluster by (i) its top terms, and (ii) a concise topic label that we assigned manually.

Documents and terms belonging to Clusters 3 & 9 are siblings in the tree,

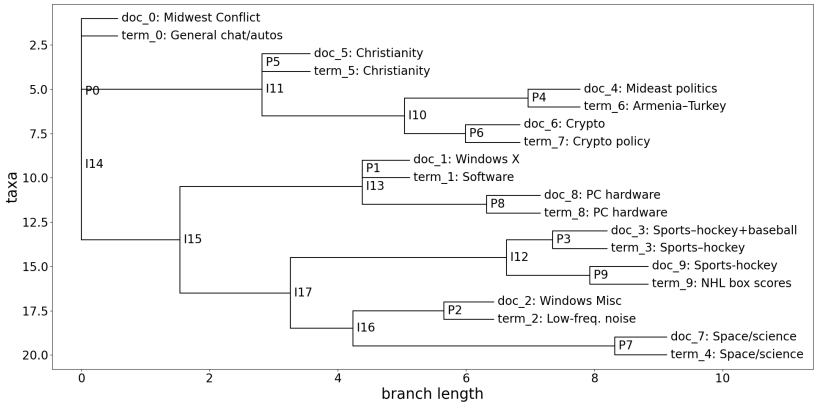


Figure 18.1: Co-hierarchy recovered from the 20-Newsgroups corpus. Each leaf node is labeled with a document (e.g., doc_0) or term (e.g., term_0), along with a manually assigned topic label (e.g., "Midwest Conflict", "Software"). Internal nodes represent cluster merges at varying branch lengths, capturing the semantic proximity between documents and terms.

	Topic (label)	Representative high-scoring terms (descending)
0	General chat / autos	*would, get, like, think, one, know, thing, people, car, time*
1	Software	*file, window, ftp, edu, server, program, version, display, image, application*
2	Low-freq. noise	*max, part, end, tad, hst, col, air, tie, wax, ahl*
3	Sports – hockey discourse	*game, team, player, play, season, league, hockey, chicago, win, detroit*
4	Space / science news	*space, research, science, nasa, mission, center, national, national, university, april, earth*
5	Christianity	*god, jesus, christian, christ, bible, church, faith, sin, belief, religion*
6	Armenian–Turkish conflict	*armenian, turkish, muslim, turk, armenia, genocide, village, turkey, argic, serdar*
7	Crypto	*key, encryption, clipper, chip, government, law, security, escrow, enforcement, secure*
8	PC hardware (SCSI/IDE)	*drive, card, scsi, disk, controller, ide, mhz, bus, floppy, ram*
9	NHL box-score abbreviations	*det, pit, bos, tor, chi, que, van, buf, nyi, stl*

Table 18.1: Recovered clusters (k = 10). Topic labels are added post-hoc for readability.

confirming the shared sports theme but finer lexical separation. Clusters 1 & 8 (software vs. hardware) form another subtree, reflecting the broader "computing" super-topic. The conversation-style Cluster 0 sits close to the root, acting as a linguistic "hub" with common functional words.

19.1 Exact Recovery Guarantee

Theorem 19.1.1 (Correctness of Red–Blue Cherries via Hungarian Algorithm)**.** *Applying the Hungarian algorithm guarantees correct identification of the lowest-level red-blue cherries.*

Proof. Assume for contradiction that the Hungarian algorithm picks the pairing (r_1, b_2) and (r_2, b_1) instead of true pairs (r_1, b_1) and (r_2, b_2). Let p_1, p_2 be internal parents for $(r_1, b_1), (r_2, b_2)$.

We note that

$$d(r_1, b_2) + d(r_2, b_1) = d(r_1, b_1) + d(r_2, b_2) + 2d(p_1, p_2).$$

Since $d(p_1, p_2) > 0$, we have:

$$d(r_1, b_2) + d(r_2, b_1) > d(r_1, b_1) + d(r_2, b_2).$$

This contradicts optimality, hence the Hungarian solution correctly identifies cherries. □

19.2 Perturbation Stability

Next, we characterize the perturbation stability of our tree recovery algorithm. In particular, we first analyze the sensitivity of the Hungarian matching step to perturbations in the distance matrix D. Combined with known perturbation results for Neighbor Joining (NJ), this analysis will allow us to bound the overall stability of the recovery procedure.

Sensitivity Analysis for Hungarian Matching Given a distance matrix D for two equal-sized sets R (red leaves) and B (blue leaves), the Hungarian

algorithm computes an optimal assignment

$$\Pi^* = \arg\min_{\Pi \in \mathcal{P}_k} f(D, \Pi), \quad \text{with} \quad f(D, \Pi) = \sum_{(r,b)\in\Pi} D_{r,b},$$

where $\mathcal{P}_k$ denotes the set of $k \times k$ permutation matrices.

To quantify robustness, we consider the allowable perturbation matrix Δ_D which that does not change optimal assignment plan:

$$\Pi^* = \arg\min_{\Pi \in \mathcal{P}_k} f(D, \Pi) = \arg\min_{\Pi \in \mathcal{P}_k} f(D + \Delta_D, \Pi).$$

In the next proposition, we characterize the allowable perturbation via a concept named *sensitivity* $s_{r,b}$, which is the minimal cost increase when reversing the assignment status of (r, b):

$$s_{r,b} = \begin{cases} f(D, \Pi_{r,b}) - f(D, \Pi^*), & \text{if } (r,b) \in \Pi^*, \\ f(D, \Pi^*) - f(D, \Pi_{r,b}), & \text{if } (r,b) \notin \Pi^*, \end{cases}$$

where $\Pi_{r,b}$ denotes the optimal assignment when the pairing (r, b) is forced to change its status.

Proposition 19.2.1 (Perturbation Stability of Hungarian Matching)**.** *Let* $k = |R| = |B|$. *Define the perturbation matrix*

$$\Delta_D = \left\{ \delta_{r,b} = \frac{s_{r,b}}{2k} \right\}_{(r,b)\in R\times B}.$$

Then, under the perturbed distance matrix $D + \Delta_D$, *the Hungarian algorithm recovers the original optimal assignment* Π^*.

Proof. To show that Π^* remains optimal, we must verify that for any alternative assignment $\Pi \neq \Pi^*$:

$$f(D + \Delta_D, \Pi^*) - f(D + \Delta_D, \Pi) \leqslant 0.$$

Equivalently, we need:

$$f(\Delta_D, \Pi^*) - f(\Delta_D, \Pi) \leqslant f(D, \Pi) - f(D, \Pi^*).$$

Assume that Π differs from Π^* in exactly t pairs (with $t \leqslant k$). Then the maximal cost advantage under the perturbation satisfies:

$$\sum_{(r,b)\in\Pi^*\setminus\Pi} \delta_{r,b} \leqslant 2t\frac{\max_{r,b}|s_{r,b}|}{2k} \leqslant \frac{2t}{2k}\Big(f(D,\Pi)-f(D,\Pi^*)\Big) \leqslant f(D,\Pi)-f(D,\Pi^*).$$

Since this extra cost is less than the original cost difference $f(D, \Pi) - f(D, \Pi^*)$, it follows that Π^* remains the optimal assignment under the perturbation. □

In our context, we can explicitly compute the sensitivity. We present it in the following proposition.

Proposition 19.2.2. *Suppose* (r_i, b_i) *are optimal assignments corresponding to parent nodes* p_i*. The sensitivity is explicitly computed as:*

$$s_{r_i,b_i} = 2d(p_i, p_j), \quad s_{r_i,b_j} = -2d(p_i, p_j).$$

Proof. If the optimal pairing (r_i, b_i) is removed from Π^*, the next-best pairing will involve the closest red-blue cherry with minimum distances between the parents, say (r_j, b_j). Incorporating any additional pair (r_m, b_m) only adds non-negative cost:

$$\begin{aligned}&\Big[d(r_i, b_j) + d(r_j, b_m) + d(r_m, b_i)\Big] - \Big[d(r_i, b_j) + d(r_j, b_i) + d(r_m, b_m)\Big]\\ &\quad = d(p_j, p_m) + d(p_m, p_i) - d(p_i, p_j) \geqslant 0.\end{aligned}$$

Thus, the minimal extra cost incurred by flipping the assignment is $2d(p_i, p_j)$ for an existing pair, and the analogous reasoning for a non-existent pair yields $-2d(p_i, p_j)$. □

Sensitivity Analysis for Tree Reconstruction Next, we combine the above perturbation analysis for the Hungarian algorithm with the perturbation result for the NJ algorithm to obtain an overall allowable perturbation bound on the partial distance matrix. This bound ensures that: (1) The Hungarian algorithm recovers the original optimal assignment; (2) The Neighbor Joining (NJ) algorithm recovers a tree that is topologically equivalent to the true tree.

Atteson [1997] showed that NJ is consistent if the pairwise distance matrix is perturbed by at most $\eta/2$, where η is the length of the shortest edge in the true tree. Combined with our earlier result on the Hungarian matching step, we obtain the following theorem.

Theorem 19.2.3. *Let* $\Delta_D = \{\delta_{r,b}\}_{(r,b)\in R\times B}$ *be a perturbation matrix such that:*

$$|\delta_{r_i,b_j}| \leqslant \min\left\{\frac{d(p_i,p_j)}{2k}, \frac{\eta}{4}\right\} \leqslant \min(\frac{1}{k}, \frac{1}{4})\eta,$$

for all $(r_i, b_j) \in R \times B$, *then, under the perturbed distance matrix* $D + \Delta_D$, *Algorithm 5 will recover a tree that is topologically equivalent.*

Proof. By our earlier analysis (see Proposition 19.2.1), the condition

$$|\delta_{r_i,b_j}| \leqslant \frac{d(p_i,p_j)}{2k}$$

ensures that the optimal assignment Π^* remains unchanged under the perturbed matrix $D + \Delta_D$. Meanwhile, Atteson [1997] demonstrated that the Neighbor Joining algorithm recovers the correct tree topology provided the error in the pairwise distances is at most $\eta/2$, where η is the shortest edge length in the true tree. In our context, the perturbation in the inter-parent distance satisfies

$$\delta(p_i,p_j) \leqslant \frac{1}{2}\Big(|\delta(r_i,b_i)| + |\delta(r_j,b_j)| + |\delta(r_i,b_j)| + |\delta(r_j,b_i)|\Big) \leqslant \frac{\eta}{2},$$

since each term is bounded by $\eta/4$. Therefore, both the Hungarian matching and the NJ steps are robust to the perturbation, ensuring that the original optimal assignment and the corresponding tree topology are preserved.

□

A APPENDIX FOR PART I

Supplementary materials contain additional experiment details, results and proofs. We provide glossary table in Section A.1, extended discussion on related work in Section A.2, additional experiment details in Section A.3 and results in Section A.4, technical lemmas and additional theoretical results in Section A.5, A.6, and proof in Section A.7.

A.1 Glossary

The glossary is given in Table A.1 below.

A.2 Extended Related Work

Hypothesis Test for Rotation-Sensitive Structure Early work on multivariate inference already framed "no preferred direction" as a null hypothesis and developed tests for departures from *spherical symmetry*. Classical procedures such as Mauchly's test for sphericity (Mauchly, 1940), John's test for identity covariance (John, 1971), and the Bingham–Watson family of uniformity tests on the hypersphere treat rotational invariance as the baseline state of a distribution; significant rejections therefore signal the presence of directional (i.e., rotation-sensitive) structure. In high-dimensional settings, modern random-matrix tests-e.g. the Ledoit–Wolf (Ledoit and Wolf, 2002) and Chen–Lei–Mao (Chen et al., 2010) statistics for detecting covariance spikes—extend the same principle by comparing observed eigen-spectra with isotropic nulls. Our approach inherits these ideas but adapts them to neural embeddings, where raw coordinates can be arbitrarily scaled or rotated. We deploy a rotation-invariant bootstrap that gauges whether an embedding layer departs from isotropy before we perform Varimax for interpretability.

Symbol	Definition
U	Matrix containing singular vectors from SVD decomposition
Z	Image loading matrix, represents how images load onto concepts
SO_k	Special orthogonal group (rotation matrices) in $\mathbb{R}^{k\times k}$
A	Input embedding matrix of size $\mathbb{R}^{n\times d}$
Y	Estimated concept matrix of size $\mathbb{R}^{d\times k}$
k	Number of concepts
n	Number of data points/images
d	Dimension of embeddings
R	Rotation matrix
T	Text embedding matrix
C_j	The j-th concept
$\sigma_d(Z)$	The absolute d-th largest singular value of Z
$\boldsymbol{C}^*$	Ground-truth latent concept matrix
$\boldsymbol{C}_W$	Word concept matrix
$\nu(U, R)$	Varimax objective function
$TS_1(U)$	Kurtosis test statistic
$TS_2(U)$	Varimax objective function test statistic
$TS_3(U)$	Rescaled kurtosis test statistic
$\|\cdot\|_F$	Frobenius norm
$\text{kurtosis}(U_{\cdot i})$	Kurtosis of the i-th column of U
$\mathcal{P}(k)$	Set of permutation matrices in $\mathbb{R}^{k\times k}$

Table A.1: Glossary of Notation

Rotation-sensitive Structure and Factor Interpretability Classical factor analysis has long recognized that raw factors are *rotation–indeterminate*: any orthogonal transform of the loading matrix yields the same likelihood, so interpretability hinges on choosing a "simple-structure" rotation such as Varimax (Kaiser, 1958). Recent statistical results show this is not just cosmetic — Varimax can be viewed as a consistent spectral estimator that recovers the true sparse structure under mild conditions (Rohe and Zeng, 2023). Independent-component analysis resolves the same indeterminacy by exploiting non-Gaussianity to make the rotation identifiable, demonstrating that probabilistic assumptions can anchor the axes in a

semantically meaningful way (Hyvärinen et al., 2023). In modern representation learning, the *same symmetry* re-appears: embeddings trained with contrastive or language-model objectives are identifiable only up to an unknown linear map, implying that any axis-aligned interpretation is fragile unless one fixes the rotation with additional bias (Roeder et al., 2021). Empirically, post-hoc rotations have been shown to sharpen semantics — e.g. Park et al. (2017) rotate word-embedding bases to make individual dimensions correspond to human concepts without hurting downstream accuracy. Our work unifies these threads: we provide a hypothesis test that *detects* rotation-sensitive structure in CLIP embeddings, then apply a statistically-grounded Varimax rotation to expose sparse, concept-aligned axes, thus reconciling fidelity with interpretability within a single framework.

A.3 Additional Experiment Details

All our experiment results are carried out using frozen pretrained weights from open-clip (ViT-B/32 and ViT-L/14), and no additional model training is involved.

Concept Decomposition Algorithm Details

Scaling Data Matrix We scaled data matrix $A \in \mathbb{R}^{n \times d}$ before applying SVD in the concept decomposition algorithm. Define the row normalization vector as:

$$deg_r = A\mathbf{1}_d \in \mathbb{R}^n, \quad \tau_r = \frac{1}{n}\mathbf{1}_n^T deg_r \in \mathbb{R}, \quad D_r = \mathrm{diag}(deg_r + \tau_r \mathbf{1}_n) \in \mathbb{R}^{n \times n}$$

Similarly, define the column quantities $deg_c = \mathbf{1}_n^T A \in \mathbb{R}^d$, $\tau_c = \frac{1}{d} deg_c \mathbf{1}_d \in \mathbb{R}$, and $D_c = \mathrm{diag}(deg_c + \tau_c \mathbf{1}_d) \in \mathbb{R}^{d \times d}$. The scaled data matrix is then defined as $\tilde{A} = D_r^{-1/2} A D_c^{-1/2}$, which we use as input to the concept decom-

position algorithm instead of the original matrix A. This scaling step with regularization parameters τ_r and τ_c helps stabilize the spectral estimation and prevents potential outliers in the singular vectors that could arise from noise in the data matrix (Le et al., 2017; Zhang and Rohe, 2018).

Hypothesis Test Experiment Details

To validate our hypothesis testing framework, we conducted experiments on both a real dataset—the ImageNet validation set—and two synthetic datasets: white-noise image embeddings and pure white-noise embeddings. The goal was to assess the framework's ability to detect non-random structures in different types of data.

Datasets.

- **ImageNet Validation Set:** We used embeddings computed by a pretrained Vision Transformer (ViT-B/32) model on images from the ImageNet validation set.
- **White-Noise Image Embeddings:** We generated 10,000 white-noise images, each of size 224×224 pixels with 3 color channels. Each pixel value was drawn independently from a standard Gaussian distribution. These images were then processed by the pretrained ViT model to obtain embeddings of size $10,000 \times 512$.
- **Pure White-Noise Embeddings:** We directly generated a random noise matrix of dimensions $10,000 \times 512$, with each entry sampled from a standard Gaussian distribution, without passing through the embedding model.

Experiment Details. For each dataset, we obtained an embedding matrix $\tilde{A}$ as described above. We then performed Singular Value Decomposition (SVD) on $\tilde{A}$, decomposing it into $\tilde{A} = UDV^\top$. Here, U is a matrix whose

columns are the left singular vectors, representing orthogonal directions in the embedding space, and whose rows correspond to the images.

We observed that the first column of U (the first principal component) often captured mean or bias effects in the embeddings. The loadings on this component were concentrated around a constant value, offering limited information about the latent structure of the data. Therefore, we excluded the first column of U, defining $\tilde{U} = U[:, 2:]$, to focus on more informative components.

Next, we applied our hypothesis testing framework to $\tilde{U}$ to compute p-values and test statistics, assessing the statistical significance of any non-random patterns present in the data.

Randomness in the Procedure

Our hypothesis testing procedure involves randomness in two key aspects:

1. **Row-wise Random Rotations:** To generate conditionally rotation-invariant data, we applied random rotations to each row of $\tilde{U}$. This step introduces randomness in the transformation of the data.

2. **Generation of Synthetic Data:** The white-noise image and pure white-noise embeddings were generated using random sampling from standard Gaussian distributions.

To account for the variability introduced by these random processes, we performed additional tests using 5 different random seeds and varied the rank k of $\tilde{U}$.

Selection of Rank k We investigated how the choice of rank k, the number of singular vectors retained in $\tilde{U}$, affects the results of the hypothesis tests. We expect the p-values to increase with larger k, indicating a decreased ability to detect rotation-sensitive structure. As k increases, more columns of U are included, potentially introducing additional noise and reducing

the statistical power to detect non-Gaussian signals. We report our results in Figure A.1. We observed that for white noise image embeddings, p-values increase as k increases, which aligns with our expectation. For white noise embedding, we observed p-values are oscillating around 0.5 and show no clear pattern as k changes. This aligns with our theoretical results from Example 4, which suggests U follows a rotationally invariant distribution, and p-value should follow an approximately uniform distribution between 0 and 1.

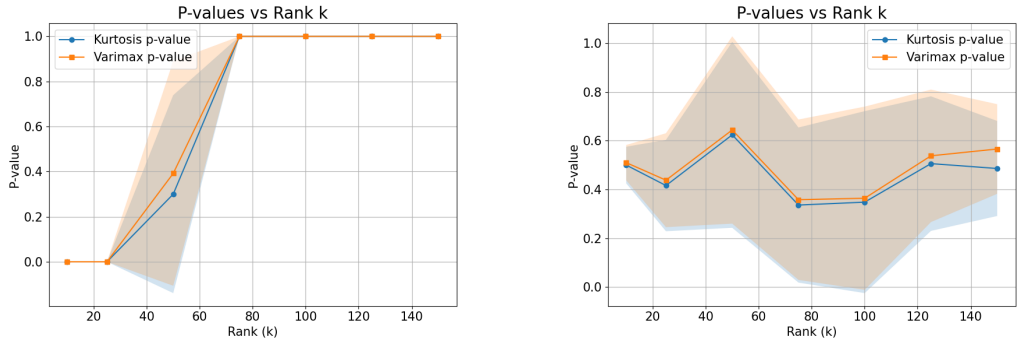


Figure A.1: Illustration of how p-values change with rank k. Left: white noise image embedding from pretrained ViT-L/14 model. Right: white noise embedding of dimension $10,000 \times 768$.

Spurious Concept Removal Experiment

Datasets details

- **Waterbirds** (Sagawa et al., 2019): Bird species (waterbird/landbird) with spurious background correlations (water/land)
- **WILDS-iWildCam** (Beery et al., 2020; Koh et al., 2021): Animal species classification with spurious location-specific features across camera traps

Dataset	Task	Spurious Attribute	#Images	#Classes
Waterbirds	Bird Species	Background	4,795	2
iWildCam	Animal Species	Location	42,791	182
CelebA	Hair Color	Eyeglasses	19,962	2

Table A.2: Dataset characteristics and their corresponding spurious correlations.

- **CelebA** (Liu et al., 2015): Hair color classification (blonde/non-blonde) with spurious attribute correlation (eyeglasses)

Dataset statistics can be found in table A.2.

Spurious Concept Detection Methods. We develop strategies to automatically identify spurious concepts, tailored to each dataset's characteristics:

- For **Waterbirds**, we generate text descriptions following the template "A {bird_type} with a {background_type} background." We identify spurious concepts as those where top-ranking descriptions share common backgrounds but varied bird species.
- For **iWildCam**, we employ a contrastive approach using two sets of descriptions: one focusing on animal features and another on location attributes. We compute cosine similarities between concept embeddings and these description embeddings to identify concepts that correlate strongly with location features.
- For **CelebA**, we generate descriptions emphasizing either hair color or eyeglasses attributes, using a similar contrastive approach to separate target concepts from spurious ones.

Removing Spurious Concepts To remove spurious concepts, we reconstruct image embeddings while setting the coefficients of identified spurious concepts to zero. Given the decomposition $A_{i.} = \sum_j \alpha_j C_j$ where C_j

are learned concept vectors, we enforce $\alpha_{\mathrm{spurious}} = 0$ to filter out spurious information.

Waterbirds Experiment For Waterbirds experiment, we use ['a landbird', 'a waterbird'] as class prompts. In the zero-shot prediction experiment, we first compute text embeddings for class prompts, and compute cosine similarity between class prompt embeddings and image embeddings. Then for each image, we extract the class with higher similarity as the prediction.

For removing spurious concepts, we first decompose image embedding into a linear combination of concepts with Algorithm 2: $A_{i\cdot} = \sum_j \alpha_j C_j$. Suppose we have identified spurious concepts with our proposed method as explained in main content, by removing spurious concepts we set the coefficients for spurious concepts to 0. In other words, $\alpha_{\mathrm{spurious}} = 0$.

Algorithm to Generate Rotation-Invariant Matrix

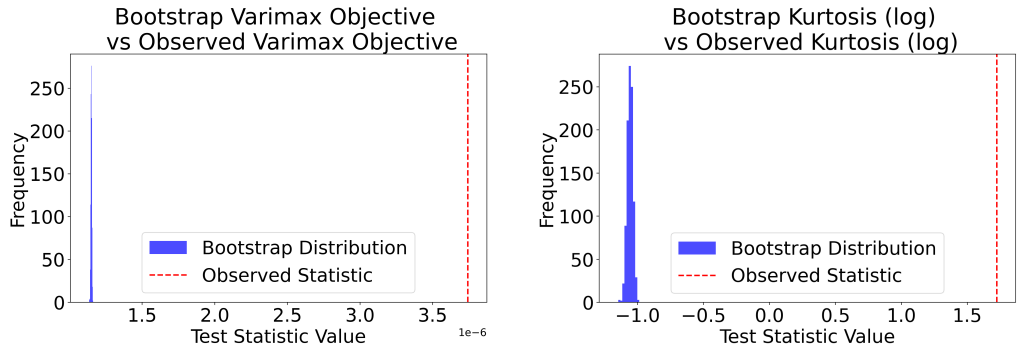
Algorithm 8 Generate Rotation-Invariant Matrix

Input: Matrix $U \in \mathbb{R}^{n \times k}$
Output: Rotation-invariant matrix $U^{\mathrm{rot}} \in \mathbb{R}^{n \times k}$
for each row $u_i \in \mathbb{R}^k$, $i = 1, 2, \ldots, n$ **do**
    Generate random rotation matrix $R_i \in \mathbb{R}^{k \times k}$
    $u_i^{\mathrm{rot}} \leftarrow R_i u_i$ ▷ Rotate the row u_i
end for
$U^{\mathrm{rot}} \leftarrow [u_1^{\mathrm{rot}}, u_2^{\mathrm{rot}}, \ldots, u_n^{\mathrm{rot}}]^\top$ ▷ Matrix of rotated rows
return U^{rot}

A.4 Additional Experiment Results

Bootstrap simulation results

In Figure A.2, we present bootstrap kurtosis and observed kurtosis distribution defined in Section 2.3.



(a) Bootstrap Varimax Objective vs Observed

(b) Bootstrap Kurtosis vs Observed Kurtosis

Figure A.2: Comparison of bootstrap distributions and observed test statistics. The blue histograms show the distribution of test statistics computed from rotation-invariant resamples under the null hypothesis. The red dashed lines indicate the observed test statistics computed from CLIP embeddings of ImageNet validation set.

Additional Concept Results for ImageNet

We provide additional concept results for ImageNet validation set in Figure A.3. Embeddings are computed by ViT-B-32 model.

Concept Results for Waterbirds

We provide concept results for Waterbirds dataset in Figure A.4.

Additional experiment results

Concept Learns Analogical Relations

Word embeddings are known to capture semantic relationships through vector arithmetic, famously demonstrated by analogies such as "king - man + woman = queen" (Mikolov et al., 2013). We demonstrate that our learned concepts exhibit similar compositional properties with visual concepts.

Figure A.3: Top-24 concepts using our method with leading images and corresponding text descriptions. We observe image and text concepts are well-aligned with similar semantic topics.

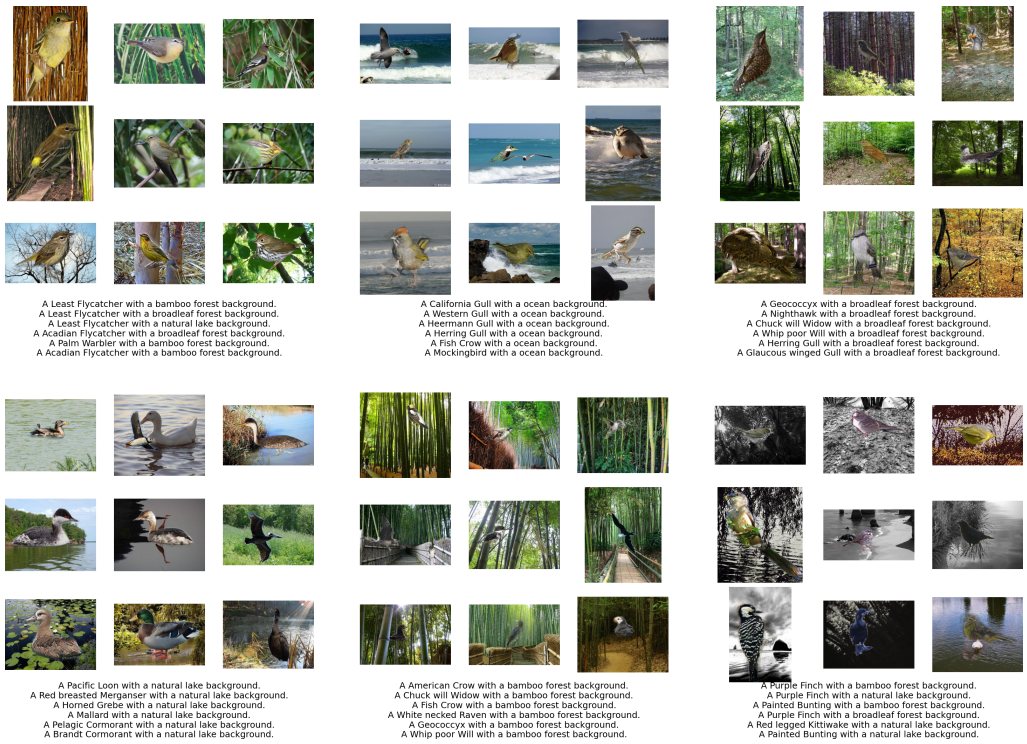


Figure A.4: Top-6 waterbirds concepts with text descriptions. We noticed there are bird-focused concepts (e.g. first row, left column) that specify the species more clearly and mention distinctive features. There are background-focused concepts (e.g. first row, middle column), that highlight the type of environment. We also observed a multiple birds concept (second row, left column).

To evaluate this, we identify three key concepts from our learned representation: C_{gd} (representing groups of dogs), C_d (single dog), and C_b (bird). We then construct a new concept through vector arithmetic: $C = C_{gd} - C_d + C_b$. Intuitively, this operation should capture the transformation from "single entity" to "group" and apply it to birds. We evaluate this constructed concept in two ways: by projecting image embeddings ($Score = AC$ where A contains image embeddings) and text embeddings onto this concept space. As shown in Figure A.5, both the top-scoring images and their associated text descriptions align with our expectation,

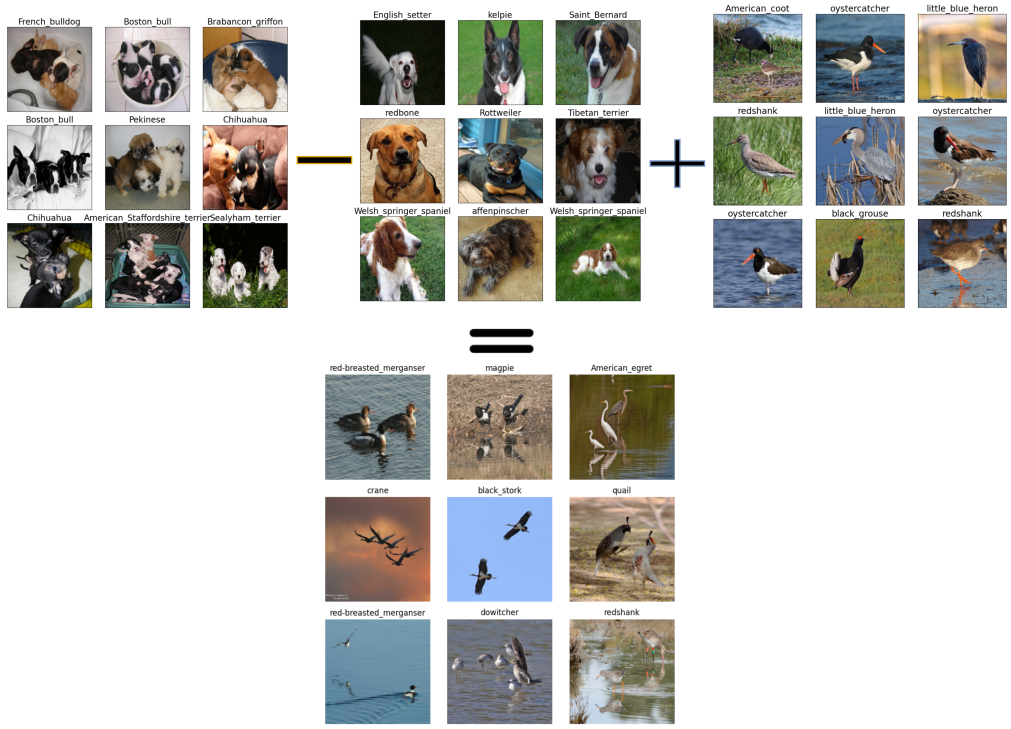


Figure A.5: Demonstration of analogical reasoning with concepts. The equation C_{gd} (group of dogs) $- C_d$ (single dog) $+ C_b$ (single bird) yields a concept that correctly identifies groups of birds in both image and text spaces.

consistently returning groups of birds, demonstrating that our method successfully captures and transfers the concept of collectiveness across different semantic categories.

A.5 Technical Lemmas

In this section, we provide some technical results for convenience.

The following lemma is a generalization to (Li et al., 2023, Lemma H.5) for a non-square matrix. We recall $\sigma_k(Z) = \sqrt{\sigma_k(Z^\top Z)}$ as the k-th largest absolute singular value of Z.

Lemma A.5.1. *For* $\mathbf{A} \in \mathbb{R}^{d\times k}$, $\mathbf{B} \in \mathbb{R}^{k\times n}$ *where* $n > d \geqslant k$, *we have*

$$\|\mathbf{A}\|_F \cdot \sigma_k(\mathbf{B}) \leqslant \|\mathbf{AB}\|_F \leqslant \|\mathbf{A}\|_F \cdot \sigma_1(\mathbf{B}). \tag{A.1}$$

Proof. Assume the SVD of $\mathbf{B}$ is $U_{\mathbf{B}}D_{\mathbf{B}}V_{\mathbf{B}}$. Then,

$$\|\mathbf{AB}\|_F = \|\mathbf{A}U_{\mathbf{B}}D_{\mathbf{B}}V_{\mathbf{B}}\|_F = \|\mathbf{A}U_{\mathbf{B}}D_{\mathbf{B}}\|_F. \tag{A.2}$$

By applying a similar induction proof as in (Li et al., 2023, Lemma H.5) to $\mathbf{A}U_{\mathbf{B}}D_{\mathbf{B}}$ where $\mathbf{A}U_{\mathbf{B}} \in \mathbb{R}^{d\times k}$ and $D_{\mathbf{B}} \in \mathbb{R}^{k\times k}$, we obtain

$$\|\mathbf{A}U_{\mathbf{B}}D_{\mathbf{B}}\|_F \geqslant \|\mathbf{A}U_{\mathbf{B}}\|_F \cdot \sigma_k(D_{\mathbf{B}}) = \|\mathbf{A}\|_F \cdot \sigma_k(D_{\mathbf{B}}) \tag{A.3}$$

and

$$\|\mathbf{A}U_{\mathbf{B}}D_{\mathbf{B}}\|_F \leqslant \|\mathbf{A}U_{\mathbf{B}}\|_F \cdot \sigma_1(D_{\mathbf{B}}) = \|\mathbf{A}\|_F \cdot \sigma_1(D_{\mathbf{B}}). \tag{A.4}$$

This concludes the proof. □

A.6 Additional theoretical results

Example 4. Let $A \in \mathbb{R}^{n\times m}$ with A_{ij} i.i.d. generated from a Gaussian mixture model $\frac{1}{2}N(1,1) + \frac{1}{2}N(-1,1)$. When $\min\{n,k\} > 2$, the left and right singular vector matrices of A are rotation-sensitive.

A.7 Missing proofs

Proof of Example 1

Proof. The density of the standard multivariate normal distribution is given by

$$f(x) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|x\|^2\right),$$

which depends only on the L_2 norm $\|x\|$. For any rotation matrix R, we have $\|xR\| = \|x\|$ because rotations preserve norms. Therefore,

$$f(xR) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|xR\|^2\right) = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|x\|^2\right) = f(x).$$

This shows that the density (and thus the distribution) of x is unchanged under rotations, proving rotational invariance. □

Proof of Example 2

Proof. It suffices to show the result when $k = \min\{n, m\}$. Due to the nature of normally distributed random variables, for any orthogonal matrices $G \in \mathbb{R}^{m\times m}$ and $H \in \mathbb{R}^{n\times n}$, the entries of GAH are i.i.d. and follow a standard normal distribution. Therefore, this guarantees spherical symmetry for the left and right singular matrices U and $V^\top$ of A, implying that both U and $V^\top$ follow a uniform distribution with respect to the Haar measure on the Stiefel manifold.

For any rotation matrix $R \in \mathbb{R}^{k\times k}$, we have $(UR)^\top UR = \mathcal{I}_k$. Consider $f_R(U) = UR$, which defines a one-to-one map from the Stiefel manifold to itself. By the one-to-one property, UR has the same distribution as U, namely following the uniform distribution to the Haar measure on the Stiefel manifold. A similar result holds for $V^\top$.

This guarantees rotation invariance. □

Proof of Example 3

Proof. To show that this distribution is not rotationally invariant, we need to find a rotation matrix R such that the distribution of xR differs from the distribution of x.

Let R be a rotation matrix that rotates the vector μ to another direction. For simplicity, consider a rotation R that maps μ to $R\mu = \nu$, where $\nu =$

$(0, 1, 0, \ldots, 0)^\top$.

The original distribution of x has two components centered at μ and $-\mu$. After rotation, the distribution of xR has components centered at $R\mu = \nu$ and $R(-\mu) = -\nu$.

However, the probability density function (pdf) of x before rotation is

$$f(x) = \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|x-\mu\|^2\right) + \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|x+\mu\|^2\right).$$

After rotation, the pdf becomes

$$f_R(x) = f(xR) = \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|xR-\mu\|^2\right) + \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|xR+\mu\|^2\right).$$

But since $\|xR-\mu\|^2 \neq \|x-\mu\|^2$ in general, the pdf $f_R(x)$ is not equal to $f(x)$. Specifically, the locations of the mixture components have changed, leading to a different distribution.

Moreover, consider evaluating the probability at a specific point. For example, at $x = \mu$, we have

$$\begin{aligned} f(\mu) &= \frac{1}{2(2\pi)^{d/2}} \exp\Big(-\tfrac{1}{2}\|\mu-\mu\|^2\Big) + \frac{1}{2(2\pi)^{d/2}} \exp\Big(-\tfrac{1}{2}\|\mu+\mu\|^2\Big) \\ &= \frac{1}{2(2\pi)^{d/2}}\Big[1 + \exp\big(-2\|\mu\|^2\big)\Big]. \end{aligned}$$

After rotation, at $x = \mu$, we have

$$f_R(\mu) = f(\mu R) = \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|\mu R-\mu\|^2\right) + \frac{1}{2}\frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|\mu R+\mu\|^2\right).$$

Since $\mu R \neq \mu$, the values of $f(\mu)$ and $f_R(\mu)$ are different, confirming that the distribution is not rotationally invariant. □

Proof of Example 4

Proof. We present the proof that the right singular vector matrix of A is rotation-sensitive, and the proof of the left singular vector matrix follows similarly. It suffices to show the probability density function on two right singular vectors in $\mathbb{R}^k$ are different.

Consider $v = (1, 0, 0, \ldots, 0)$ and $v' = (\frac{1}{\sqrt{k}}, \frac{1}{\sqrt{k}}, \ldots, \frac{1}{\sqrt{k}})$. Then, $(Av)_i \sim \frac{1}{2}N(1,1) + \frac{1}{2}N(-1,1)$ for $i = 1, \ldots, n$, and $(Av')_i$ i.i.d. follows a Binomial distribution of Gaussian mixture model. When $\min\{n, k\} > 2$, the probability density function on these two vectors is different because the variance of $(Av)_i$ equals 1, and the variance of $(Av')_i$ is apparently smaller than 1. This completes the proof. □

Proof of Proposition 2.2.1

Under H_0, to show that A^{rot} is rotation invariant, we need to prove that for any fixed rotation matrix $R \in SO(d)$, the distribution of $A^{rot}R$ is the same as that of A^{rot}.

Since conditional on the same $\|A_i\|_2 = \|A_i^{rot}\|_2$, for any A_i^{rot} there must exist R_i such that

$$A_i^{rot} = A_i R_i,$$

where R_i uniformly sampled from $SO(d)$. For any $R \in SO(d)$, multiplying A^{rot} on the right by R:

$$A_i^{rot} R = (A_i R_i) R = A_i (R_i R) = A_i \tilde{R}_i,$$

where we define $\tilde{R}_i = R_i R$. Since R_i are uniformly distributed over $SO(d)$ and independent, and R is a fixed element of $SO(d)$, the products $\tilde{R}_i = R_i R$ are also uniformly distributed over $SO(d)$, independent from each other, and independent from A_i. Therefore, the distribution of $A_i \tilde{R}_i$ is the same

as that of $A_i R_i$:

$$A_i \tilde{R}_i \overset{d}{=} A_i R_i.$$

This implies that the rows of $A^{rot}R$ have the same joint distribution as the rows of A^{rot}:

$$\{A_i^{rot} R\}_{i=1}^n \overset{d}{=} \{A_i^{rot}\}_{i=1}^n.$$

This completes the proof.

Proof of Theorem 2.3.1

For theoretical analysis, we derive an equivalent standardized form:

$$TS_3(U) = \frac{\sqrt{nk}}{\sqrt{33}} \left(\frac{1}{k} \sum_{i=1}^{k} |\text{kurtosis}(U_{\cdot i})| - \frac{3n}{n+2} \right). \tag{A.5}$$

Recall that $U^\top U = \mathbf{I}$, therefore we have $\sum_{j=1}^n U_{ji}^2 = 1$ for any i. We can simplify the rescaled kurtosis as

$$TS_3(U_{\cdot i}) = \frac{\sqrt{nk}}{\sqrt{33}} \left(\frac{n}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} U_{ji}^4 - \frac{3n}{n+2} \right). \tag{A.6}$$

We recall from the proof of Example 2 that for fixed i, $U_{\cdot i}$ follows a normal distribution on Haar measure. Denote

$$X_i = n \sum_{j=1}^{n} U_{ji}^4 - \frac{3n}{n+2}.$$

We compute

$$\mathbb{E}[U_{ij}^{2s}] = \frac{\Gamma(\frac{n}{2})\Gamma(s+\frac{1}{2})}{\Gamma(\frac{n}{2}+s)\Gamma(\frac{1}{2})}.$$

When $s = 2$, we have

$$\mathbb{E}[U_{ij}^4] = \frac{3}{n(n+2)}.$$

Therefore, by plugging this formula in our computation, we obtain

$$\mathbb{E}[X_i] = \frac{3n}{n+2} - \frac{3n}{n+2} = 0,$$

and

$$\mathbb{E}[U_{ij}^8] = \frac{\frac{7}{2}\frac{5}{2}\frac{3}{2}\frac{1}{2}}{(\frac{n}{2}+3)(\frac{n}{2}+2)(\frac{n}{2}+1)\frac{n}{2}}.$$

On the other hand, by rewriting U_{ij} as $\frac{S_i}{\sqrt{\sum_i S_i^2}}$, we obtain $(U_{ij}, U_{i'j})$ and $(\frac{U_{ij}+U_{i'j}}{\sqrt{2}}, \frac{U_{ij}-U_{i'j}}{\sqrt{2}})$ are identically distributed. Therefore, we have

$$\mathbb{E}\left[U_{ij}^4 U_{i'j}^4\right] = \mathbb{E}\left[\left(\frac{U_{ij}+U_{i'j}}{\sqrt{2}}\right)^4 \left(\frac{U_{ij}-U_{i'j}}{\sqrt{2}}\right)^4\right].$$

This can be reduced to

$$\mathbb{E}\left[U_{ij}^8\right] = 4\mathbb{E}\left[U_{ij}^6 U_{i'j}^2\right] + 5\mathbb{E}\left[U_{ij}^4 U_{i'j}^4\right]. \tag{A.7}$$

On the other hand, we have

$$\mathbb{E}\left[U_{i'j}^6 U_{i'j}^2\right] = \frac{1}{n-1}\left(\mathbb{E}\left[U_{i'j}^6\right] - \mathbb{E}\left[U_{i'j}^6\right]\right) = \frac{15}{(n+6)(n+4)(n+2)n}. \tag{A.8}$$

Combining equation A.7 and equation A.8, we obtain,

$$\mathbb{E}\left[U_{ij}^4 U_{i'j}^4\right] = \frac{9}{(n+6)(n+4)(n+2)n}.$$

By plugging the computations of moments, we obtain

$$\begin{aligned}\mathbb{E}\left[X_i^2\right] &= n^3\mathbb{E}[U_{ij}^8] + n^3(n-1)\mathbb{E}[U_{ij}^4 U_{i'j}^4] - n^3(n-1)(\mathbb{E}[U_{ij}^4])^2 \\ &= \frac{105 \times n^2}{(n+6)(n+4)(n+2)} + \frac{9n^2(n-1)}{(n+6)(n+4)(n+2)} - \frac{9n(n-1)}{(n+2)^2}.\end{aligned}$$

The leading order term of this variance is $\frac{33}{n}$. By the central limit theorem

and the fact that X_i^2 are i.i.d. random variables, we conclude the result.

Proof of Theorem 4.1.2

Recall Assumption 4.1.1 that $\mathbb{E}[\tilde{Z}_{ij}] = 0$, $\mathbb{E}(\tilde{Z}_{ij}^2) = \sigma_j^2$, $\mathbb{E}(\tilde{Z}_{ij}^4) = \eta_j \geqslant 3\sigma_j^4$.

$$\nu(R, \tilde{Z}\tilde{R}^\top) = \frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\left([\tilde{Z}\tilde{R}R]_{i\ell}^4 - \left(\frac{1}{n}\sum_{q=1}^{n}[\tilde{Z}\tilde{R}R]_{q\ell}^2\right)^2\right).$$

To simplify notation, we denote $O = \tilde{R}R \in \mathcal{O}(k)$. We want to optimize $\nu(R, \tilde{Z}^\top\tilde{R}^\top)$ over O. We analyze two terms, respectively. For the fourth moment term

$$\begin{aligned}
\mathbb{E}\left(\frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}[\tilde{Z}O]_{i\ell}^4\right) &= \mathbb{E}\left(\frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\left(\sum_{j=1}^{k}\tilde{Z}_{ij}O_{j\ell}\right)^4\right) \\
&= \mathbb{E}\left(\frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\tilde{Z}_{ij}^4O_{j\ell}^4 + 3\sum_{h\neq h'}\tilde{Z}_{ih}^2\tilde{Z}_{ih'}^2O_{hl}^2O_{h'l}^2\right)\right) \\
&= \frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\left(\sum_{j=1}^{k}\eta_jO_{j\ell}^4 + 3\sum_{h\neq h'}\sigma_h^2\sigma_h'^2O_{hl}^2O_{h'l}^2\right) \\
&= \sum_{\ell=1}^{k}\left(\sum_{j=1}^{k}\eta_jO_{j\ell}^4 + 3\sum_{h\neq h'}\sigma_h^2\sigma_h'^2O_{hl}^2O_{h'l}^2\right),
\end{aligned}$$

because the expectation of all other cross terms in the computation contains at least one moment of an entry, which is 0 according to the independence

and $\mathbb{E}[\tilde{Z}_{ij}] = 0$. For the second moment term,

$$\begin{aligned}
\mathbb{E}\Big[\frac{1}{n}\sum_{\ell=1}^{k}\sum_{i=1}^{n}\big(\tfrac{1}{n}\sum_{q=1}^{n}[\tilde{Z}O]_{q\ell}^2\big)^2\Big] &= \frac{1}{n^2}\sum_{\ell=1}^{k}\Big(\sum_{q=1}^{n}\big(\sum_{j=1}^{k}\tilde{Z}_{qj}O_{j\ell}\big)^2\Big)^2 \\
&= \mathbb{E}\Big[\frac{1}{n^2}\sum_{\ell=1}^{k}\Big(\sum_{q=1}^{n}\sum_{j=1}^{k}\tilde{Z}_{qj}^2O_{j\ell}^2 + \sum_{q=1}^{n}\sum_{h\neq h'}\tilde{Z}_{qh}\tilde{Z}_{qh'}O_{h\ell}O_{h'\ell}\Big)^2\Big] \\
&= \frac{1}{n^2}\text{cross terms} + \underbrace{\mathbb{E}\Big[\frac{1}{n^2}\sum_{\ell=1}^{k}\big(\sum_{q=1}^{n}\sum_{j=1}^{k}\tilde{Z}_{qj}^2O_{j\ell}^2\big)^2\Big]}_{\text{Term 1}} \\
&\quad + \underbrace{\mathbb{E}\Big[\frac{1}{n^2}\sum_{\ell=1}^{k}\big(\sum_{q=1}^{n}\sum_{h\neq h'}\tilde{Z}_{qh}\tilde{Z}_{qh'}O_{h\ell}O_{h'\ell}\big)^2\Big]}_{\text{Term 2}}.
\end{aligned}$$

since $\mathbb{E}(\tilde{Z}_{qj}) = 0$, the expectation of all the cross terms should be 0 and can be removed from the equation. Now we compute terms 1 and 2,

respectively. For term 1,

$$\begin{aligned}
\mathbb{E}\left(\text{Term 1}\right) &= \frac{1}{n^2}\sum_{\ell=1}^{k}\mathbb{E}\left(\sum_{q=1}^{n}\sum_{j=1}^{k}\tilde{Z}_{qj}^2 O_{j\ell}^2\right)^2 \\
&= \frac{1}{n^2}\sum_{\ell=1}^{k}\left(\text{Var}\left(\sum_{q=1}^{n}\sum_{j=1}^{k}\tilde{Z}_{qj}^2 O_{j\ell}^2\right) + \left(\mathbb{E}\left[\sum_{q=1}^{n}\sum_{j=1}^{k}\tilde{Z}_{qj}^2 O_{j\ell}^2\right]\right)^2\right) \\
&= \frac{1}{n^2}\sum_{\ell=1}^{k}\left(\sum_{q=1}^{n}\sum_{j=1}^{k}O_{j\ell}^4\text{Var}\left(\tilde{Z}_{qj}^2\right) + \left(\sum_{q=1}^{n}\sum_{j=1}^{k}O_{j\ell}^2\mathbb{E}(\tilde{Z}_{qj}^2)\right)^2\right) \\
&= \sum_{\ell=1}^{k}\left(\frac{1}{n}\sum_{j=1}^{k}O_{j\ell}^4(\eta_j - \sigma_j^4) + \left(\sum_{j=1}^{k}O_{j\ell}^2\sigma_j^2\right)^2\right) \\
&= \sum_{\ell=1}^{k}\left(\sum_{j=1}^{k}O_{j\ell}^2\sigma_j^2\right)^2 + \frac{1}{n}\sum_{\ell=1}^{k}\sum_{j=1}^{k}(\eta_j - \sigma_j^4)O_{j\ell}^4.
\end{aligned}$$

For term 2,

$$\begin{aligned}
\mathbb{E}\left(\text{Term 2}\right) &= \frac{1}{n^2}\sum_{\ell=1}^{k}\mathbb{E}\left(\sum_{q=1}^{n}\sum_{h\neq h'}\tilde{Z}_{qh}\tilde{Z}_{qh'}O_{h\ell}O_{h'\ell}\right)^2 \\
&= \frac{2}{n^2}\sum_{\ell=1}^{k}\sum_{q=1}^{n}\sum_{h\neq h'}\left(\mathbb{E}(\tilde{Z}_{qh}^2)\mathbb{E}(\tilde{Z}_{qh'}^2)O_{h\ell}^2O_{h'\ell}^2\right) \\
&= \frac{2}{n}\sum_{\ell=1}^{k}\sum_{h\neq h'}\left(\sigma_h^2\sigma_{h'}^2O_{h\ell}^2O_{h'\ell}^2\right) \\
&= \frac{2}{n}\sum_{\ell=1}^{k}\left(\left(\sum_{h=1}^{k}\sigma_h^2O_{h\ell}^2\right)^2 - \sum_{h=1}^{k}\sigma_h^4O_{h\ell}^4\right).
\end{aligned}$$

Combing the computation for second and fourth moment, we obtain

$$
\begin{aligned}
\nu\big(R, \tilde{Z}\,\tilde{R}^\top\big) &= \sum_{\ell=1}^{k}\Big(\sum_{j=1}^{k}\eta_j\, O_{j\ell}^4 + 3\sum_{h\neq h'}\sigma_h^2\,\sigma_{h'}^2\, O_{h\ell}^2\, O_{h'\ell}^2\Big) \\
&\quad - \sum_{\ell=1}^{k}\Big(\sum_{j=1}^{k} O_{j\ell}^2\,\sigma_j^2\Big)^2 - \frac{1}{n}\sum_{\ell=1}^{k}\sum_{j=1}^{k}(\eta_j - \sigma_j^4)\, O_{j\ell}^4 \\
&\quad - \frac{2}{n}\sum_{\ell=1}^{k}\Big((\sum_{h=1}^{k}\sigma_h^2\, O_{h\ell}^2\big)^2 - \sum_{h=1}^{k}\sigma_h^4\, O_{h\ell}^4\Big) \\
&= \sum_{\ell=1}^{k}\sum_{j=1}^{k}\Big(\tfrac{n-1}{n}\eta_j - \tfrac{3n-3}{n}\sigma_j^4\Big)\, O_{j\ell}^4 + \Big(2-\tfrac{2}{n}\Big)\sum_{\ell=1}^{k}\Big(\sum_{j}\sigma_j^2\, O_{j\ell}^2\Big)^2 \\
&\leqslant \sum_{\ell=1}^{k}\sum_{j=1}^{k}\Big(\tfrac{n-1}{n}\eta_j - \tfrac{3n-3}{n}\sigma_j^4\Big) + \Big(2-\tfrac{2}{n}\Big)\sum_{\ell=1}^{k}\sum_{j}\sigma_j^4\sum_{j} O_{j\ell}^4 \\
&\leqslant \sum_{\ell=1}^{k}\sum_{j=1}^{k}\Big(\tfrac{n-1}{n}\eta_j - \tfrac{3n-3}{n}\sigma_j^4\Big) + \Big(2-\tfrac{2}{n}\Big)\sum_{\ell=1}^{k}\sum_{j}\sigma_j^4.
\end{aligned}
$$

Equality can and can only be achieved when O is a permutation matrix, where each row and each column have and only have exactly one 1. This completes the proof.

Proof of Theorem 4.2.1

Denote the generalized inverse of $\boldsymbol{C}_W^\top \boldsymbol{C}_W$ as $(\boldsymbol{C}_W^\top \boldsymbol{C}_W)^\dagger$. Then, by projecting from the column space of $\boldsymbol{C}_W$ to $\boldsymbol{C}^*$, we can rewrite the condition $\min_{P\in\mathbb{R}^{m\times k}} \|\boldsymbol{C}_W P - \boldsymbol{C}^*\|_F \geqslant \delta$ as

$$
\|\boldsymbol{C}_W(\boldsymbol{C}_W^\top \boldsymbol{C}_W)^\dagger \boldsymbol{C}_W^\top \boldsymbol{C}^* - \boldsymbol{C}^*\|_F \geqslant \delta. \tag{A.9}
$$

Similarly, by plugging $A = Z^* \boldsymbol{C}^{*\top}$,

$$\begin{aligned}\min_{Z \in \mathbb{R}^{n \times k}} \|A - Z\boldsymbol{C}_W^\top\|_F &= \min_{Z \in \mathbb{R}^{n \times k}} \|Z^* \boldsymbol{C}^{*\top} - Z\boldsymbol{C}_W^\top\|_F \\ &= \|Z^* \boldsymbol{C}^{*\top} - Z^* \boldsymbol{C}^{*\top} \boldsymbol{C}_W (\boldsymbol{C}_W^\top \boldsymbol{C})^\dagger \boldsymbol{C}_W^\top\|_F \\ &\geqslant \|\boldsymbol{C}_W (\boldsymbol{C}_W^\top \boldsymbol{C}_W)^\dagger \boldsymbol{C}_W^\top \boldsymbol{C}^* - \boldsymbol{C}^*\|_F \cdot \sigma_k(Z^*),\end{aligned}$$

where the inequality follows from Lemma A.5.1. We conclude the result by plugging equation A.9 in the above equation.

B APPENDIX FOR PART II

The appendix is structured as follows. It starts with the glossary table, defining key notations used throughout the paper in Appendix B.1. Next, Appendix B.2 discusses additional related work. In Appendix B.3, we introduce details about our tensor-based CARE algorithm, discussion for general CARE method, and additional discussion about method heuristics. Following this, Appendix B.4 offers theoretical support of our approach and supported proofs. It includes the graphical model formulation, graph structure recovery error bound, sample complexity, and the misspecification error arising from incorrectly characterized confounding factors. Subsequently, Appendix B.5 provides experimental details and additional experiment results. Finally, Appendix B.6 concludes by discussing the broader impacts and limitations of the work.

B.1 Glossary

The notations are summarized in Table B.1 below.

B.2 Extended Related Work

Biases in LLM–as–a–Judge

Large language models (LLMs) have quickly become the standard automatic evaluators for generation tasks because they correlate well with human judgments in translation and summarization (Kocmi and Federmann, 2023; Shen et al., 2023; Chiang and yi Lee, 2023). Yet a growing body of work shows that these models are far from impartial. **Positional bias**—preferring the *second* answer in a pairwise comparison—was first noted in MT-Bench (Zheng et al., 2023) and later quantified in detail by Wang

Table B.1: Glossary of variables and symbols used in this paper.

Symbol	Definition
$(J_1, \ldots, J_p)$	p vector of Judges score
Q	True-quality latent variable
$(C_1, \ldots, C_k)$	k latent confounder variables
H	All the hidden variables (true + confounder) i.e $(Q\ C_1, \ldots, C_k)$
h	dimension of H i.e all hidden variables = $k + 1$
X	Score matrix of dimension $(n \times p)$ where n is the number of examples and p is the number of judges
K	Precision matrix
K_{oo}	Observable-observable connection matrix
K_{oh}	Observable-latent connection matrix
K_{hh}	Latent-latent connection matrix
Σ_o	Covariance matrix of observable variables
S	Sparse matrix $(\mathbb{R}^{p \times p})$ which encodes edges between judges
L	Low-rank matrix (with $\mathrm{rank}(L) \leqslant h$) which captures dependencies mediated by latent variables
R	Rotation matrix $(\mathbb{R}^{h \times h})$
γ_n	Regularization for sparse and low-rank matrix S in Algorithm 4
τ	Regularization for low-rank matrix L in Algorithm 4
$\hat{s}_{agg}^{(i)}$	Aggregated scores for ith example in the dataset from p judges
$\hat{\Sigma}$	Sample precision estimation or covariance matrix
$\hat{S}$	Sample Sparse matrix $(\mathbb{R}^{p \times p})$ which encodes direct connectional edges among judges
$\hat{L}$	Sample Low-rank matrix (with $\mathrm{rank}(L) \leqslant h$) which captures dependencies mediated by latent variables
U	Latent factor extraction matrix i.e latent-judge connections $(\mathbb{R}^{p \times h})$ from Algorithm 4
Θ	Precision matrix
w	Weight for aggregating judges
λ	Singular values of L
$u^\star$	Singular vector of L corresponds to true quality factor
$\lambda^\star$	Singular value of L that corresponds to true quality factor
μ_{qc}	Conditional mean of judges given $Q = q, C = c$
$\hat{\mu}_{qc}$	Estimated conditional mean of judges given $Q = q, C = c$
π_{qc}	Probability of $Q = q, C = c$
$\hat{\pi}_{qc}$	Estimation of probability of $Q = q, C = c$
$\{\mathcal{G}_\ell\}_{\ell=1}^3$	Groups of judges that are independent of judges outside the group
$\hat{T}$	Empirical 3-way tensor
$\hat{\mu}_{qc}^{(1)}, \hat{\mu}_{qc}^{(2)}, \hat{\mu}_{qc}^{(3)}$	Estimated conditional mean of three views
$\hat{\mu}_{p(r)}$	Estimated conditional mean of judges after permutation
$\mu_{\mathrm{anchor}(r)}$	Conditional mean of anchor sets

et al. (2024a), who observed reversals of up to 30% when simply swapping order. **Verbosity bias**, wherein longer answers receive higher scores regardless of quality, is highlighted by Chen et al. (2024). LLM judges also display **self-enhancement bias**, overrating responses produced by models from the same family (Zeng et al., 2024). Less studied but equally problematic are **concreteness/authority biases**: judges over-reward answers that contain citations, numbers, or confident tone even when these features are irrelevant (Park et al., 2024).

Mitigation strategies span two levels. *Prompt-level interventions* randomize answer order, enforce symmetric formatting, and instruct the judge to

ignore superficial features (Wang et al., 2024a; Li et al., 2024d). Adding chain-of-thought rationales or decomposing the rubric into sub-criteria (accuracy, conciseness, style) also moderates shallow heuristics (Khan et al., 2024). On the *model level*, fine-tuned evaluators such as JudgeLM (Zhu et al., 2023) and Split-and-Merge Judge (Li et al., 2024d) are trained on curated data that explicitly counter positional and length biases. Peer-review and debate schemes go a step further: PRD lets a second LLM critique the first judge and often corrects biased decisions (Li et al., 2024c), while Khan et al. (2024) show that dialog with a more persuasive model leads to more truthful verdicts.

Despite progress, most debiasing work treats a *single* judge in isolation. When evaluations aggregate many LLM scorers—for robustness, cost sharing, or diversity—biases can compound in complex ways that individual fixes do not capture.

Label Aggregation for Multiple Noisy Evaluators

Weak-supervision. Treating each LLM prompt or model as a noisy *labeling function* aligns aggregation with modern weak supervision. Snorkel (Ratner et al., 2017; Bach et al., 2019) estimates source accuracies and dependencies to denoise programmatic labels, laying the foundation for LLM-prompt aggregation. Fu et al. (2020) introduces a scalable moment-matching estimator with closed-form weights.Shin et al. (2022) generalizes label models beyond categorical labels to arbitrary metric spaces, greatly expanding their applicability. Cachay et al. (2021) jointly optimizes a classifier and a differentiable label model, outperforming two-stage pipelines when sources are dependent. Firebolt further removes requirements on known class priors or source independence, estimating class-specific accuracies and correlations in closed form Kuang et al. (2022). Shin et al. (2023) shows that fixing source bias in labeling functions using optimal transport can improve both accuracy and fairness.

Aggregation of multiple *LLM* judges. Recent work shows that *ensembling smaller evaluators can beat a single large judge*. The **PoLL** jury combines three diverse 7–35B models and attains higher correlation with human ratings than GPT-4 while costing 7× less and reducing bias (Verga et al., 2024b). **GED** merges preference graphs from weak evaluators (Llama3-8B, Mistral-7B, Qwen2-7B) and denoises cycles; its DAG ranking surpasses a single 72B judge on ten benchmarks Hu et al. (2024). **JudgeBlender** ensembles either multiple models or multiple prompts, improving precision and consistency of relevance judgments over any individual LLM (Rahmani et al., 2024). These findings echo classic "wisdom-of-crowds" results—when paired with principled aggregation, a panel of smaller, heterogeneous judges can outperform a much larger model, offering a practical path toward reliable, low-cost evaluation.

Our Contribution in Context

Prior research either (i) debiases one judge at a time or (ii) aggregates multiple judges assuming independent noise. Our confounder-aware aggregation unifies these threads. We posit latent factors (e.g., verbosity, formality) that influence *all* judges simultaneously and show how to infer both the latent truth and the shared confounders. This yields more reliable consensus scores when individual judges—human or LLM—share systemic biases.

B.3 Algorithm Details

This section details the implementation of our CARE framework. Specifically, it includes the full CARE tensor algorithm, details about SVD baseline method for comparing our tensor-based algorithm, generalizations beyond Gaussian assumptions, and practical heuristics to address non-

orthogonality in latent factors and justification for where the sparse structure lies in mixed Gaussian data.

Tensor-based CARE Algorithm

SVD Baseline in Synthetic Experiment

We form the empirical two-way moment between view 1 and view 2:

$$\widehat{M}_{1,2} \;=\; \frac{1}{n}\sum_{i=1}^{n} X_1^{(i)}\, X_2^{(i)\,\top} \;=\; \sum_{q,c} \pi_{q,c}\; \mu_{1,q,c}\; \mu_{2,q,c}^{\top} \;+\; \text{sampling noise},$$

where $\pi_{q,c} = \Pr[Q = q, C = c]$ and $\mu_{v,q,c} = \mathbb{E}[J_v \mid Q = q, C = c]$ for judge/view v A singular-value decomposition

$$\widehat{M}_{1,2} \;=\; U_{12}\, \Sigma_{12}\, V_{12}^{\top}$$

yields factor matrices

$$U_{12}\, \Sigma_{12}^{1/2} \approx [\mu_{1,q,c}]\, R, \quad V_{12}\, \Sigma_{12}^{1/2} \approx [\mu_{2,q,c}]\, R,$$

where $R \in O(4)$ is an unknown orthogonal matrix.

Repeating on $\widehat{M}_{1,3} = \frac{1}{n}\sum_i X_1^{(i)} X_3^{(i)\,\top} = U_{13}\, \Sigma_{13}\, V_{13}^{\top}$ produces a second rotated copy of $[\mu_{1,q,c}]$. We solve the Procrustes problem

$$R \;=\; \arg\min_{O \in O(4)} \left\| U_{12}\, \Sigma_{12}^{1/2} \;-\; U_{13}\, \Sigma_{13}^{1/2}\, O \right\| * F,$$

then set $\hat{\mu}_{2,q,c} = (V_{12}\, \Sigma_{12}^{1/2})\, R^{\top}$ and $\hat{\mu}_{3,q,c} = (V_{13}\, \Sigma_{13}^{1/2})\, R^{\top}$ to align all three views.

This SVD baseline recovers $\{\mu_{v,q,c}\}$ up to the permutation/sign ambiguity inherent in any orthogonal transform.

Algorithm 9 CARE (T)

Input: Score matrix $J \in \mathbb{R}^{n \times p}$, tolerance ε.
Output: Estimates $\left\{\hat{\mu}_{qc}, \hat{\pi}_{qc}\right\}_{q,c \in \{0,1\}}$.

A. Anchor discovery (graph partition)

Compute the sample covariance $\hat{\Sigma} = J^\top J/n$ and perform the sparse+low-rank split $\hat{\Sigma} \approx \hat{S} + \hat{L}$ (Alg. 4).

Partition judges into three disjoint groups $\{\mathcal{G}_\ell\}_{\ell=1}^3$ that satisfy

$$a \neq b,\ j_1 \in \mathcal{G}_a,\ j_2 \in \mathcal{G}_b \implies |\hat{S}_{j_1,j_2}| \leqslant \varepsilon,$$

ensuring no direct edges with strength greater than ϵ can exist across groups.

B. Empirical third-order moment tensor

for $\ell = 1, 2, 3$ **do**
    $X_\ell \leftarrow$ columns of J indexed by $\mathcal{G}_\ell$ ▷ $X_\ell \in \mathbb{R}^{n \times |\mathcal{G}_\ell|}$
end for
Compute

$$\hat{T} = \frac{1}{n} \sum_{i=1}^{n} X_1^{(i)} \otimes X_2^{(i)} \otimes X_3^{(i)} \ \in\ \mathbb{R}^{|\mathcal{G}_1| \times |\mathcal{G}_2| \times |\mathcal{G}_3|}.$$

C. Tensor decomposition

Run a CP tensor-power decomposition on $\hat{T}$ to obtain $k = 4$ components $\left\{(\hat{\pi}_{qc}, \hat{\mu}_{qc}^{(1)}, \hat{\mu}_{qc}^{(2)}, \hat{\mu}_{qc}^{(3)})\right\}_{q,c \in \{0,1\}^2}$, where $\hat{\pi}_{qc} > 0$ and $\hat{\mu}_{qc}^{(\ell)} \in \mathbb{R}^{|\mathcal{G}_\ell|}$.

D. Assemble full means

for $q, c \in \{0,1\}^2$ **do**
    $\hat{\mu}_{qc} \leftarrow \mathrm{concat}\big(\hat{\mu}_{qc}^{(1)}, \hat{\mu}_{qc}^{(2)}, \hat{\mu}_{qc}^{(3)}\big) \in \mathbb{R}^p$.
end for

E. State alignment with anchors

Find the permutation ρ of $\{1, \ldots, 4\}$ that minimizes $\sum_{r=1}^{4} \big\|\hat{\mu}_{\rho(r)} - \mu_{\mathrm{anchor}(r)}\big\|_2^2$, where the four anchor prototypes correspond to $(Q, C) = \{00, 01, 10, 11\}$.

$(\hat{\mu}_{00}, \hat{\mu}_{01}, \hat{\mu}_{10}, \hat{\mu}_{11}) \leftarrow (\hat{\mu}_{\rho(1)}, \hat{\mu}_{\rho(2)}, \hat{\mu}_{\rho(3)}, \hat{\mu}_{\rho(4)})$.

F. Mixing weights

$(\hat{\pi}_{00}, \hat{\pi}_{01}, \hat{\pi}_{10}, \hat{\pi}_{11}) \leftarrow (\hat{\pi}_{\rho(1)}, \hat{\pi}_{\rho(2)}, \hat{\pi}_{\rho(3)}, \hat{\pi}_{\rho(4)})$.

return $\{\hat{\mu}_{qc}, \hat{\pi}_{qc}\}_{q,c \in \{0,1\}}$.

General CARE Setup

Extension Beyong the Gaussian Observation Model. The multivariate-Gaussian assumption for $J|H$ is convenient—its first two or three moments already encode all information needed for the sparse + low-rank and tensor steps—but it is not a requirement. Because CARE learns the *graphical* structure, the same pipeline applies whenever each judge's conditional distribution lies in an exponential family or, more generally, a latent-variable generalized linear model (GLM):

- **Categorical or ordinal scores.** For Likert ratings or pairwise preferences we can set

  $$J_i \mid H \sim \text{Categorical}\big(\text{softmax}(W_i^\top H)\big) \quad \text{or} \quad \text{Ordinal}-\text{logit}(W_i^\top H).$$

  The graph—hence the sparse mask S—is unchanged; only the node-wise likelihoods differ. We still recover S from conditional-mutual-information or pseudo-likelihood scores, and we still factorize higher-order indicator moments such as $\mathbb{E}\big[\mathbf{1}_{\{J_a=\ell\}}\,\mathbf{1}_{\{J_b=m\}}\,\mathbf{1}_{\{J_c=n\}}\big]$.

- **Mixed Discrete-Continous Scores.** When some judges output real scores and others categorical flags, we use a mixed conditional distribution:

  $$p(J|H) = \big[\Pi_{i\in\text{Cont.}}\mathcal{N}(J_i;\mu_{H_i},\sigma_i^2)\big]\,\big[\Pi_{j\in\text{Disc.}}\text{Bernoulli}(\sigma(W_j^\top H))\big]\,.$$

  CARE forms mixed raw/indicator moments, and identifiability again follows from standard tensor-decomposition guarantees for mixed conditional means.

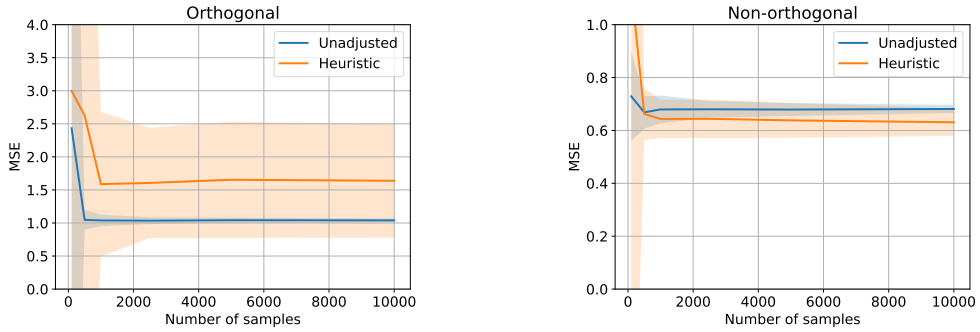
- **Heavy-tailed or skewed real scores.** When numeric scores are skewed or contain outliers, a multivariate-t or Gaussian scale mixture is appropriate. Up to a scalar factor, the covariance still decomposes

as sparse + low-rank, so Steps 1–2 of Algorithm 4 work after a simple rescaling.

Empirically, we find that replacing the Gaussian local likelihood only affects the estimation of sparse structure and extraction of latent factors, not the subsequent symmetry-breaking or posterior computation; thus the overall CARE pipeline generalizes with minimal adjustments.

Heuristics and Justifications

Heuristic for Addressing Orthogonality Violations in CARE (SVD).



(a) Orthogonal (b) Non-Orthogonal

Figure B.1: Effect of the proposed heuristic in a fully Gaussian synthetic setup. We estimate the true quality variable Q and report the mean squared error. The heuristic improves estimation in the non-orthogonal setting, but slightly degrades performance in the orthogonal setting where true and confounding components are disjoint.

Existing heuristics for identifying the true quality latent factor can estimate corresponding weights, but they often suffer from bias when the orthogonality assumption—central to the application of SVD—is violated. This issue commonly arises in real-world datasets. We found the following

weighting rule effective in both synthetic and real-world settings:

$$w \leftarrow \lambda^\star u^\star - \sum_{u_i \in U \setminus \{u^\star\}} \lambda_i u_i,$$

where w represents the learned weights for each judge, λ^* and u^* is the singular value and vector of L that corresponds to the direction that is closest to true quality latent variable, λ_i, u_i represent rest of the singular values and vectors, which can be interpreted as spurious/confounding factors.

This rule intuitively subtracts the influence of overlapping (non-orthogonal) confounding components from the estimated true score factor.

Figure B.1 illustrates the effect of this heuristic in a synthetic fully Gaussian setup. In the non-orthogonal case—where confounding components overlap with the true signal—the heuristic improves the estimation of the true latent variable. In contrast, it underperforms in the orthogonal case, where judges influenced by true scores are cleanly separated from those influenced by confounders.

Justification of Decomposing Covariance Matrix. In the joint Gaussian setting we decompose the *precision* matrix, whose sparsity pattern directly encodes conditional independences in an undirected graphical model. For a *mixed* Gaussian model, however, each observation $J \in \mathbb{R}^p$ is generated by first drawing a latent class label $Q, C \in \{0, 1\}^2$ (with probabilities π_{qc}) and then sampling

$$J \mid Q, C = q, c \;\sim\; \mathcal{N}\big(\mu_{qc}, \Sigma\big),$$

where the within-component covariance Σ does not depend on q, c. Because the latent variable only perturbs the mean, the marginal covariance

of J splits, via the *law of total covariance*, into

$$\mathrm{Cov}(J) \;=\; \underbrace{\mathbb{E}\big[\mathrm{Cov}(J \mid Q, C)\big]}_{=\Sigma} \;+\; \underbrace{\mathrm{Cov}\big(\mathbb{E}[J \mid Q, C]\big)}_{=\sum_{q,c} \pi_{qc}\,(\mu_{qc}-\bar{\mu})(\mu_{qc}-\bar{\mu})^{\top}} \quad, \quad \bar{\mu} := \sum_{q,c} \pi_{qc}\mu_{qc}. \tag{B.1}$$

The first term, Σ, is the same sparse block-diagonal matrix we plant in the simulator to model direct judge–judge interactions; the second term is an outer-product mixture of at most 4 linearly independent directions and hence has rank $\leqslant 4$. Equation equation B.1 therefore exhibits the population covariance as a *sparse* + *low-rank* decomposition,

$$\mathrm{Cov}(J) \;=\; S \;+\; L, \quad S = \Sigma \text{ (sparse)}, \quad L = \mathrm{Cov}\big(\mathbb{E}[J \mid Q, C]\big) \text{ (low rank)}.$$

Because sparsity now lives in S, not in the inverse covariance, estimating S and L by fitting a sparse-plus-low-rank model directly to the empirical covariance is both natural and statistically identifiable for the mixed Gaussian case.

B.4 Theory

Proof of Theorem 10.2.3

Proof. Let low-rank matrix satisfies $L = \sum_{i=1}^{h} d_i\, u_i u_i^{\top}$ with u_i the i-th column of K_{oh}. By Assumption 10.2.2 the u_i are mutually orthogonal, and by Assumption 10.2.1 the eigenvalues $d_1 > \cdots > d_h$ are distinct; hence this rank-1 decomposition is the (unique) spectral decomposition of L. Thus each u_i is identifiable from L up to sign and ordering, proving the theorem. □

Proof of Theorem 10.2.4

Proof. We apply standard matrix perturbation theory for eigenvectors. Starting from the eigenvalue decomposition:

$$L\,u_i = \lambda_i\,u_i,$$

we write the perturbed matrix as

$$\tilde{L} = (K_{oh} + E)K_{hh}^{-1}(K_{oh} + E)^\top = L \;+\; K_{oh}K_{hh}^{-1}E^\top \;+\; EK_{hh}^{-1}K_{oh}^\top \;+\; EK_{hh}^{-1}E^\top.$$

Let $\Delta L = \tilde{L} - L$. By the Davis–Kahan theorem,

$$\|\hat{u}_i - u_i\|_2 \;\leqslant\; \frac{2\,\|\Delta L\|_2}{\delta_i},$$

where $\delta_i = \min_{j \neq i} |\lambda_i - \lambda_j| > 0$. Moreover,

$$\|\Delta L\|_2 \;\leqslant\; 2\,\|K_{oh}\|_2\,\|K_{hh}^{-1}\|_2\,\|E\|_2 \;+\; O(\|E\|_2^2)$$

and $\|K_{oh}\|_2 = 1$. Hence

$$\|\hat{u}_i - u_i\|_2 \;\leqslant\; \frac{2\,\|K_{hh}^{-1}\|_2\,\|E\|_2}{\delta_i} \;+\; O(\|E\|_2^2).$$

This completes the proof. □

Proof of Theorem 10.3.1

Proof of Theorem 10.3.1. **Step 1 – Spectral error of $\hat{L}_n$.** Apply Chandrasekaran et al.'s Theorem 4.1 with the regularization parameters

$$\gamma_n \;=\; \frac{48\sqrt{2}D\psi(2-\nu)}{\xi(T)\nu}\sqrt{\frac{\epsilon}{n}}, \qquad \sigma, \theta \text{ as in their conditions (3)–(4).}$$

Under the incoherence and curvature conditions of their Proposition 3.3, there exists a universal constant $C_1 > 0$ such that, with probability at least

$1 - 2e^{-\epsilon}$,

$$\left\| \hat{L}_n - L^* \right\|_2 \leqslant C_1 \frac{\sqrt{\epsilon/n}}{\xi(T)}. \tag{B.2}$$

Step 2 – Eigenvector perturbation via Davis–Kahan. Let $L^* = U \Lambda U^\top$ with $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_h, 0, \ldots, 0)$ and collect the top–h eigenvectors in $U_h = [u_1, \ldots, u_h]$. Write the spectral decomposition of the estimator as $\hat{L}_n = \hat{U}_h \hat{\Lambda} \hat{U}_h^\top + R$, where R contains only the eigen-components of rank $> h$. Set the perturbation $E := \hat{L}_n - L^*$.

Applying Corollary 3 from Yu et al. (2015) to the i-th eigenpair gives

$$\|u_i - \hat{u}_i\|_2 \leqslant \frac{2^{3/2} \|E\|_2}{\delta_i}. \tag{B.3}$$

Step 3 – Combine the two bounds. Insert equation B.2 into equation B.3:

$$\| \hat{u}_i - u_i \|_2 \leqslant \frac{2^{3/2} C_1}{\delta \, \xi(T)} \sqrt{\frac{\epsilon}{n}} \qquad \forall \, i \in [h],$$

and take the maximum over i. This proves the advertised high-probability bound

$$\max_{i \leqslant h} \| \hat{u}_i - u_i \|_2 = O\Big(\tfrac{\sqrt{\epsilon/n}}{\xi(T)\,\delta} \Big).$$

Step 4 – Invert to a sample-size requirement. Setting the right-hand side to a target accuracy $\varepsilon \in (0, 1)$ and solving for n yields $n \geqslant \frac{4C_1^2}{\varepsilon^2} \frac{\epsilon}{\xi(T)^2 \delta^2}$, which is the sample-complexity statement in the theorem. □

Proof of Theorem 10.4.1

Proof. By Davis-Kahan theorem (Theorem 2 in Yu et al. (2015)), if $\|\mathbf{E}\|_{\text{op}} \leqslant \delta/2$, then the ℓ_2 distance between $\mathbf{v}_1$ and $\mathbf{u}_1^{\text{true}}$ (after aligning their signs via $s = \pm 1$) is bounded by:

$$\left|\left| \mathbf{v}_1 - s \cdot \mathbf{u}_1^{\text{true}} \right|\right|_2 \leqslant \frac{2 \, \|\mathbf{E}\|_{\text{op}}}{\delta}.$$

Plugging in E yields the desired result:

$$\left|\left|\mathbf{v}_1 - s \cdot \mathbf{u}_1^{\text{true}}\right|\right|_2 \leqslant \frac{2\left|\left|\sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T\right|\right|_{\text{op}}}{\frac{1}{d_1}\left\|\mathbf{k}_1\right\|_2^2}.$$

□

Proof of Corollary 10.4.2

Proof. The absolute difference is:

$$\begin{aligned}
\left|\mathbb{E}[Q|\mathbf{o}]_{\text{mis}} - \mathbb{E}[Q|\mathbf{o}]_{\text{true}}\right| &= \left|-\frac{\left\|\mathbf{k}_1\right\|_2}{d_1}(s \cdot \mathbf{v}_1)^T\mathbf{o} - \left(-\frac{\left\|\mathbf{k}_1\right\|_2}{d_1}(\mathbf{u}_1^{\text{true}})^T\mathbf{o}\right)\right| \\
&= \left|-\frac{\left\|\mathbf{k}_1\right\|_2}{d_1}(s \cdot \mathbf{v}_1 - \mathbf{u}_1^{\text{true}})^T\mathbf{o}\right| \\
&= \frac{\left\|\mathbf{k}_1\right\|_2}{d_1}\left|(s \cdot \mathbf{v}_1 - \mathbf{u}_1^{\text{true}})^T\mathbf{o}\right|
\end{aligned}$$

By the Cauchy-Schwarz inequality, $\left|(\mathbf{x})^T\mathbf{y}\right| \leqslant \left\|\mathbf{x}\right\|_2 \left\|\mathbf{y}\right\|_2$. Applying this:

$$\left|\mathbb{E}[Q|\mathbf{o}]_{\text{mis}} - \mathbb{E}[Q|\mathbf{o}]_{\text{true}}\right| \leqslant \frac{\left\|\mathbf{k}_1\right\|_2}{d_1}\left|\left|s \cdot \mathbf{v}_1 - \mathbf{u}_1^{\text{true}}\right|\right|_2 \left\|\mathbf{o}\right\|_2$$

The term $\left\|s \cdot \mathbf{v}_1 - \mathbf{u}_1^{\text{true}}\right\|_2$ is equivalent to $\left\|\mathbf{v}_1 - s \cdot \mathbf{u}_1^{\text{true}}\right\|_2$ from the main theorem statement, where s aligns $\mathbf{u}_1^{\text{true}}$ with $\mathbf{v}_1$. From the preceding Theorem, we have the bound (where $\delta = \frac{1}{d_1}\left\|\mathbf{k}_1\right\|_2^2$):

$$\left|\left|\mathbf{v}_1 - s \cdot \mathbf{u}_1^{\text{true}}\right|\right|_2 \leqslant \frac{2\left\|\mathbf{E}\right\|_{\text{op}}}{\delta} = \frac{2\left|\left|\sum_{\ell=2}^{h} \frac{1}{d_\ell}\mathbf{k}_\ell\mathbf{k}_\ell^T\right|\right|_{\text{op}}}{\frac{1}{d_1}\left\|\mathbf{k}_1\right\|_2^2}$$

Substituting this bound into the inequality for the error in the conditional

mean:

$$
\begin{aligned}
|\mathbb{E}[Q|\mathbf{o}]_{\text{mis}} - \mathbb{E}[Q|\mathbf{o}]_{\text{true}}| &\leqslant \frac{\|\mathbf{k}_1\|_2}{d_1} \left(\frac{2\,\|\mathbf{E}\|_{\text{op}}}{\frac{1}{d_1}\,\|\mathbf{k}_1\|_2^2} \right) \|\mathbf{o}\|_2 \\
&= \frac{\|\mathbf{k}_1\|_2}{d_1} \cdot \frac{2 d_1\,\|\mathbf{E}\|_{\text{op}}}{\|\mathbf{k}_1\|_2^2} \cdot \|\mathbf{o}\|_2 \\
&= \frac{2\,\|\mathbf{E}\|_{\text{op}}}{\|\mathbf{k}_1\|_2} \|\mathbf{o}\|_2 \\
&= \frac{2 \left\| \sum_{\ell=2}^{h} \frac{1}{d_\ell} \mathbf{k}_\ell \mathbf{k}_\ell^T \right\|_{\text{op}}}{\|\mathbf{k}_1\|_2} \|\mathbf{o}\|_2
\end{aligned}
$$

□

Proof for Theorem 10.4.4

Proof sketch. **Step 1: Concentration of the empirical tensor.** Let $\hat{M} := \frac{1}{n} \sum_{i=1}^{n} X_1^{(i)} \otimes X_2^{(i)} \otimes X_3^{(i)}$. Because each X_ℓ is sub-Gaussian with proxy $\sigma_{\max}$, the operator-norm Bernstein bound for order-3 tensors (Lemma 5 of Hsu and Kakade, 2013) yields

$$
\|\hat{M} - M\|_{\text{op}} \;=\; O\Big(\sigma_{\max}^3 \sqrt{\tfrac{p \log(p/\varepsilon)}{n}} \Big) \quad \text{w.p. } 1 - \varepsilon/2.
$$

Step 2: Robust CP decomposition. Applying the non-symmetric tensor power method of (Anandkumar et al., 2014, Alg. 2) to $\hat{M}$ and invoking their perturbation bound (Theorem 5.1 therein) gives, for every component $r \in [4]$,

$$
\big\| (\hat{a}_r, \hat{b}_r, \hat{c}_r) - (a_r, b_r, c_r) \big\|_2 \;=\; O\Big(\tfrac{1}{\delta}\, \|\hat{M} - M\|_{\text{op}} \Big).
$$

Step 3: Assembling full means. Algorithm 9 concatenates the three block-means, so $\hat{\mu}_r - \mu_r = (\hat{a}_r - a_r, \hat{b}_r - b_r, \hat{c}_r - c_r)$, and the same $O(\cdot)$ factor carries through.

Step 4: Mixing-weight estimation. Given accurate factor recovery, the usual least-squares re-estimation of weights satisfies $|\hat{\pi}_{qc} - \pi_{qc}| = O(\|\hat{M} - M\|_{op})$ (Anandkumar et al., 2014, Theorem B.1), yielding the stated rate.

Step 5: Union bound. Combine Steps 1–4 and union-bound over the four components to obtain the final probability $1 - \varepsilon$. □

B.5 Experiment Details

In this section, we provide experimental details and additional experiment results. We describe datasets details, evaluation prompts we used to collect LLM judgments, and individual judge performance. In addition, we introduce the construction of programmatic judge, and ablation studies including prompt-based interventions. Finally, we include additional experiment results for our tensor-based CARE algorithm: synthetic experiments results on graph-aware judge partition, and real-world applications.

Datasets

FeedbackQA (Li et al., 2022). A question-answering dataset with human-provided scalar ratings of answer helpfulness, ranging from 1 to 5. We use the validation set in our experiments, treating the average of two human ratings as the ground truth.

HelpSteer2 (Wang et al., 2024b). A large-scale dataset of assistant responses annotated with real-valued scores (0 to 4) across multiple axes, including helpfulness, correctness, coherence, complexity, and verbosity. We use the validation set and take the helpfulness score as the ground truth.

UltraFeedback (Cui et al., 2023). A scalar feedback dataset where assistant responses are rated from 0 to 10 based on overall quality, using scores aggregated from GPT-4 and human raters. We randomly sample 5,000 examples for evaluation.

Synthetic Dataset (Section 11.6). We construct a synthetic dataset with latent state probabilities set to $\pi_{qc} = [0.2, 0.2, 0.3, 0.3]$, corresponding to latent states (Q, C) as $(0,0), (0,1), (1,0), (1,1)$ respectively. The judges are organized into three distinct groups, each containing four judges whose conditional means μ_{qc} are randomly drawn from a uniform distribution ranging between 1 and 4. Dependence structures are imposed explicitly: for judges independent of the true quality variable Q, we constrain their conditional means such that averages depend solely on the confounder C (i.e., rows corresponding to $Q = 0$ and $Q = 1$ are identical for each C state).

Prompt Templates

In this subsection we provide the prompts we used for collecting LLM judgements.

LLM Judge Scoring Template (FeedbackQA, HelpSteer2, Ultrafeedback)

You will be given a user_question and system_answer couple.

Your task is to provide a 'total rating' scoring how well the system_answer answers the user concerns expressed in the user_question. Give your answer as a float on a scale of {min_rating} to {max_rating}, where {min_rating} means that the system_answer is not helpful at all, and {max_rating} means that the answer completely and helpfully addresses the question.

Provide your feedback as follows:

Feedback:::
Total rating: (your rating, as a float between {min_rating} and {max_rating})

Now here are the question and answer.

Question: {question}
Answer: {answer}

Feedback:::
Total rating:

Intervened LLM Judge Scoring Template (Section 11.4)

You are a careful and impartial evaluator. Your task is to rate how well the system_answer responds to the user_question, focusing on substance, correctness, and completeness.

Rate the answer on a scale from {min_rating} to {max_rating}, where:
- {min_rating} means the answer does not address the question or is misleading,
- {max_rating} means the answer fully and helpfully addresses the question with accurate, relevant, and concise information.

Avoid common judgment biases:
- Do **not** reward irrelevant fluency, length, or politeness.
- Do **not** penalize answers solely for brevity if they are complete.
- Do **not** be influenced by phrasing style unless it affects clarity or correctness.

Judge only the answer's content in relation to the question. Ignore formatting, grammar, or tone unless they impact understanding or utility.

Provide your feedback as follows:

Feedback:::
Total rating: (a float from {min_rating} to {max_rating})

Now, here are the inputs:

Question: {question}
Answer: {answer}

Feedback:::
Total rating:

LLM Judge Toxicity Scoring Template (CivilComments)

You will be given a user_question and system_answer couple.

Your task is to provide a 'total rating' scoring how well the system_answer answers the user concerns expressed in the user_question. Give your answer as a float on a scale of {min_rating}

to {max_rating}, where {min_rating} means that the system_answer is not helpful at all, and {max_rating} means that the answer completely and helpfully addresses the question.

Provide your feedback as follows:

Feedback:::
Total rating: (your rating, as a float between {min_rating} and {max_rating})

Now here are the question and answer.

Question: {question}
Answer: {answer}

Feedback:::
Total rating:

Individual LLM Judge Performance

Table B.2 shows the performance of individual LLM judges. Overall, we find that aggregation methods outperform or match the best single judge. Since the relative performance of individual LLM judges is not known a priori in practical settings, aggregation offers a useful approach for integrating judges in an agnostic manner.

Table B.2: Individual Judge Performance in Section 11.1

	FeedbackQA		HelpSteer2		UltraFeedback	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
gemma-3-1b-it	1.0073	0.2315	1.0666	0.0825	1.0606	0.1812
gemma-3-4b-it	**0.7578**	0.4537	0.9920	0.1402	0.8492	0.2309
Llama-3.1-8B-Instruct	0.8148	0.4341	1.1364	0.1261	0.8648	0.3194
Llama-3.2-1B	1.2219	-0.0525	1.0049	-0.0132	1.0119	0.0752
Llama-3.2-3B	1.0362	0.0051	0.9995	0.0251	1.1522	0.1648
Mistral-7B-Instruct-v0.3	1.0244	0.4539	1.0793	0.1116	0.8572	0.1735
Phi-4-mini-instruct	0.8082	**0.4557**	1.0692	0.1576	0.8355	0.3147
Qwen3-0.6B	1.0969	0.2073	1.1255	0.0370	1.0233	0.1254
Qwen3-1.7B	1.1507	0.2485	1.0693	0.1049	1.1382	0.1926
Qwen3-4B	1.0999	0.2854	**0.9675**	**0.2290**	**0.7088**	**0.3921**
Qwen3-8B	1.0517	0.4417	**0.9675**	0.2094	0.7512	0.3140

Programmatic Judges

Programmatic judges, synthesized by LLMs, distill and convert evaluation logic into interpretable, cheap-to-obtain program code (Huang et al., 2025). These program judges provide specialized, independent assessments compared to using LLMs directly as evaluators. We integrate these judge sets into CARE to enhance evaluation signals.

We describe the creation of programmatic judges and the criteria they encode. Using OpenAI's GPT-4o (Hurst et al., 2024), we generate judges with the following prompt:

Program Judge Template

You are now a judge to evaluate LLM generated response with a given question. You will write your evaluation logic into code and generate python programs to return their scores. Higher represents better response quality. Consider complex criteria for assessing responses, leveraging third-party models, embedding models, or text score evaluators as needed.

Function signature: def _judging_function(self, question, re-

sponse):

We synthesize 23 programs and select 10 representative ones for our experiments (see Section 11.2 and Section 11.3). These programs evaluate responses based on diverse criteria: (i) structure, (ii) readability, (iii) safety, (iv) relevance, and (v) factuality. For example:

- **Structure**: A judge counts transition markers (e.g., "therefore," "however") to assess coherence, with more markers indicating better quality.
- **Relevance**: A judge uses TF-IDF to convert questions and responses into vectors, computing cosine similarity to measure semantic alignment (see Program B.1). Another employs semantic embeddings for similarity metrics (see Program B.2).
- **Readability**: A judge leverages a third-party API to evaluate complexity, using metrics like the Flesch–Kincaid grade level (see Program B.3).

All judging logic, conditions, and pre-defined keyword lists are generated by the LLM. Below, we provide examples to illustrate this approach.

Listing B.1: Lexical Overlap Computation using TF-IDF.

```
def _lexical_overlap(self, question, response):
    """Compute lexical overlap using TF–IDF for
      ↪ relevance evaluation."""
    # Preprocess input question and response (e.g.,
      ↪ lowercase, remove stopwords)
    question_clean = self._preprocess(question)
    response_clean = self._preprocess(response)
```

```
    # Return 0.0 if either input is empty after
      ↪ preprocessing
    if not question_clean.strip() or not
      ↪ response_clean.strip():
        return 0.0

    # Transform inputs to TF–IDF vectors using the
      ↪ vectorizer
    tfidf_matrix =
      ↪ self.tfidf_vectorizer.fit_transform([question_clean,
      ↪ response_clean])
    question_vec = tfidf_matrix[0]  # Extract
      ↪ question vector
    response_vec = tfidf_matrix[1]  # Extract
      ↪ response vector

    # Compute cosine similarity between vectors and
      ↪ return as float
    return float(cosine_similarity(question_vec,
      ↪ response_vec)[0][0])
```

Listing B.2: Semantic Similarity using Embedding Model.

```
def _semantic_similarity_strong(self, question,
  ↪ response):
    """Compute semantic similarity between question
      ↪ and response."""
    # Return 0.0 if either input is empty
    if not question.strip() or not response.strip():
        return 0.0

    # Encode question and response into dense
```

```
        ↪ vectors using the embedding model
    question_embedding =
        ↪ self.semantic_embedding_strong_model.encode(
          question, max_length=4096
    )['dense_vecs']
    response_embedding =
        ↪ self.semantic_embedding_strong_model.encode(
          response, max_length=4096
    )['dense_vecs']

    # Compute dot product similarity between
        ↪ embeddings
    similarity = question_embedding @
        ↪ response_embedding

    # Clamp similarity score between 0.0 and 1.0 and
        ↪ return as float
    return float(max(0.0, min(1.0, similarity)))
```

Listing B.3: Readability Metrics Calculation.

```
def _readability(self, response):
    """Calculate readability metrics for response."""
    # Compute readability scores using textstat
        ↪ library
    return {
        # Flesch Reading Ease (inverted: higher
            ↪ score means harder to read)
        "flesch_reading_ease": 100 -
            ↪ textstat.flesch_reading_ease(response),
        # SMOG Index (higher score indicates higher
            ↪ reading difficulty)
```

```
        "smog_index": textstat.smog_index(response),
    }
```

We report the performance of individual program judges in Table B.3. While their standalone performance is limited, they provide useful signals for the integration strategies discussed in Sections 11.2 and 11.3.

Table B.3: Program Judge Performance. (*) represents the selected judges in Section 11.2.

	FeedbackQA		**HelpSteer2**		**UltraFeedback**	
	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)	MAE (↓)	τ (↑)
factuality_check_score (*)	1.9956	0.0872	1.1992	0.0075	1.1910	0.0492
factuality_factKB_score (*)	1.0343	0.2288	1.7180	0.0414	1.4342	0.1051
readability_flesch_reading (*)	1.2185	0.0431	2.5682	0.0445	2.5145	0.1396
readability_smog (*)	0.9805	0.1277	2.3286	0.0283	2.3122	0.1604
relevance_bleu	1.4035	0.0126	2.7452	-0.0355	2.7330	0.0560
relevance_keyword_overlap	1.2779	0.1977	2.3735	0.0138	2.2725	0.1461
relevance_lexical_overlap (*)	1.1371	0.2316	2.0148	-0.0144	1.9182	0.1187
relevance_rouge	1.3079	0.2066	2.5603	0.0232	2.4838	0.1327
relevance_semantic_sim_strong (*)	**0.8759**	**0.4092**	1.1182	0.0395	**0.9866**	0.1601
safety_toxicity (*)	1.5396	-0.0380	**1.1105**	0.0300	1.0139	-0.0043
structure_avg_paragraph_length_dist	1.4560	-0.1883	2.5562	-0.0081	2.4637	0.1074
structure_avg_sentence_length_dist	1.5248	-0.0140	2.4407	-0.0287	2.4179	0.1612
structure_cohesion_score	1.4078	0.2070	2.7139	0.0345	2.6578	0.1567
structure_emphasis_count	1.2826	0.1988	2.6642	0.0482	2.5955	0.2060
structure_headings	1.4765	0.0423	2.6521	-0.0340	2.5916	0.1049
structure_lexical_diversity	1.0672	0.1625	2.1864	0.0444	2.0981	0.1935
structure_list_usage	1.6284	0.0159	3.0208	-0.0108	3.0132	0.0872
structure_logical_transitions (*)	1.2694	0.1743	2.2693	0.0520	2.4355	0.2263
structure_max_sentence_length (*)	1.3039	0.1272	2.7532	0.0104	2.7511	0.1377
structure_min_sentence_length	1.3568	0.1887	2.4872	0.0400	2.4322	0.2046
structure_questions	1.2443	0.2692	2.4910	0.0360	2.4064	0.2114
structure_sentence_balance	1.4423	0.1835	2.6757	0.0501	2.6444	0.2203
structure_sentence_count (*)	1.3099	0.1742	2.4408	**0.0807**	2.6570	**0.2300**

Effects of Prompt-Based Intervention (Section 11.4)

We begin by analyzing how the intervention using a robust prompt affects the performance of individual LLM judges. Figures B.2 (MAE) and B.3 (Kendall's τ) present the performance differences relative to the vanilla prompt. While the intervention aims to reduce confounding signals, its impact varies—some model–dataset combinations show improvement, while others show degradation.

We then assess how these shifts influence aggregate performance. Figures B.4 and B.5 show the corresponding changes in aggregation accuracy. Most baseline methods benefit from the intervention, whereas CARE shows a slight performance drop. A plausible explanation is that once confounding signals are mitigated, the additional latent variables in CARE may begin to model residual noise rather than meaningful structure, slightly reducing its performance. Nevertheless, as shown in Section 11.4, CARE without intervention still outperforms other baselines with the robustness prompt, highlighting its effectiveness even without manually crafted interventions for hidden confounders.

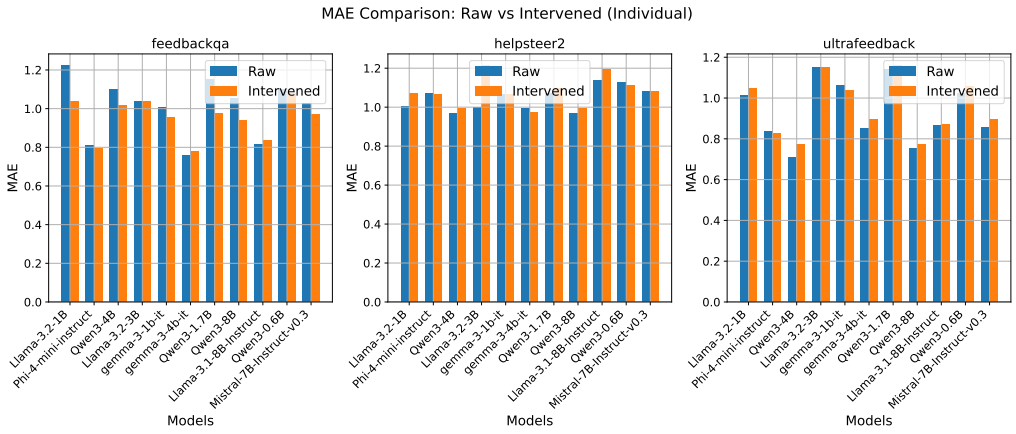


Figure B.2: Change in MAE (↓) for individual LLM judges after applying the robustness prompt.

Additional Real-World Experiment on Gaussian Mixture

We consider a Gaussian mixture setting where the latent variable is binary, but the observables (judge outputs) are real-valued Gaussian scores. This experiment evaluates the effectiveness of Algorithm 9 on a real dataset.

Setup. We use a subset of the CivilComments dataset (Borkan et al., 2019), randomly sample 5,000 examples. The ground-truth label is binary

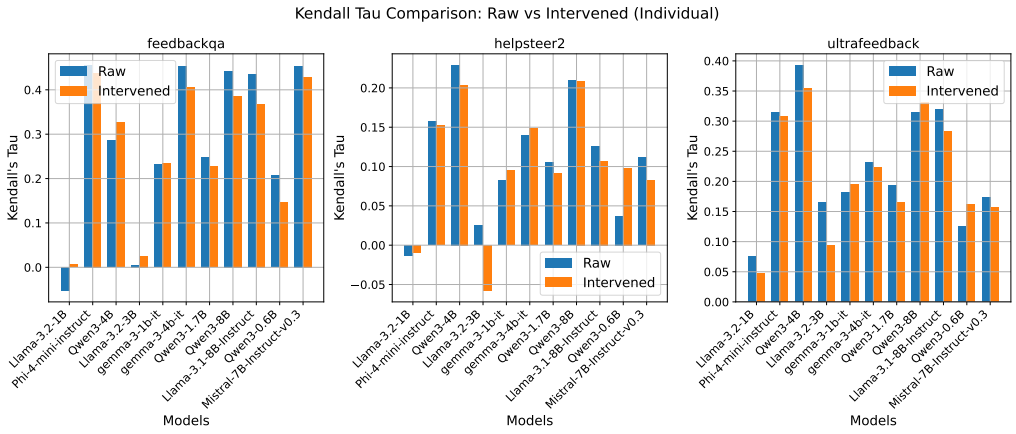


Figure B.3: Change in Kendall's τ ($\uparrow$) for individual LLM judges after the robustness prompt.

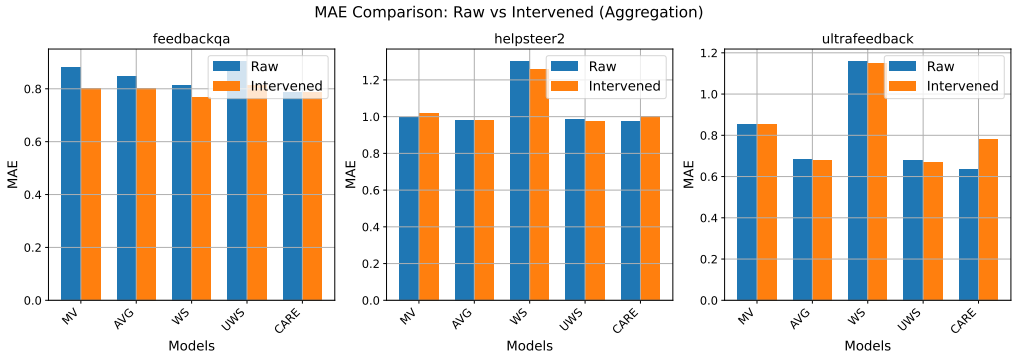


Figure B.4: Change in aggregate MAE ($\downarrow$) after propagating the robustness prompt through each aggregation method.

toxicity (0 or 1), while LLM judges provide real-valued toxicity scores ranging from 0 to 9. In addition to the original LLM judges, we include five LLMs:

- `meta-llama/Meta-Llama-3-8B-Instruct`,
- `mistralai/Mistral-7B-Instruct-v0.2`,
- `Qwen/Qwen2.5-0.5B-Instruct`,

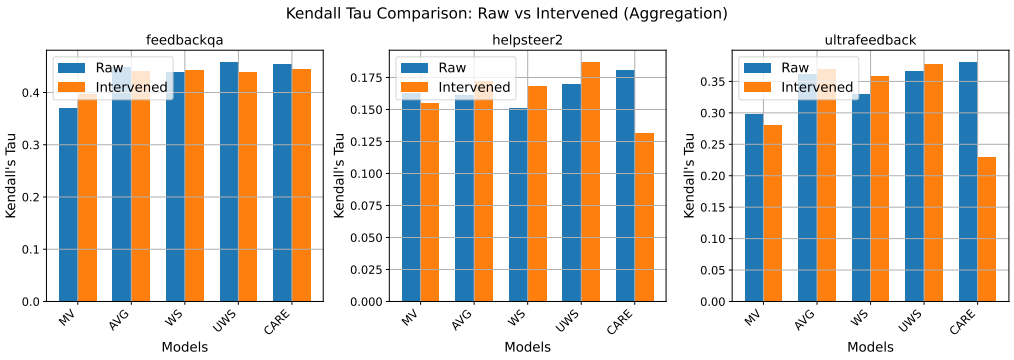


Figure B.5: Change in aggregate Kendall's τ ($\uparrow$) after the robustness prompt.

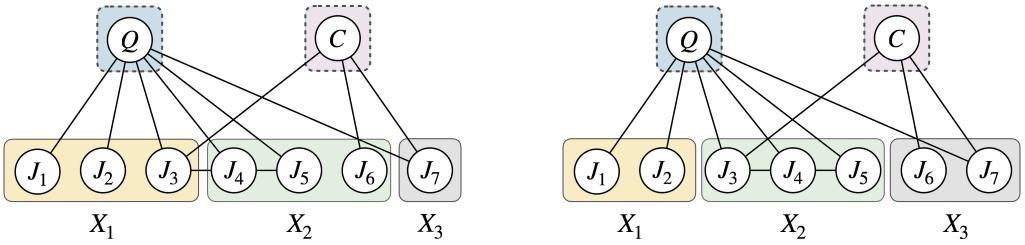
- `Qwen/Qwen2.5-1.5B-Instruct`,
- `Qwen/Qwen2.5-3B-Instruct`.

For the MV and WS baselines, we first discretize judge scores using a threshold of 4.5 before applying majority vote or weighted sum. For AVG and UWS, we aggregate scores first, then apply the threshold. CARE (Algorithm 9) directly infers the latent binary label from continuous scores. We evaluate all methods using classification accuracy.

Table B.4: Aggregated accuracy (higher is better) in CivilComments dataset.

Method	**Acc. (%)**
MV	74.32%
AVG	73.80%
WS	74.95%
UWS	74.95%
CARE	**75.27%**

Results. Table B.4 shows that CARE achieves the highest accuracy. This result highlights its ability to better handle confounding factors and per-



(a) Random Partitioning (b) Graph-aware Partitioning

Figure B.6: Random Partitioning vs. Graph Aware Partitioning. A random partitioning (a) leaves cross-view edges that violate the independence assumptions of tensor methods, whereas the graph-aware partitioning (b) considers cross-view edges and restores the required separation.

form effective latent inference, even when the observed data (continuous scores) differ from the latent variable type (binary labels).

Synthetic Experiment on Graph-Aware Tensor Decomposition

When judges exhibit conditional dependencies, naively partitioning them into views violates the independence assumptions required by tensor decomposition. We hypothesize that partitioning judges via a graph-aware procedure that respects dependency structure yields substantially better estimation than random partitioning.

Setup. We simulated 10,000 items scored by $p = 12$ judges, partitioned into three views of four judges each. To induce conditional dependencies, we planted edges of strength 0.3 within each true view at 40% density. We then compared two grouping strategies across ten random seeds:

- **Random**: assign judges to views uniformly at random;
- **Graph-Aware**: assign views to minimize cross-block edges in the empirical precision matrix.

Performance was measured by the ℓ_2 error in recovering the latent component means, i.e. $\|\mu_{qc} - \hat{\mu}_{qc}\|_2$).

Results. As shown in Figure B.7, graph-aware grouping dramatically reduces reconstruction error—by more than an order of magnitude—compared to random grouping. This confirms the importance of respecting dependency structure during view formation and underscores the advantage of CARE, which integrates graph structure directly into the tensor decomposition procedure.

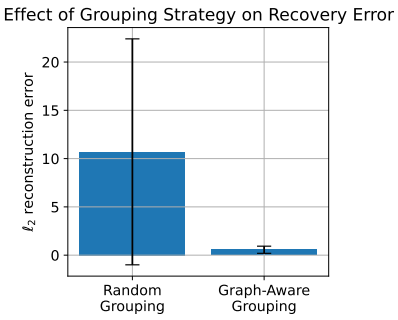


Figure B.7: ℓ_2 reconstruction error (mean $\pm$ SD) for random vs. graph-aware grouping.

Computing Resources

We used a server equipped with an NVIDIA RTX 4090 (24GB). Generating LLM judge outputs took up to 3 hours per dataset. In contrast, the aggregation algorithms were efficient, completing in under 1 minute for datasets with approximately 5,000 rows.

B.6 Broader Impact Statement

This work presents a novel approach to aggregate scores from multiple LLMs serving as judges by identifying confounding variables and thus potentially reducing the bias in the overall judge scores. The potential broader impact includes a framework for improved LLM-as-a-judge scores which can be used at various applications. However, it is important to acknowledge that using LLMs as potential judges to automate labor-intense annotation tasks which is an active area of research carries some limitations

and past research has discussed some unintended consequences, such as over-reliance on judge outputs, misuse and misinterpretation of results which might carry high real-world stakes. It is crucial to use automated LLM-as-a-judge tools responsibly and ethically, considering potential biases in data and models, and ensuring transparency and accountability in their application.

REFERENCES

Abouelenin, Abdelrahman, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Anandkumar, Animashree, Rong Ge, Daniel J Hsu, Sham M Kakade, Matus Telgarsky, et al. 2014. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.* 15(1):2773–2832.

Bach, Stephen H, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. 2019. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 international conference on management of data*, 362–375.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Balasubramanian, Sriram, Samyadeep Basu, and Soheil Feizi. 2024. Decomposing and interpreting image representations via text in vits beyond clip. *arXiv preprint arXiv:2406.01583*.

Beery, Sara, Elijah Cole, and Arvi Gjoka. 2020. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*.

Bhalla, Usha, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. 2024. Interpreting clip with sparse linear concept embeddings (splice). *arXiv preprint arXiv:2402.10376*.

Borkan, Daniel, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, 491–500.

Cachay, Sebastian Rühling, Benjamin Boecking, and Artur Dubrawski. 2021. End-to-end weak supervision. In *Advances in neural information processing systems*.

Chakrabarti, Deepayan, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. 2004. Fully automatic cross-associations. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*, 79–88. KDD '04, New York, NY, USA: Association for Computing Machinery.

Chandrasekaran, Venkat, Pablo A. Parrilo, and Alan S. Willsky. 2012. Latent variable graphical model selection via convex optimization. *The Annals of Statistics* 40(4).

Chen, Guiming Hardy, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, 8301–8327. Miami, Florida, USA: Association for Computational Linguistics.

Chen, Song Xi, Li-Xin Zhang, and Ping-Shou Zhong. 2010. Tests for high-dimensional covariance matrices. *Journal of the American Statistical Association* 105(490):810–819.

Chen, Wei, Hongjun Wang, Zhiguo Long, and Tianrui Li. 2023. Fast flexible bipartite graph model for co-clustering. *IEEE Transactions on Knowledge and Data Engineering* 35(7):6930–6940.

Chiang, Cheng-Han, and Hung yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st annual meeting of the association for computational linguistics (acl)*, 15607–15631.

Cui, Ganqu, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Dawid, Alexander Philip, and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* 20–28.

Deutsch, Daniel, Rotem Dror, and Dan Roth. 2022. On the limitations of reference-free evaluations of generated text. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, 10960–10977. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Dhillon, Inderjit S. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh acm sigkdd international conference on knowledge discovery and data mining*, 269–274. KDD '01, New York, NY, USA: Association for Computing Machinery.

Fel, Thomas, Victor Boutin, Louis Béthune, Rémi Cadène, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. 2024. A holistic approach to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems* 36.

Fu, Daniel Y., Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. 2020. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th international conference on machine learning (icml 2020)*.

Furniturewala, Shaz, Surgan Jandial, Abhinav Java, Pragyan Banerjee, Simra Shahid, Sumit Bhatia, and Kokil Jaidka. 2024. "thinking" fair

and slow: On the efficacy of structured prompts for debiasing language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, ed. Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, 213–227. Miami, Florida, USA: Association for Computational Linguistics.

Gandelsman, Yossi, Alexei A Efros, and Jacob Steinhardt. 2024a. Interpreting clip's image representation via text-based decomposition. In *The twelfth international conference on learning representations*.

———. 2024b. Interpreting the second-order effects of neurons in clip. *arXiv preprint arXiv:2406.04341*.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences* 120(30):e2305016120.

Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Graziani, Mara, An-phi Nguyen, Laura O'Mahony, Henning Müller, and Vincent Andrearczyk. 2023. Concept discovery and dataset exploration with singular value decomposition. In *Iclr 2023 workshop on pitfalls of limited data and computation for trustworthy ml*.

Guo, Yue, Yi Yang, and Ahmed Abbasi. 2022. Auto-debias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 1012–1023. Dublin, Ireland: Association for Computational Linguistics.

Hovy, Dirk, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of naacl-hlt*, 1120–1130.

Hsu, Daniel, and Sham M Kakade. 2013. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on innovations in theoretical computer science*, 11–20.

Hu, Zhengyu, Jieyu Zhang, Zhihan Xiong, Alexander Ratner, Hui Xiong, and Ranjay Krishna. 2024. Language model preference evaluation with multiple weak evaluators. *arXiv preprint arXiv:2410.12869*.

Huang, Tzu-Heng, Catherine Cao, Vaishnavi Bhargava, and Frederic Sala. The alchemist: Automated labeling 500x cheaper than llm data annotators. In *The thirty-eighth annual conference on neural information processing systems*.

Huang, Tzu-Heng, Catherine Cao, Spencer Schoenberg, Harit Vishwakarma, Nicholas Roberts, and Frederic Sala. 2025. Scriptoriumws: A code generation assistant for weak supervision. *arXiv preprint arXiv:2502.12366*.

Hurst, Aaron, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Hyvärinen, Aapo, Ilyes Khemakhem, and Hiroshi Morioka. 2023. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns* 4(10).

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-

Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. 2310.06825.

Jiao, Tong, Jian Zhang, Kui Xu, Rui Li, Xi Du, Shangqi Wang, and Zhenbo Song. 2024. Enhancing fairness in llm evaluations: Unveiling and mitigating biases in standard-answer-based evaluations. In *Proceedings of the aaai symposium series*, vol. 4, 56–59.

John, S. 1971. Some optimal multivariate tests. *Biometrika* 58(1):123–127.

Kaiser, Henry F. 1958. The varimax criterion for analytic rotation in factor analysis. *Psychometrika* 23(3):187–200.

Kendall, Maurice. 1938. A new measure of rank correlation. *Biometrika* 81–89.

Khan, Akbir, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rockt"aschel, and Ethan Perez. 2024. Debating with more persuasive LLMs leads to more truthful answers. *arXiv preprint arXiv:2402.06782*.

Kocmi, Tom, and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th annual conference of the european association for machine translation (eamt)*, 193–203.

Koh, Pang Wei, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. 2012.07421.

Kuang, Zheng, Chidubem Arachie, Brian Liang, Pratyush Narayana, Grace DeSalvo, Michael Quinn, Bo Huang, Gabriel Downs, and Yiming Yang. 2022. Firebolt: Weak supervision under weaker assumptions. In *Proceedings of the 25th international conference on artificial intelligence and statistics*.

Le, Can M, Elizaveta Levina, and Roman Vershynin. 2017. Concentration and regularization of random graphs. *Random Structures & Algorithms* 51(3):538–561.

Ledoit, Olivier, and Michael Wolf. 2002. Some hypothesis tests for the covariance matrix when the dimension is large compared to the sample size. *The Annals of Statistics* 30(4):1081–1102.

Li, Chenghui, Rishi Sonthalia, and Nicolas Garcia Trillos. 2023. Spectral neural networks: Approximation theory and optimization landscape. *arXiv preprint arXiv:2310.00729*.

Li, Dawei, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference leakage: A contamination problem in llm-as-a-judge. 2502.01534.

Li, Haitao, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. 2024a. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. *arXiv preprint arXiv:2410.15393*.

Li, Haitao, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024b. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Li, Jingxuan, and Tao Li. 2010. Hcc: a hierarchical co-clustering algorithm. In *Proceedings of the 33rd international acm sigir conference on research and development in information retrieval*, 861–862. SIGIR '10, New York, NY, USA: Association for Computing Machinery.

Li, Jingxuan, Bo Shao, Tao Li, and Mitsunori Ogihara. 2012. Hierarchical co-clustering: A new way to organize the music data. *IEEE Transactions on Multimedia* 14(2):471–481.

Li, Ruosen, Teerth Patel, and Xinya Du. 2024c. PRD: Peer rank and discussion improve large language model based evaluations. *Transactions on Machine Learning Research*.

Li, Zichao, Prakhar Sharma, Xing Han Lu, Jackie Chi Kit Cheung, and Siva Reddy. 2022. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the association for computational linguistics: Acl 2022*, 926–937.

Li, Zongjie, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024d. Split and merge: Aligning position biases in LLM-based evaluators. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, 11084–11108. Miami, Florida, USA: Association for Computational Linguistics.

Liu, Haotian, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of international conference on computer vision (iccv)*.

Mauchly, John W. 1940. Significance test for sphericity of a normal n-variate distribution. *The Annals of Mathematical Statistics* 11(2):204–209.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north American chapter of the association for computational linguistics: Human language technologies*, ed. Lucy Vanderwende, Hal

Daumé III, and Katrin Kirchhoff, 746–751. Atlanta, Georgia: Association for Computational Linguistics.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35: 27730–27744.

Park, Junsoo, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the association for computational linguistics: Emnlp 2024*, 1043–1067.

Park, Sungjoon, JinYeong Bak, and Alice Oh. 2017. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, 401–411.

Pensa, Ruggero G., Dino Ienco, and Rosa Meo. 2014. Hierarchical co-clustering: off-line and incremental approaches. *Data Mining and Knowledge Discovery* 28(1):31–64.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rahmani, Hossein A., Emine Yilmaz, Nick Craswell, and Bhaskar Mitra. 2024. Judgeblender: Ensembling judgments for automatic relevance assessment. *arXiv preprint arXiv:2412.13268*.

Ratner, Alexander, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the vldb endowment*, vol. 11, 269–282.

Roeder, Geoffrey, Luke Metz, and Durk Kingma. 2021. On linear identifiability of learned representations. In *International conference on machine learning*, 9030–9039. PMLR.

Rohe, Karl, Tai Qin, and Bin Yu. 2012. Co-clustering for directed graphs: the stochastic co-blockmodel and spectral algorithm di-sim. *arXiv preprint arXiv:1204.2296*.

Rohe, Karl, and Muzhe Zeng. 2020. Vintage factor analysis with varimax performs statistical inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

———. 2023. Vintage factor analysis with varimax performs statistical inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 85(4):1037–1060. `https://academic.oup.com/jrsssb/article-pdf/85/4/1037/52714936/qkad029.pdf`.

Roucher, Aymeric. n.d. Using LLM-as-a-judge for an automated and versatile evaluation. `https://huggingface.co/learn/cookbook/en/llm_judge`. Accessed: 2025-05-15.

Sagawa, Shiori, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Shen, Chenhui, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In *Findings of the association for computational linguistics: Emnlp 2023*, 4215–4233.

Shi, Lin, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791*.

Shin, Changho, Sonia Cromp, Dyah Adila, and Frederic Sala. 2023. Mitigating source bias for fairer weak supervision. In *Advances in neural information processing systems* (*neurips*).

Shin, Changho, Winfred Li, Harit Vishwakarma, Nicholas Carl Roberts, and Frederic Sala. 2022. Universalizing weak supervision. In *International conference on learning representations* (*iclr*).

Team, Gemma, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Team, Qwen. 2025. Qwen3.

Verga, Pat, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024a. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Verga, Pat, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, et al. 2024b. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Wang, Peiyi, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics* (*acl*), 9440–9450.

Wang, Victor, Michael JQ Zhang, and Eunsol Choi. 2025. Improving llm-as-a-judge inference with the judgment distribution. *arXiv preprint arXiv:2503.03064*.

Wang, Yidong, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, et al. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *The twelfth international conference on learning representations*.

Wang, Zhilin, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024b. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.

Whitehill, Jacob, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul Ruvolo. 2009. Whose vote should count more? optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, vol. 22, 2035–2043.

Ye, Jiayi, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2025. Justice or prejudice? quantifying biase in LLM-as-a-judge. In *The thirteenth international conference on learning representations*.

Ye, Jiayi, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. In *Neurips safe generative ai workshop 2024*.

Yu, Yi, Tengyao Wang, and Richard J Samworth. 2015. A useful variant of the davis–kahan theorem for statisticians. *Biometrika* 102(2):315–323.

Zeng, Zhiyuan, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *Proceedings of the 12th international conference on learning representations (iclr)*.

Zhang, Ruihan, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. 2021. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *Proceedings of the AAAI Conference on Artificial Intelligence* 35(13):11682–11690.

Zhang, Yilin, and Karl Rohe. 2018. Understanding regularized spectral clustering via graph conductance. *Advances in Neural Information Processing Systems* 31.

Zheng, Liang, Yuzhong Qu, Xinqi Qian, and Gong Cheng. 2018. A hierarchical co-clustering approach for entity exploration over linked data. *Knowledge-Based Systems* 141:200–210.

Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh conference on neural information processing systems datasets and benchmarks track*.

Zhu, Lianghui, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.