Mutual Predictability Decomposition: Learning Interpretable Cross-Set Structure via Bi-Directional Prediction

Zixi Qin*
University of Wollongong
zq181@uowmail.edu.au

Yuanhao Zhuo*
University of Wollongong
yzhuo@uow.edu.au

Wanqing Li†
University of Wollongong
wanqing@uow.edu.au

Guoxin Su
University of Wollongong
guoxin@uow.edu.au

Abstract

Understanding relationships between variable sets is fundamental in machine learning, particularly in medical data analysis. Existing methods typically provide only coarse shared/private partitioning of variables, limiting their ability to reveal finer-grained predictive structures across views. We propose Mutual Predictability Decomposition (MPD), a framework that decomposes the variables in each view into three subsets: mutually predictable variables that can be inferred across views, auxiliary variables that are themselves not cross-view predictable but are necessary for predicting the mutually predictable subset, and unpredictable variables that do not contribute to cross-view predictability. MPD is formulated as a coupled bi-directional prediction problem with sparse variable assignment, and optimized through a differentiable relaxation implemented using gated neural networks. Experiments on synthetic and real-world medical datasets demonstrate that MPD recovers interpretable cross-view structures, improves cross-view prediction, and provides more informative decompositions than existing relationship analysis methods. *Code is publicly available at* `https://github.com/UOW-AMRL/NeurIPS26-MPD`.

1 Introduction

Understanding relationships between variable sets is a fundamental problem in machine learning, with broad applications in multimodal learning [Liu and Zheng, 2022], medical data analysis [Krones et al., 2025, Zhuo et al., 2026], and representation learning [Yin et al., 2025, Liu et al., 2026]. In many real-world settings, data arise from multiple views or sources—for example, clinical variables and laboratory measurements in healthcare [Li et al., 2020, Dauvin et al., 2019]. **A central challenge is to determine not only which variables are predictive across views, but also the distinct roles that variables play in cross-view prediction.**

Existing approaches provide only partial answers to this question, as shown in Table 1. Feature selection methods identify variables useful for prediction, but do not distinguish whether these variables are themselves predictable from other sources [Theng and Bhoyar, 2024]. Correlation-based methods such as canonical correlation analysis (CCA) [Weenink, 2003, Ullah and Arora, 2023, Li et al., 2024] identify shared structure between variable sets, but primarily operate at the representation

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

level and provide limited interpretability at the level of individual variables. Similarly, modern multimodal learning methods focus on representation alignment or predictive performance without explicitly characterizing the functional roles of variables in cross-view relationships [Radford et al., 2021, Ma et al., 2024]. As a result, existing methods typically yield only coarse shared/private distinctions and fail to capture richer predictive dependencies among variables.

In this work, we introduce a new perspective on cross-view relationships through the lens of *mutual predictability*. We argue that variables should be characterized not only by whether they are useful for prediction, but also by whether they are themselves predictable from another view. This leads to a ***ternary decomposition of variables*** in each view into three disjoint subsets: (i) *mutually predictable* variables that can be reliably inferred across views, (ii) *auxiliary* variables that are themselves not cross-view predictable but are necessary for predicting the mutually predictable subset, and (iii) *unpredictable* variables that neither contribute to nor participate in cross-view prediction.

This decomposition reveals structural relationships that are not captured by conventional shared/private formulations. In particular, auxiliary variables play a fundamentally different role from shared variables: although they cannot themselves be reconstructed from another view, they provide essential contextual information that enables accurate prediction of predictable variables. Distinguishing these roles is especially important in domains such as healthcare, where certain measurements may be indispensable for inference despite lacking direct cross-view correspondence.

Based on this idea, we propose the *Mutual Predictability Decomposition* (MPD) framework. MPD formulates the decomposition as a coupled bi-directional prediction problem between two variable sets, where predictable variables and their supporting auxiliary subsets must be jointly identified. This induces a challenging combinatorial optimization problem due to the circular dependency between the subsets: the predictability of a variable depends on the selected auxiliary variables, while the utility of auxiliary variables depends on the predictable subset they support.

To address this challenge, we derive a differentiable relaxation that replaces discrete subset assignments with sparse continuous gating and soft predictability scores, enabling end-to-end optimization with neural networks. The resulting framework is scalable, flexible, and applicable to high-dimensional data settings.

We evaluate the proposed method on both synthetic and real-world medical datasets. On synthetic data with known ground-truth structure, MPD accurately recovers mutually predictable, auxiliary, and unpredictable variables under varying noise conditions. On real-world datasets, MPD improves cross-view prediction performance while providing interpretable insights into variable relationships that are not revealed by existing approaches.

Our contributions are summarized as follows:

- We introduce the problem of *mutual predictability decomposition*, which partitions variables into mutually predictable, auxiliary, and unpredictable subsets based on cross-view predictability.
- We formulate the decomposition as a coupled bi-directional prediction problem that jointly identifies predictable variables and the auxiliary variables required to support their prediction.
- We develop a differentiable relaxation with sparse gating that enables scalable end-to-end optimization in high-dimensional settings.
- We demonstrate on synthetic and real-world medical datasets that the proposed framework recovers interpretable variable-level structure and improves cross-view prediction performance over existing methods.

Overall, this work provides a new framework for understanding cross-view relationships by explicitly disentangling predictability, utility, and structural dependency among variables.

2 Related Work

Feature selection and sparse learning. Feature selection methods aim to identify subsets of variables that are predictive of a target while promoting sparsity and interpretability. Classical approaches include LASSO [Tibshirani, 1996] and elastic net regularization [Zou and Hastie, 2005], along with numerous sparse regression and structured sparsity formulations [Bach et al., 2012]. These

Table 1: Comparison between existing paradigms and the proposed MPD framework.

Method family	Primary objective	Limitation relative to MPD
CCA / Deep CCA	Learn correlated shared structure	No variable-role decomposition
Sparse regression	Select predictive variables	No cross-view predictability semantics
Shared-private learning	Separate shared/private representations	No auxiliary predictive role modeling
Information decomposition	Quantify redundancy/synergy	Not prediction-driven or scalable
Causal discovery	Recover causal dependencies	Focus on causality rather than predictive roles
MPD (ours)	Decompose variables into predictable, auxiliary, and unpredictable subsets	—

methods focus on predictive utility but do not distinguish whether selected variables are themselves predictable from another view. In contrast, our framework explicitly separates variables that are mutually predictable from those that only play an auxiliary predictive role.

Canonical correlation analysis and multi-view learning. Canonical correlation analysis (CCA) [Hotelling, 1992] and its extensions aim to identify correlated structure between multiple variable sets. Variants including kernel CCA [Akaho, 2006], deep CCA [Andrew et al., 2013], and sparse CCA [Witten et al., 2009, Lindenbaum et al., 2021, Li et al., 2024] improve nonlinear modeling and variable selection capabilities. More broadly, multi-view and multimodal learning methods seek shared representations across modalities through alignment, contrastive learning, or cross-modal prediction [Radford et al., 2021, Zhang et al., 2024, Wang et al., 2023]. While these approaches identify shared information or aligned latent spaces, they generally do not characterize the distinct predictive roles of variables. Our approach instead performs an explicit variable-level decomposition into mutually predictable, auxiliary, and unpredictable subsets.

Shared-private and disentangled representation learning. A closely related line of work decomposes representations into shared and private components across views or modalities [Bousmalis et al., 2016, Yang et al., 2022]. Such methods have been widely used in domain adaptation, multimodal learning, and disentangled representation learning. However, shared/private decompositions typically operate in latent representation spaces and treat private variables as residual modality-specific information. In contrast, our framework distinguishes between variables that are merely view-specific and variables that are functionally necessary for predicting cross-view structure despite not being directly predictable themselves. This auxiliary role is not explicitly modeled in existing shared/private formulations.

Information-theoretic dependency analysis. Information-theoretic methods characterize statistical dependence using measures such as mutual information, redundancy, and synergy [Cover and Thomas, 1991, Williams and Beer, 2010]. Recent work on partial information decomposition (PID) seeks to disentangle redundant, unique, and synergistic information contributions among variables [Bertschinger et al., 2014, Makkeh et al., 2021]. Although conceptually related, these methods primarily focus on information attribution rather than predictive decomposition, and are often difficult to scale to high-dimensional settings. Our framework instead provides a prediction-driven and scalable variable decomposition based on cross-view predictability.

Causal and dependency modeling. Causal modeling targets causal directions or interventional effects, whereas dependency modeling characterizes statistical or conditional-dependence structures [Pearl, 2009, Richardson et al., 2023, Montagna et al., 2023]. These approaches provide powerful tools for reasoning about causation and conditional independence, but typically rely on strong structural assumptions and are not designed to identify the predictive roles of variables across views. In contrast, MPD focuses on bidirectional prediction across views without assuming that either view causes the other. Its ternary decomposition is defined by cross-view predictability and predictive utility, providing variable-level insight into which measurements can be predicted from the opposite view and which measurements support those predictions, rather than estimating a causal or dependency structure.

Subset selection and differentiable combinatorial optimization. The proposed formulation is also

related to subset selection and combinatorial optimization, which are generally NP-hard. Recent work has explored differentiable relaxations for discrete optimization using continuous gates, stochastic masks, or sparsity-inducing mechanisms [Louizos et al., 2018, Yamada et al., 2020]. Our method builds on this idea by introducing sparse differentiable assignments for jointly identifying mutually predictable and auxiliary variable subsets.

3 Ternary Decomposition and Differentiable Relaxation

Given two variable sets $X = \{X_1, \ldots, X_p\}$ and $Y = \{Y_1, \ldots, Y_q\}$, we seek partitions $X = X_{\mathrm{MP}} \cup X_{\mathrm{AO}} \cup X_{\mathrm{UP}}$ and $Y = Y_{\mathrm{MP}} \cup Y_{\mathrm{AO}} \cup Y_{\mathrm{UP}}$, where X_{MP} (Y_{MP}) is reliably predictable from Y (X), X_{AO} (Y_{AO}) is not mutually predictable but required for predicting Y_{MP} (X_{MP}), and X_{UP} (Y_{UP}) is neither predictable nor useful for mutual prediction. The *ternary decomposition* is formulated as a joint optimization:

$$\max_{X_{\mathrm{MP}}, Y_{\mathrm{MP}}, \mathcal{I}_X, \mathcal{I}_Y} \quad |X_{\mathrm{MP}}| + |Y_{\mathrm{MP}}| \tag{1}$$

$$\text{s.t.} \quad X_{\mathrm{MP}} = \{X_i \in X \mid \mathcal{E}(X_i; g(\mathcal{I}_Y)) \le \varepsilon\}, \qquad Y_{\mathrm{MP}} = \{Y_j \in Y \mid \mathcal{E}(Y_j; f(\mathcal{I}_X)) \le \varepsilon'\},$$
$$\mathcal{I}_X \in \arg\min_{S \subseteq X} \mathcal{E}(Y_{\mathrm{MP}}; f(S)), \qquad \mathcal{I}_Y \in \arg\min_{T \subseteq Y} \mathcal{E}(X_{\mathrm{MP}}; g(T)).$$

where $\mathcal{E}$ is the prediction loss (e.g., MSE), ε is a threshold defining predictability, f, g are the predictors, and $\mathcal{I}_X, \mathcal{I}_Y$ are the essential supporting sets. Auxiliary and unpredictable sets follow: $X_{\mathrm{AO}} = \mathcal{I}_X \setminus X_{\mathrm{MP}}$, $X_{\mathrm{UP}} = X \setminus (\mathcal{I}_X \cup X_{\mathrm{MP}})$ (analogous for Y).

Differentiable Relaxation. Eq. (1) defines a discrete optimization over the mutually predictable sets X_{MP}, Y_{MP} and the supports $\mathcal{I}_X, \mathcal{I}_Y$. For the direction $X \to Y$, Eq. (1) seeks a subset $Y_{MP} \subseteq Y$ with maximal cardinality such that every selected variable can be predicted from a minimal support $\mathcal{I}_X \subseteq X$ under the prescribed error tolerance. The reverse direction $Y \to X$ is defined analogously. The resulting difficulty is that both the predictor support and the predictable subset are discrete and mutually coupled, which makes Eq. (1) combinatorial and non-differentiable.

To derive a tractable surrogate, we relax the discrete set indicators into continuous gates. For $X \to Y$, let $s_d^X \in [0,1],\ d = 1, \ldots, p$, denote the relaxed indicator for whether X_d belongs to the auxiliary support $\mathcal{I}_X$, and let $h_j^Y \in [0,1],\ j = 1, \ldots, q$, denote the relaxed indicator for whether Y_j belongs to the mutually predictable set Y_{MP}. We then define the gated input

$$\tilde{X} = s^X \odot X, \qquad \text{and predictor} \qquad \hat{Y} = f(\tilde{X}),$$

where $f(\cdot)$ is the predictor introduced in Sec. 3.1. Similarly, for $Y \to X$, we introduce gates s^Y and h^X and define

$$\tilde{Y} = s^Y \odot Y, \qquad \hat{X} = g(\tilde{Y}).$$

Under this relaxation, the discrete objective in Eq. (1) is replaced by a soft optimization that balances three requirements: (i) selected targets should be predicted accurately, (ii) the auxiliary support should remain sparse, and (iii) as many predictable targets as possible should be retained. The relaxed directional objective for $X \to Y$ is written as

$$\mathcal{L}_{X \to Y}^{\mathrm{relax}} = \frac{1}{q} \sum_{j=1}^{q} h_j^Y \, \mathrm{MSE}_j + \lambda_{in} R_{in}(s^X) - \lambda_{out} G_{incentive}(h^Y), \tag{2}$$

where MSE_j denotes the per-dimension prediction error for Y_j, R_{in} denotes the input-gate sparsity regularizer, and $G_{incentive}$ denotes the output-gate activation reward. Likewise, the reverse-direction relaxed objective is

$$\mathcal{L}_{Y \to X}^{\mathrm{relax}} = \frac{1}{p} \sum_{i=1}^{p} h_i^X \, \mathrm{MSE}_i + \lambda_{in} R_{in}(s^Y) - \lambda_{out} G_{incentive}(h^X). \tag{3}$$

The full differentiable relaxation of Eq. (1) is therefore

$$\min_{\Theta} \mathcal{L}_{X \to Y}^{\mathrm{relax}} + \mathcal{L}_{Y \to X}^{\mathrm{relax}}, \tag{4}$$

where Θ contains the parameters of the predictors and all gate variables.

Recovering discrete subsets. The soft gates are thresholded at $\tau_h = 0.5$ and $\tau_s = 0.5$ to recover discrete sets:

$$Y_{\mathrm{MP}} = \{j : h_j^Y > \tau_h\}, \qquad X_{\mathrm{MP}} = \{i : h_i^X > \tau_h\},$$
$$\mathcal{I}_X = \{d : s_d^X > \tau_s\}, \qquad \mathcal{I}_Y = \{r : s_r^Y > \tau_s\}.$$

The remaining subsets follow deterministically:

$$X_{\mathrm{AO}} = \mathcal{I}_X \setminus X_{\mathrm{MP}}, \qquad X_{\mathrm{UP}} = X \setminus (\mathcal{I}_X \cup X_{\mathrm{MP}}),$$

with analogous definitions for Y.

3.1 Deep Gated Neural Network (DGNN) Implementation

The **Deep Gated Neural Network (DGNN)** is a practical neural instantiation of the differentiable relaxation in Eq. (4).

DGNN contains two symmetric branches: $X \to Y$ and $Y \to X$. The branch $X \to Y$ learns which variables in X are useful for predicting variables in Y, while the reverse branch learns the complementary mapping. Joint training of both directions yields the final ternary partition on both variable sets.

Each branch consists of three modules: an *input selector* that identifies a sparse predictive support and is implemented using stochastic gates (STG) [Yamada et al., 2020], an MLP-based *predictor* that maps the selected inputs to all target variables, and an *output selector* that determines which targets are reliably predictable and is implemented using hard-concrete gates (HCG) [Louizos et al., 2018].

Training schedule. Directly optimizing all gates simultaneously can be unstable because input and output selection interact strongly. We therefore adopt a three-stage schedule:

1. *Support discovery:* train predictors with STG input gates while predicting all targets;
2. *Predictable-target discovery:* freeze the input gates and optimize the output gates;
3. *Joint refinement:* fix selected outputs and fine-tune input supports and predictors.

This staged optimization improves convergence and empirically yields more stable decompositions than fully joint training. Detailed model architecture and training procedures are provided in Appendix B.3.2 and Appendix B.3.3.

3.2 Theoretical Analysis

The following proposition formalizes that Eqs. (2)– (4) constitute a differentiable relaxation of Eq. (1).

Proposition 1 (Differentiable relaxation of Eq. (1))**.** *Let s^X, s^Y be continuous surrogates of the auxiliary-support indicators for $\mathcal{I}_X, \mathcal{I}_Y$, and let h^Y, h^X be continuous surrogates of the membership indicators for Y_{MP}, X_{MP}. Then the optimization problem in Eqs. (2)–(4) is a differentiable relaxation of Eq. (1): the cardinality terms $|Y_{MP}|$ and $|X_{MP}|$ are replaced by soft coverage rewards $\sum_j h_j^Y$ and $\sum_i h_i^X$, while the minimal-support constraints on $\mathcal{I}_X$ and $\mathcal{I}_Y$ are replaced by sparsity regularizers on s^X and s^Y. (See Section A of the supplementary material for proof).*

The next theorem shows that the discrete formulation is recovered when the relaxed gates become binary.

Theorem 1 (Consistency with Eq. (1))**.** *Assume that at convergence the relaxed gates satisfy $s_d^X, s_d^Y, h_i^X, h_j^Y \in \{0, 1\}$ for all dimensions. Then any stationary point of Eq. (4) induces a feasible discrete solution to Eq. (1) via*

$$\mathcal{I}_X = \{X_d : s_d^X = 1\}, \qquad Y_{MP} = \{Y_j : h_j^Y = 1\},$$

and analogously

$$\mathcal{I}_Y = \{Y_d : s_d^Y = 1\}, \qquad X_{MP} = \{X_i : h_i^X = 1\}.$$

Moreover, under this binary limit, the relaxed objective reduces to the discrete objective up to the surrogate approximation used for the error constraint. (See Section A for proof).

In Section A.1 of the supplementary material, we also show that, under the assumptions stated there, the proposed MP/AO/UP decomposition is (i) uniquely defined at the population level, (ii) recoverable from finite samples in a sparse linear regime, and (iii) approximated by the differentiable relaxation in Eq. (4).

4 Experiments

We design experiments to evaluate three central claims: **Recoverability:** the proposed MP/AO/UP decomposition can recover the ground-truth role structure when such structure is known. **Utility:** the recovered decomposition yields strong cross-view prediction and compact downstream feature sets relative to existing multi-view and sparse-selection baselines. **Meaningfulness of AO:** auxiliary-only variables are neither merely unpredictable noise nor selected supports; instead, they are essentially useful for predicting MP variables in the opposite view.

Experimental Setup We compare DGNN with representative baselines that operate in the original variable space and capture either sparse cross-view associations or sparse cross-view predictions on both synthetic and real datasets.

Baselines: (a) ***Sparse CCA (SCCA)***, (b) ***Deep Sparse CCA (DSCCA)***, and (c) ***Multi-task Lasso (Lasso)***. Note that SCCA and DSCCA only generate partitions MP and AO $\cup$ UP, and Lasso only generates partitions MP $\cup$ AO and UP.

Evaluation protocols & metrics: All methods are evaluated under the same train/validation/test splits, and hyperparameters are selected using the validation data. $F1$ score is used for recoverability of MP/AO/UP. To measure the utility, i.e., cross-view prediction, R^2 and the concordance correlation coefficient (CCC) over the predicted targets are used. Variables whose CCC is above a threshold θ are considered as predictable. For SCCA and DSCCA, predictions are made between X(MP) $\leftrightarrow$ Y(MP) and predictions are made between X(MP $\cup$ AO) $\leftrightarrow$ Y(MP $\cup$ AO) for Lasso because of their binary partition nature. For DGNN, predictions are made between X (MP $\cup$ AO) $\rightarrow$ Y (MP) and Y (MP $\cup$ AO) $\rightarrow$ X (MP). We report the mean and standard deviation over 10 runs unless otherwise stated.

4.1 Synthetic Datasets

Synthetic datasets are first used for evaluation. We generate paired views (X, Y) with known MP, AO, and UP variables under **three regimes**: (a) ***R1: Linear persistent-support AO*** constructs linear cross-view mappings in which AO variables provide stable support for predicting opposite-view MP variables but are not themselves mutually predictable; (b) ***R2: Nonlinear persistent-support AO*** preserves the same role structure but replaces the linear mappings with nonlinear interactions and heterogeneous noise; and (c) ***R3: Ambiguity-resolution AO*** creates branch, sign, phase, or many-to-one ambiguities, where AO variables are needed to disambiguate the opposite-view MP variables, but are not recoverable as MP variables themselves.

Results on synthetic data of $n = 5000$ samples with $p = 21$ and $q = 19$, and at a moderate noise level $\mathrm{SNR} = 13\ \mathrm{dB}$ are reported below. We further examine robustness across lower and higher noise levels in Appendix C.1. Details about the data-generation are provided in Appendix B.1.

Results:

We first evaluate whether the learned variable sets match the ground-truth roles in synthetic data. Table 2 reports the recovery performance for the individual MP, AO, and UP partitions, while Table 3 presents the recovery of composite partitions and the corresponding predictive performance.

As shown in Tables 2 and 3, DGNN effectively recovers the full ternary structure across all three regimes. It achieves high MP-F1 and AO-F1 scores in R1 and R2, while maintaining strong recovery performance in the more challenging R3. The recovery of AO is accompanied by substantial improvements in predictive performance, with DGNN achieving the highest average R^2 and CCC across all regimes. Additionally, DGNN improves the recovery of the corresponding partitions identified by the baselines in nearly all comparisons, with the sole exception of MP-F1 against DSCCA in R2.

Table 2: Recovery of individual variable partitions across three regimes at $\mathrm{SNR} = 13.0$ dB and $\theta = 0.6$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	MP-F1	AO-F1	UP-F1
R1.SCCA	$0.562_{\pm 0.053}$	–	–
R1.DSCCA	$0.942_{\pm 0.043}$	–	–
R1.Lasso	–	–	$0.996_{\pm 0.012}$
R1.Ours	$\mathbf{0.962_{\pm 0.040}}$	$\mathbf{0.927_{\pm 0.072}}$	$\mathbf{1.000_{\pm 0.000}}$
R2.SCCA	$0.624_{\pm 0.065}$	–	–
R2.DSCCA	$\mathbf{0.905_{\pm 0.052}}$	–	–
R2.Lasso	–	–	$0.995_{\pm 0.014}$
R2.Ours	$0.891_{\pm 0.017}$	$\mathbf{0.838_{\pm 0.020}}$	$\mathbf{1.000_{\pm 0.000}}$
R3.SCCA	$0.583_{\pm 0.057}$	–	–
R3.DSCCA	$0.664_{\pm 0.023}$	–	–
R3.Lasso	–	–	$0.669_{\pm 0.085}$
R3.Ours	$\mathbf{0.878_{\pm 0.060}}$	$\mathbf{0.819_{\pm 0.080}}$	$\mathbf{0.939_{\pm 0.081}}$

Table 3: Recovery of composite variable partitions and predictive performance across three regimes at $\mathrm{SNR} = 13.0$ dB and $\theta = 0.6$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	AO+MP-F1	AO+UP-F1	Avg. R^2	Avg. CCC
R1.SCCA	–	$0.574_{\pm 0.062}$	$0.322_{\pm 0.054}$	$0.374_{\pm 0.053}$
R1.DSCCA	–	$0.941_{\pm 0.040}$	$0.602_{\pm 0.028}$	$0.727_{\pm 0.036}$
R1.Lasso	$0.998_{\pm 0.005}$	–	$0.516_{\pm 0.010}$	$0.593_{\pm 0.009}$
R1.Ours	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{0.966_{\pm 0.033}}$	$\mathbf{0.721_{\pm 0.020}}$	$\mathbf{0.831_{\pm 0.014}}$
R2.SCCA	–	$0.625_{\pm 0.066}$	$0.253_{\pm 0.059}$	$0.331_{\pm 0.074}$
R2.DSCCA	–	$0.887_{\pm 0.060}$	$0.381_{\pm 0.022}$	$0.527_{\pm 0.030}$
R2.Lasso	$0.998_{\pm 0.005}$	–	$0.349_{\pm 0.015}$	$0.456_{\pm 0.015}$
R2.Ours	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{0.911_{\pm 0.012}}$	$\mathbf{0.584_{\pm 0.007}}$	$\mathbf{0.743_{\pm 0.006}}$
R3.SCCA	–	$0.542_{\pm 0.055}$	$0.426_{\pm 0.025}$	$0.543_{\pm 0.031}$
R3.DSCCA	–	$0.584_{\pm 0.040}$	$0.530_{\pm 0.027}$	$0.683_{\pm 0.023}$
R3.Lasso	$0.934_{\pm 0.014}$	–	$0.480_{\pm 0.019}$	$0.626_{\pm 0.021}$
R3.Ours	$\mathbf{0.990_{\pm 0.014}}$	$\mathbf{0.892_{\pm 0.049}}$	$\mathbf{0.721_{\pm 0.017}}$	$\mathbf{0.838_{\pm 0.012}}$

AO intervention test. We next test whether recovered AO variables are functionally necessary for cross-view prediction. After learning and fixing the MP/AO/UP partition, we retrain the same prediction head to predict opposite-view MP variables from the full source view while removing one recovered subset at a time. Unlike the support-based prediction evaluation above, this tests the functional contribution of each recovered role in the original variable space. If AO variables are merely noise or failed MP variables, removing them should have little effect beyond removing random variables; if they provide genuine auxiliary support, their removal should degrade prediction. We compare four interventions: removing the recovered MP, AO, or UP subset, or a size-matched random mixed-role subset.

Table 4 confirms the important utility of AO. Both R^2 and CCC are significantly reduced by removing MP or AO for all three regimes. Removing of UP has no adverse impact and may even improve the R^2 and CCC scores.

Table 4: AO intervention across three regimes at $\mathrm{SNR} = 13.0$ dB. The first column indicates the removed subset. −Rand. removes a same-size random mixed-role subset sampled from the recovered partition without enforcing a single role. Metrics are aggregated over both prediction directions.

Intervention	Avg. CCC			Avg. R^2		
	R1	R2	R3	R1	R2	R3
All	$0.831_{\pm 0.012}$	$0.739_{\pm 0.008}$	$0.801_{\pm 0.042}$	$0.724_{\pm 0.016}$	$0.586_{\pm 0.010}$	$0.688_{\pm 0.046}$
−MP	$0.060_{\pm 0.048}$	$0.180_{\pm 0.040}$	$0.422_{\pm 0.077}$	$0.051_{\pm 0.043}$	$0.140_{\pm 0.034}$	$0.321_{\pm 0.075}$
−AO	$0.686_{\pm 0.028}$	$0.499_{\pm 0.027}$	$0.409_{\pm 0.060}$	$0.552_{\pm 0.019}$	$0.360_{\pm 0.019}$	$0.343_{\pm 0.047}$
−UP	$0.832_{\pm 0.011}$	$0.745_{\pm 0.007}$	$0.801_{\pm 0.052}$	$0.728_{\pm 0.015}$	$0.593_{\pm 0.010}$	$0.701_{\pm 0.058}$
−Rand.	$0.709_{\pm 0.017}$	$0.632_{\pm 0.031}$	$0.686_{\pm 0.058}$	$0.610_{\pm 0.016}$	$0.494_{\pm 0.023}$	$0.573_{\pm 0.059}$

Validation of the relaxation. To directly assess the relaxation, we construct low-dimensional versions of the three synthetic regimes with $p = 6$ and $q = 7$, where exhaustive discrete search is feasible. The low-dimensional role counts differ from those in the main benchmark and are reported in Appendix B.1. For each regime, we compare the exact discrete optimum and the thresholded DGNN solution against the ground-truth MP/AO/UP roles using per-role F1 scores and average assignment error, defined as the fraction of variables assigned to an incorrect role.

Table 5: Ground-truth recovery on low-dimensional synthetic instances with $p = 6, q = 7$ at $\mathrm{SNR} = 13.0$ dB. Avg. Err. denotes the fraction of variables assigned to an incorrect role. All metrics are aggregated over both prediction directions and reported as $\mathrm{mean}_{\pm\mathrm{std}}$ over 10 random seeds.

Regime.Method	MP-F1	AO-F1	UP-F1	Avg. Err.
R1.Exhaustive search	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$1.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
R1.DGNN relaxation	$0.972_{\pm 0.042}$	$0.940_{\pm 0.092}$	$1.000_{\pm 0.000}$	$0.023_{\pm 0.035}$
R2.Exhaustive search	$0.969_{\pm 0.038}$	$0.956_{\pm 0.054}$	$1.000_{\pm 0.000}$	$0.031_{\pm 0.038}$
R2.DGNN relaxation	$0.912_{\pm 0.059}$	$0.884_{\pm 0.069}$	$1.000_{\pm 0.000}$	$0.085_{\pm 0.054}$
R3.Exhaustive search	$0.924_{\pm 0.032}$	$0.852_{\pm 0.075}$	$1.000_{\pm 0.000}$	$0.077_{\pm 0.049}$
R3.DGNN relaxation	$0.884_{\pm 0.032}$	$0.723_{\pm 0.097}$	$0.960_{\pm 0.080}$	$0.138_{\pm 0.046}$

Table 5 validates that DGNN achieves results comparable to exhaustive search in all cases. Note that the exhaustive search does not achieve perfect results. This is likely due to imperfections in the predictor. The gap is largest in R3, where ambiguity-resolution makes the discrete role structure intrinsically harder to recover, but DGNN still preserves the same qualitative decomposition pattern.

4.2 Real-World Datasets

Datasets. We use two paired-view datasets derived from the Alzheimer's Disease Neuroimaging Initiative phase 2 (ADNI-2). The ADAS-CDR dataset consists of 13 Alzheimer's Disease Assessment Scale–Cognitive Subscale (ADAS) and 9 Clinical Dementia Rating (CDR) features. Dataset (qCPs-NB) consists of 26 Embic Corporation Quantified Cognitive Processes (qCPs) and 35 Neuropsychological Battery (NB) features. Details are provided in Appendix B.2.

Cross-view prediction. We evaluate whether the subsets identified from one view can be reliably predicted from the other view. For binary partition baseline methods, we use the selected subset from one view to predict the selected subset in the other view. To ensure a fair comparison, we match the number of selected variables to that of the MP variables discovered by DGNN. Detailed configurations are provided in Appendix B.3.5.

Tables 6 and 7 report the results for both datasets and both prediction directions. Across all settings, DGNN consistently recovers a larger number of variables that can be predicted with high CCC. Additional results are provided in Appendix C.2.2.

Table 6: Cross-system feature prediction on the ADAS-CDR dataset. **Avg.** R^2 denotes the average R^2 across all selected output targets. $\overline{\mathrm{CCC}}{>}\,\theta$ denotes the average CCC among features with CCC exceeding threshold θ, and $\#\mathrm{CCC}{>}\,\theta$ denotes the median number of features with CCC above θ. Best results are bolded, and second-best results are underlined.

Direction	Methods	Avg. R^2	$\overline{\mathrm{CCC}}_{>.5}$	$\overline{\mathrm{CCC}}_{>.7}$	$\#\mathrm{CCC}_{>.5}$	$\#\mathrm{CCC}_{>.7}$
ADAS → CDR	SCCA	$\underline{0.379_{\pm 0.041}}$	$0.709_{\pm 0.034}$	$0.738_{\pm 0.020}$	$\underline{5.00}$	$\underline{3.00}$
	DSCCA	$0.369_{\pm 0.040}$	$0.704_{\pm 0.035}$	$0.738_{\pm 0.020}$	$\underline{5.00}$	$\underline{3.00}$
	Lasso	$0.367_{\pm 0.043}$	$\underline{0.714_{\pm 0.035}}$	$\underline{0.739_{\pm 0.021}}$	4.50	$\underline{3.00}$
	Ours	$\mathbf{0.558_{\pm 0.011}}$	$\mathbf{0.733_{\pm 0.047}}$	$\mathbf{0.757_{\pm 0.014}}$	**7.00**	**5.00**
CDR → ADAS	SCCA	$\underline{0.259_{\pm 0.066}}$	$\underline{0.620_{\pm 0.038}}$	–	$\underline{2.00}$	0.00
	DSCCA	$0.232_{\pm 0.072}$	$0.619_{\pm 0.036}$	–	$\underline{2.00}$	0.00
	Lasso	$0.236_{\pm 0.065}$	$0.617_{\pm 0.040}$	–	$\underline{2.00}$	0.00
	Ours	$\mathbf{0.421_{\pm 0.096}}$	$\mathbf{0.649_{\pm 0.028}}$	$\mathbf{0.759_{\pm 0.040}}$	**4.00**	**1.00**

Classification with feature partitions. We further evaluate the learned variable partitions through downstream classification tasks using an SVM classifier trained on the variable subsets. For baselines, we evaluate both the selected (Sel.) and unselected (Unsel.) subsets. For DGNN, we evaluate MP,

Table 7: Cross-view prediction results on the qCPs–NB dataset. Format and notation are consistent with Table 6.

Direction	Methods	Avg. R^2	$\overline{\text{CCC}}_{>.5}$	$\overline{\text{CCC}}_{>.7}$	$\#\text{CCC}_{>.5}$	$\#\text{CCC}_{>.7}$
qCPs → NB	SCCA	$0.197_{\pm 0.028}$	$0.636_{\pm 0.039}$	–	2.00	0.00
	DSCCA	$0.170_{\pm 0.039}$	$0.630_{\pm 0.043}$	–	1.50	0.00
	Lasso	$0.186_{\pm 0.035}$	$0.632_{\pm 0.043}$	–	2.00	0.00
	Ours	$\mathbf{0.514_{\pm 0.046}}$	$\mathbf{0.683_{\pm 0.029}}$	$\mathbf{0.721_{\pm 0.011}}$	**6.00**	**2.50**
NB → qCPs	SCCA	$0.319_{\pm 0.029}$	$0.632_{\pm 0.127}$	$0.729_{\pm 0.016}$	8.00	1.00
	DSCCA	$0.309_{\pm 0.031}$	$0.626_{\pm 0.016}$	$\mathbf{0.730_{\pm 0.016}}$	8.00	1.00
	Lasso	$0.314_{\pm 0.027}$	$0.630_{\pm 0.011}$	$0.729_{\pm 0.016}$	7.00	1.00
	Ours	$\mathbf{0.448_{\pm 0.046}}$	$\mathbf{0.645_{\pm 0.034}}$	$0.724_{\pm 0.010}$	**15.0**	**3.50**

Table 8: Comparison of feature partitions across methods for downstream classification on two datasets: (a) ADAS-CDR and (b) qCPs-NB.

(a) ADAS–CDR

Method	Setting	ADAS	CDR
All	All	$0.627_{\pm 0.021}$	$0.764_{\pm 0.030}$
SCCA	Unsel	$0.606_{\pm 0.029}$	$0.694_{\pm 0.068}$
	Sel	$0.623_{\pm 0.018}$	$0.765_{\pm 0.034}$
DSCCA	Unsel	$0.583_{\pm 0.041}$	$0.580_{\pm 0.121}$
	Sel	$0.621_{\pm 0.028}$	$0.764_{\pm 0.031}$
Lasso	Unsel	$0.536_{\pm 0.040}$	$0.611_{\pm 0.112}$
	Sel	$0.610_{\pm 0.023}$	$0.757_{\pm 0.028}$
Ours	AO	$0.513_{\pm 0.024}$	$0.453_{\pm 0.025}$
	MP	$0.633_{\pm 0.016}$	$\mathbf{0.768_{\pm 0.028}}$
	AO+MP	$\mathbf{0.639_{\pm 0.020}}$	$0.766_{\pm 0.025}$
	UP	$0.494_{\pm 0.027}$	$0.411_{\pm 0.032}$
	AO+UP	$0.518_{\pm 0.030}$	$0.456_{\pm 0.026}$

(b) qCPs–NB

Method	Setting	qCPs	NB
All	All	$\mathbf{0.590_{\pm 0.016}}$	$0.591_{\pm 0.019}$
SCCA	Unsel	$0.562_{\pm 0.025}$	$0.574_{\pm 0.031}$
	Sel	$0.582_{\pm 0.024}$	$0.576_{\pm 0.028}$
DSCCA	Unsel	$0.563_{\pm 0.038}$	$0.572_{\pm 0.035}$
	Sel	$0.579_{\pm 0.022}$	$0.576_{\pm 0.026}$
Lasso	Unsel	$0.573_{\pm 0.021}$	$0.569_{\pm 0.037}$
	Sel	$0.577_{\pm 0.021}$	$0.576_{\pm 0.021}$
Ours	AO	$0.531_{\pm 0.024}$	$0.573_{\pm 0.036}$
	MP	$0.587_{\pm 0.023}$	$0.577_{\pm 0.023}$
	AO+MP	$0.584_{\pm 0.021}$	$\mathbf{0.605_{\pm 0.030}}$
	UP	$0.530_{\pm 0.047}$	$0.523_{\pm 0.051}$
	AO+UP	$0.547_{\pm 0.041}$	$0.563_{\pm 0.047}$

AO, UP, MP+AO, and UP+AO. Detailed model configurations and training protocols are provided in Appendix B.3.5.

As shown in Table 8, the ternary partitions provided by DGNN offer an effective feature set through MP or MP+AO for better classification of ADAS, CDR, and NB, except for qCPs. However, MP alone achieves classification performance comparable to that obtained using all qCPs features. Importantly, the learned MP/AO relationship is also reflected in downstream classification. As shown in Table 8, when MP achieves the best performance in one view, MP+AO achieves the best performance in the opposite view. This suggests that DGNN can recover the underlying structures of the two views.

Ablation studies and analysis. Table 9 shows the contribution of the DGNN components. Detailed results are reported in Appendix C.2.4. Ablation studies on the *training stages* are in Appendix C.2.5 and *stability analysis of the ternary partitions* are in Appendix C.2.6.

Case study The learned MP/AO/UP partitions are semantically meaningful on the real-world dataset, as supported by the feature-level analysis in Appendix C.2.6.

5 Discussion and Future Work

The proposed Mutual Predictability Decomposition (MPD) framework offers a new perspective on cross-view relationships by separating predictive utility from predictability itself. In particular, *auxiliary* variables represent an important class overlooked by existing shared/private and feature selection approaches: variables that are not themselves predictable across views, yet are essential for predicting mutually predictable variables.

This idea is conceptually related to *synergistic information* in partial information decomposition (PID) [Williams and Beer, 2010, Bertschinger et al., 2014, Makkeh et al., 2021], where variables become informative only in combination with others. Similarly, MPD auxiliary variables may not exhibit

Table 9: Ablation study on model components on the ADAS–CDR dataset. This part evaluates the contribution of each component. Format and notation are consistent with Table 6.

Direction	**Variants**	**Avg.** R^2	$\overline{\mathrm{CCC}}_{>.5}$	$\overline{\mathrm{CCC}}_{>.7}$	$\#\mathrm{CCC}_{>.5}$	$\#\mathrm{CCC}_{>.7}$
ADAS → CDR	no Gate	$-0.268_{\pm 0.219}$	$0.595_{\pm 0.040}$	$0.711_{\pm 0.004}$	6.00	0.00
	no InputGate	$0.481_{\pm 0.187}$	$0.707_{\pm 0.058}$	$0.755_{\pm 0.017}$	**7.00**	**5.00**
	no OutputGate	$-0.231_{\pm 0.400}$	$0.618_{\pm 0.044}$	$0.729_{\pm 0.025}$	6.50	0.50
	Full	$\mathbf{0.558}_{\pm 0.011}$	$\mathbf{0.733}_{\pm 0.047}$	$\mathbf{0.757}_{\pm 0.014}$	**7.00**	**5.00**
CDR → ADAS	no Gate	$-0.041_{\pm 0.145}$	$0.598_{\pm 0.024}$	$0.736_{\pm 0.045}$	4.00	0.00
	no InputGate	$0.328_{\pm 0.127}$	$0.629_{\pm 0.033}$	$\mathbf{0.765}_{\pm 0.023}$	4.00	**1.00**
	no OutputGate	$0.078_{\pm 0.120}$	$0.618_{\pm 0.035}$	$0.735_{\pm 0.026}$	4.00	**1.00**
	Full	$\mathbf{0.421}_{\pm 0.096}$	$\mathbf{0.649}_{\pm 0.028}$	$0.759_{\pm 0.040}$	4.00	**1.00**

strong direct cross-view predictability, but provide contextual information necessary for reconstructing mutually predictable variables. Unlike PID, however, MPD is prediction-driven, formulated as an optimization problem over variable subsets and scalable to high-dimensional continuous settings through differentiable optimization and sparse gating.

More broadly, MPD highlights the distinction between *shared* and *functionally supportive* variables. Traditional shared/private decompositions treat non-shared variables as modality-specific residuals, whereas MPD identifies non-predictable variables that still play an essential role in cross-view prediction. This distinction may be particularly relevant in healthcare and multimodal scientific analysis, where certain measurements provide indirect contextual support rather than direct correspondence across views.

Several limitations and future directions remain. First, the current formulation focuses on pairwise relationships between two variable sets; extending MPD to multi-view or graph-structured settings is a promising direction. Second, although the differentiable relaxation enables scalable optimization, theoretical analysis of identifiability and consistency remains open. Finally, strengthening the connection between MPD and information-theoretic synergy may provide deeper insight into the auxiliary subset and predictive structure discovery.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their valuable comments and suggestions.

The authors received no external funding specifically for this work.

The ADNI data used in this study were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI). Data collection and sharing for ADNI is funded by the National Institute on Aging (National Institutes of Health Grant U19AG024904). The grantee organization is the Northern California Institute for Research and Education. In the past, ADNI has also received funding from the National Institute of Biomedical Imaging and Bioengineering, the Canadian Institutes of Health Research, and private-sector contributions through the Foundation for the National Institutes of Health (FNIH), including contributions from AbbVie, Alzheimer's Association, Alzheimer's Drug Discovery Foundation, Araclon Biotech, BioClinica, Inc., Biogen, Bristol-Myers Squibb Company, CereSpir, Inc., Cogstate, Eisai Inc., Elan Pharmaceuticals, Inc., Eli Lilly and Company, EuroImmun, F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc., Fujirebio, GE Healthcare, IXICO Ltd., Janssen Alzheimer Immunotherapy Research & Development, LLC., Johnson & Johnson Pharmaceutical Research & Development LLC., Lumosity, Lundbeck, Merck & Co., Inc., Meso Scale Diagnostics, LLC., NeuroRx Research, Neurotrack Technologies, Novartis Pharmaceuticals Corporation, Pfizer Inc., Piramal Imaging, Servier, Takeda Pharmaceutical Company, and Transition Therapeutics.

The authors declare no competing interests.

References

Shotaro Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS)*, 2006.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1247–1255, 2013.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.

Antonin Dauvin, Carolina Donado, Patrik Bachtiger, Ke-Chun Huang, Christopher Martin Sauer, Daniele Ramazzotti, Matteo Bonvini, Leo Anthony Celi, and Molly J Douglas. Machine learning can accurately predict pre-admission baseline hemoglobin and creatinine in intensive care patients. *NPJ digital medicine*, 2(1):116, 2019.

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pages 162–190. Springer, 1992.

Felix Krones, Umar Marikkar, Guy Parsons, Adam Szmul, and Adam Mahdi. Review of multimodal machine learning approaches in healthcare. *Information Fusion*, 114:102690, 2025.

Yongchun Li, Santanu S Dey, and Weijun Xie. On sparse canonical correlation analysis. *Advances in Neural Information Processing Systems*, 37:10707–10734, 2024.

Yue Li, Pratheeksha Nair, Xing Han Lu, Zhi Wen, Yuening Wang, Amir Ardalan Kalantari Dehaghi, Yan Miao, Weiqi Liu, Tamas Ordog, Joanna M Biernacka, et al. Inferring multimodal latent topics from electronic health records. *Nature communications*, 11(1):2536, 2020.

Ofir Lindenbaum, Moshe Salhov, Amir Averbuch, and Yuval Kluger. L0-sparse canonical correlation analysis. In *International conference on learning representations*, 2021.

Xiaohao Liu, Xiaobo Xia, See-Kiong Ng, and Tat-Seng Chua. Principled multimodal representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Zhixuan Liu and Wei-Shi Zheng. Learning multimodal relationship interaction for visual relationship detection. *Pattern Recognition*, 132:108848, 2022.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

Ziping Ma, Furong Xu, Jian Liu, Ming Yang, and Qingpei Guo. Sycoca: symmetrizing contrastive captioners with attentive masking for multimodal alignment. In *Proceedings of the 41st International Conference on Machine Learning*, pages 34038–34052, 2024.

Ahmad Makkeh, Dirk O. Theis, and Raul Vicente. Broja-2pid: A robust estimator for bivariate partial information decomposition. *Entropy*, 23(6):706, 2021.

Francesco Montagna, Nicoletta Noceti, Lorenzo Rosasco, Kun Zhang, and Francesco Locatello. Causal discovery with score matching on additive models with arbitrary noise. In *Conference on Causal Learning and Reasoning*, pages 726–751. PMLR, 2023.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

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

Oliver E Richardson, Joseph Y Halpern, and Christopher De Sa. Inference for probabilistic dependency graphs. In *Uncertainty in Artificial Intelligence*, pages 1741–1751. PMLR, 2023.

Dipti Theng and Kishor K Bhoyar. Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge and Information Systems*, 66(3):1575–1637, 2024.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Enayat Ullah and Raman Arora. Generalization bounds for kernel canonical correlation analysis. *Transactions on Machine Learning Research*, 2023.

Zehan Wang, Yang Zhao, Haifeng Huang, Jiageng Liu, Aoxiong Yin, Li Tang, Linjun Li, Yongqi Wang, Ziang Zhang, and Zhou Zhao. Connecting multi-modal contrastive representations. *Advances in Neural Information Processing Systems*, 36:22099–22114, 2023.

David Weenink. Canonical correlation analysis. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, volume 25, pages 81–99. University of Amsterdam Amsterdam, 2003.

Paul L. Williams and Randall D. Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10 (3):515–534, 2009.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International conference on machine learning*, pages 10648–10659. PMLR, 2020.

Dingkang Yang, Shuai Huang, Haopeng Kuang, Yangtao Du, and Lihua Zhang. Disentangled representation learning for multimodal emotion recognition. In *Proceedings of the 30th ACM international conference on multimedia*, pages 1642–1651, 2022.

Kang Yin, Hye-Bin Shin, Dan Li, and Seong-Whan Lee. Eeg-based multimodal representation learning for emotion recognition. In *2025 13th International Conference on Brain-Computer Interface (BCI)*, pages 1–4. IEEE, 2025.

Xiaohui Zhang, Jaehong Yoon, Mohit Bansal, and Huaxiu Yao. Multimodal representation learning by alternating unimodal adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27456–27466, 2024.

Yuanhao Zhuo, Zixi Qin, Ling Qin, and Wanqing Li. Bfhd: Bidirectional feature harmonization decomposition for heterogeneous clinical assessments. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 7046–7054. International Joint Conferences on Artificial Intelligence Organization, 2026. doi: 10.24963/ijcai.2026/784.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

A Theoretical Analysis

This section provides the proof of Proposition 1 and Theorem 1, followed by partial analysis under idealized assumptions.

The following proposition formalizes that Eqs. (2)– (4) constitute a differentiable relaxation of Eq. (1).

Proposition 1(Differentiable relaxation of Eq. (1)) Let s^X, s^Y be continuous surrogates of the auxiliary-support indicators for $\mathcal{I}_X, \mathcal{I}_Y$, and let h^Y, h^X be continuous surrogates of the membership indicators for Y_{MP}, X_{MP}. Then the optimization problem in Eqs. (2)–(4) is a differentiable relaxation of Eq. (1): the cardinality terms $|Y_{MP}|$ and $|X_{MP}|$ are replaced by soft coverage rewards $\sum_j h_j^Y$ and $\sum_i h_i^X$, while the minimal-support constraints on $\mathcal{I}_X$ and $\mathcal{I}_Y$ are replaced by sparsity regularizers on s^X and s^Y.

Proof. Consider the directional problem $X \to Y$ in Eq. (1). In the original formulation, the feasible solution is specified by two discrete objects: a support set $\mathcal{I}_X \subseteq X$ and a predictable subset $Y_{MP} \subseteq Y$. Their binary membership indicators are $\mathbf{1}\{X_d \in \mathcal{I}_X\}$ and $\mathbf{1}\{Y_j \in Y_{MP}\}$, respectively. Replacing these binary variables by continuous gates $s_d^X \in [0,1]$ and $h_j^Y \in [0,1]$ yields a continuous parameterization of the same search space.

Next, the cardinality objective $|Y_{MP}|$ is equivalently written as $\sum_{j=1}^{q} \mathbf{1}\{Y_j \in Y_{MP}\}$. Its continuous surrogate is $\sum_{j=1}^{q} h_j^Y$, which is maximized through the term $-\lambda_{out} G_{incentive}(h^Y)$. Likewise, the minimality of $\mathcal{I}_X$ is captured by a sparsity penalty on the relaxed support variables s^X, realized by $R_{in}(s^X)$ in Eqs. (2). Finally, prediction quality is enforced through the gated reconstruction term $\sum_j h_j^Y \mathrm{MSE}_j$, which ensures that only variables with small prediction error are favored to receive large membership values.

The same argument applies symmetrically to $Y \to X$. Therefore, Eqs. (2)–(4) replace all discrete set variables in Eq. (1) by continuous differentiable surrogates while preserving the original optimization logic, and hence form a differentiable relaxation of Eq. (1). □

The next theorem shows that the discrete formulation is recovered when the relaxed gates become binary.

Restatement of Theorem 1. *Assume that at convergence the relaxed gates satisfy $s_d^X, s_d^Y, h_i^X, h_j^Y \in \{0,1\}$ for all dimensions. Then any stationary point of Eq. (4) induces a feasible discrete solution to Eq. (1) via*

$$\mathcal{I}_X = \{X_d : s_d^X = 1\}, \qquad Y_{MP} = \{Y_j : h_j^Y = 1\},$$

and analogously

$$\mathcal{I}_Y = \{Y_d : s_d^Y = 1\}, \qquad X_{MP} = \{X_i : h_i^X = 1\}.$$

Moreover, under this binary limit, the relaxed objective reduces to the discrete objective up to the surrogate approximation used for the error constraint.

Proof. When the gates are binary, the relaxed supports s^X and s^Y specify exact subsets of input variables, and the output gates h^Y and h^X specify exact membership in Y_{MP} and X_{MP}. The gated prediction losses then become ordinary losses restricted to the selected mutually predictable variables. The regularizer R_{in} counts the selected auxiliary variables in the same role as the support-size control in Eq. (1), while the incentive term counts the number of selected predictable variables. Therefore, the relaxed problem collapses to a discrete subset-selection problem over the same objects as Eq. (1), yielding a feasible discrete decomposition. □

A.1 Identifiability, Recovery, and Approximation Guarantees

We next provide partial theoretical analysis under idealized assumptions. These results give sufficient conditions under which the MP/AO/UP decomposition is well-defined at the population level, recoverable in a sparse linear regime, and approximated by the relaxed objective when the gates are near-binary. They should be interpreted as preliminary guarantees rather than a complete theory of identifiability and consistency for the general nonlinear DGNN.

Throughout, let

$$X \in \mathbb{R}^p, \qquad Y \in \mathbb{R}^q,$$

denote the two variable groups in Eq. (1). Recall that the goal is to partition each side into three disjoint subsets:

$$X = X_{\mathrm{MP}} \cup X_{\mathrm{AO}} \cup X_{\mathrm{UP}}, \qquad Y = Y_{\mathrm{MP}} \cup Y_{\mathrm{AO}} \cup Y_{\mathrm{UP}},$$

where MP denotes mutually predictable variables, AO denotes auxiliary-only variables, and UP denotes unpredictable variables.

Population prediction errors. For any support set $S \subseteq [p]$, define the optimal population prediction error for target coordinate Y_j as

$$\mathcal{E}_{Y,j}(S) := \inf_{f_j \in \mathcal{F}_{X \to Y}} \mathbb{E}[\ell(Y_j, f_j(X_S))] \,.$$

Likewise, for any $T \subseteq [q]$, define

$$\mathcal{E}_{X,i}(T) := \inf_{g_i \in \mathcal{F}_{Y \to X}} \mathbb{E}[\ell(X_i, g_i(Y_T))] \,.$$

A variable is deemed mutually predictable if its minimum achievable prediction error falls below the tolerance threshold in Eq. (1).

Shared assumptions. We collect the regularity conditions used in Theorems 2–4 below. The assumptions are standard in support recovery and continuous relaxation analysis. Each theorem invokes only the subset of assumptions explicitly referenced.

Assumption 1 (Unique minimal supports)**.** *For each predictable target Y_j, there exists a unique minimum-cardinality support*

$$S_j^\star = \arg\min_{S \subseteq [p]} \{|S| : \mathcal{E}_{Y,j}(S) \le \varepsilon_Y\} \,.$$

Similarly, for each predictable X_i, there exists a unique minimum-cardinality support

$$T_i^\star = \arg\min_{T \subseteq [q]} \{|T| : \mathcal{E}_{X,i}(T) \le \varepsilon_X\} \,.$$

Assumption 2 (Prediction margin)**.** *There exist constants $\gamma_Y, \gamma_X > 0$ such that for every $j \in [q]$, either*

$$\mathcal{E}_{Y,j}(S_j^\star) \le \varepsilon_Y - \gamma_Y,$$

or else

$$\mathcal{E}_{Y,j}(S) \ge \varepsilon_Y + \gamma_Y \quad \forall S \subseteq [p].$$

An analogous condition holds for every $i \in [p]$ with threshold ε_X.

Assumption 3 (Non-redundancy of selected support variables)**.** *Every selected auxiliary variable is necessary: for all $d \in S_j^\star$,*

$$\mathcal{E}_{Y,j}(S_j^\star \setminus \{d\}) > \varepsilon_Y,$$

and for all $r \in T_i^\star$,

$$\mathcal{E}_{X,i}(T_i^\star \setminus \{r\}) > \varepsilon_X.$$

Assumption 4 (Sparse linear recovery regime)**.** *For Theorem 3, assume*

$$Y = XB^\star + W_Y, \qquad X = YC^\star + W_X,$$

where W_Y, W_X are sub-Gaussian noises, and the design satisfies standard restricted-eigenvalue and incoherence conditions. Further, the active rows of $B^\star, C^\star$ satisfy a beta-min condition.

Assumption 5 (Near-binary gates)**.** *For Theorem 4, the learned gates s^X, s^Y, h^X, h^Y satisfy*

$$\max_k \min\{z_k, 1 - z_k\} \le \delta$$

for every gate coordinate z_k, with $\delta \in (0, 1/2)$.

Population decomposition. Define the predictable sets

$$Y_{\mathrm{MP}}^\star := \{j \in [q] : \exists S \subseteq [p],\ \mathcal{E}_{Y,j}(S) \le \varepsilon_Y\},$$
$$X_{\mathrm{MP}}^\star := \{i \in [p] : \exists T \subseteq [q],\ \mathcal{E}_{X,i}(T) \le \varepsilon_X\}.$$

Define the directional support unions

$$\mathcal{I}_X^\star := \bigcup_{j \in Y_{\mathrm{MP}}^\star} S_j^\star, \qquad \mathcal{I}_Y^\star := \bigcup_{i \in X_{\mathrm{MP}}^\star} T_i^\star.$$

Then

$$X_{\mathrm{AO}}^\star := \mathcal{I}_X^\star \setminus X_{\mathrm{MP}}^\star, \qquad X_{\mathrm{UP}}^\star := [p] \setminus (\mathcal{I}_X^\star \cup X_{\mathrm{MP}}^\star),$$

and analogously for Y.

Theorem 2 (Identifiability of the MP/AO/UP decomposition)**.** *Under Assumptions 1–3, the population ternary partition*

$$(X^\star_{\mathrm{MP}}, X^\star_{\mathrm{AO}}, X^\star_{\mathrm{UP}}, Y^\star_{\mathrm{MP}}, Y^\star_{\mathrm{AO}}, Y^\star_{\mathrm{UP}})$$

is uniquely determined by the coordinate-wise population MPD formulation.

Proof. By Assumption 2, no coordinate is on the predictability boundary, so $X^\star_{\mathrm{MP}}$ and $Y^\star_{\mathrm{MP}}$ are uniquely determined by the population risks. Given these MP sets, Assumption 1 gives a unique minimal support for each predictable coordinate; hence the support unions $\mathcal{I}^\star_X$ and $\mathcal{I}^\star_Y$ are unique. Assumption 3 rules out redundant variables in these supports. Finally,

$$X^\star_{\mathrm{AO}} = \mathcal{I}^\star_X \setminus X^\star_{\mathrm{MP}}, \qquad X^\star_{\mathrm{UP}} = X \setminus (\mathcal{I}^\star_X \cup X^\star_{\mathrm{MP}}),$$

with the analogous definitions for Y. Therefore all six sets in the ternary partition are uniquely determined. □

Theorem 3 (Exact recovery in an idealized sparse linear regime)**.** *Under Assumptions 2 and 4, consider an idealized sparse-linear analogue of the full relaxed objective in Eq. (4). Let $\widehat{B}, \widehat{C}$ be the two directional group-sparse estimators, and let*

$$\widehat{\mathcal{I}_X} = \{d : \|\widehat{B}_{d\cdot}\|_2 > 0\}, \qquad \widehat{\mathcal{I}_Y} = \{r : \|\widehat{C}_{r\cdot}\|_2 > 0\}.$$

Let $\widehat{\mathcal{E}}_{Y,j}, \widehat{\mathcal{E}}_{X,i}$ be held-out prediction errors, and define

$$\widehat{Y}_{\mathrm{MP}} = \{j : \widehat{\mathcal{E}}_{Y,j} \le \varepsilon_Y\}, \qquad \widehat{X}_{\mathrm{MP}} = \{i : \widehat{\mathcal{E}}_{X,i} \le \varepsilon_X\}.$$

If

$$n \ge C \max\{s_X \log p,\ s_Y \log q\},$$

where $s_X = |\mathcal{I}^\star_X|$, $s_Y = |\mathcal{I}^\star_Y|$, then with probability at least $1 - \exp(-cn)$,

$$\widehat{\mathcal{I}_X} = \mathcal{I}^\star_X, \qquad \widehat{\mathcal{I}_Y} = \mathcal{I}^\star_Y,$$

$$\widehat{X}_{\mathrm{MP}} = X^\star_{\mathrm{MP}}, \qquad \widehat{Y}_{\mathrm{MP}} = Y^\star_{\mathrm{MP}}.$$

Consequently,

$$(\widehat{X}_{\mathrm{AO}}, \widehat{X}_{\mathrm{UP}}, \widehat{Y}_{\mathrm{AO}}, \widehat{Y}_{\mathrm{UP}}) = (X^\star_{\mathrm{AO}}, X^\star_{\mathrm{UP}}, Y^\star_{\mathrm{AO}}, Y^\star_{\mathrm{UP}}).$$

Proof. We prove the result for the direction $X \to Y$; the reverse direction is identical. Under the sparse linear assumptions, standard support recovery results for group-sparse multi-output regression imply that, with probability at least $1 - \exp(-cn)$, the estimated support equals the population essential support:

$$\widehat{\mathcal{I}}_X = \mathcal{I}^\star_X.$$

The same argument gives $\widehat{\mathcal{I}}_Y = \mathcal{I}^\star_Y$ for the reverse direction.

It remains to show that the MP sets are recovered. By the Assumptions 2, every $Y_j \in Y^\star_{\mathrm{MP}}$ has population prediction error at most $\varepsilon_Y - \gamma_Y$ from the essential support, whereas every $Y_j \notin Y^\star_{\mathrm{MP}}$ has prediction error at least $\varepsilon_Y + \gamma_Y$ from any support. Since the empirical validation error is uniformly close to the corresponding population error, the thresholding rule at ε_Y selects exactly the variables in $Y^\star_{\mathrm{MP}}$. Hence

$$\widehat{Y}_{\mathrm{MP}} = Y^\star_{\mathrm{MP}}.$$

The same argument gives

$$\widehat{X}_{\mathrm{MP}} = X^\star_{\mathrm{MP}}.$$

Finally, the AO and UP sets are defined deterministically from the recovered MP sets and essential supports:

$$X_{\mathrm{AO}} = \mathcal{I}_X \setminus X_{\mathrm{MP}}, \qquad X_{\mathrm{UP}} = X \setminus (\mathcal{I}_X \cup X_{\mathrm{MP}}),$$

and analogously for Y. Therefore exact recovery of $\mathcal{I}_X, \mathcal{I}_Y, X_{\mathrm{MP}}, Y_{\mathrm{MP}}$ immediately implies exact recovery of the full MP/AO/UP decomposition. Combining the two directions with a union bound gives the stated high-probability result. This theorem establishes finite-sample recovery for the idealized sparse-linear analogue considered above, rather than a general recovery guarantee for the nonlinear DGNN objective in Eq. (4). □

Theorem 4 (Approximation of the discrete MPD surrogate by the relaxed objective)**.** *Under Assumption 5, let*

$$(\bar{s}^X, \bar{s}^Y, \bar{h}^X, \bar{h}^Y)$$

be the binary thresholding of the learned gates at $1/2$*. Let* $\mathcal{L}^{\text{relax}}$ *denote Eq. (4), and let* $\mathcal{L}^{\text{disc}}$ *denote the penalized discrete counterpart of Eqs. (2)–(4) obtained by replacing each continuous gate with its thresholded binary indicator.*

If the predictors, loss, regularizer, and incentive terms are Lipschitz on the relevant bounded domains, then there exists a constant $C > 0$ *such that*

$$\left|\mathcal{L}^{\text{relax}} - \mathcal{L}^{\text{disc}}\right| \leq C\delta.$$

Hence, as $\delta \to 0$*, the relaxed objective converges uniformly to its penalized discrete counterpart.*

Proof. Let $\bar{s}^X, \bar{s}^Y, \bar{h}^X, \bar{h}^Y$ be the binary gates obtained by thresholding the relaxed gates at $1/2$. Under Assumption 5, each relaxed gate is within δ of its thresholded binary value. Therefore, for the direction $X \to Y$,

$$\|s^X \odot X - \bar{s}^X \odot X\| \leq C_X\delta$$

on the bounded data domain. Since the predictor and loss are Lipschitz in the gated input, the per-coordinate prediction losses under s^X and $\bar{s}^X$ differ by at most $O(\delta)$. In addition,

$$|h_j^Y - \bar{h}_j^Y| \leq \delta$$

for every output coordinate, so replacing the soft output weights by their binary counterparts changes the gated prediction loss by at most $O(\delta)$.

The same argument applies to the regularization and incentive terms, since R_{in} and $G_{incentive}$ are continuous, and in particular Lipschitz, on the compact gate domain $[0, 1]$. Hence

$$\left|\mathcal{L}^{\text{relax}}_{X\to Y} - \mathcal{L}^{\text{binary}}_{X\to Y}\right| \leq C_1\delta$$

for some constant $C_1 > 0$. Applying the same reasoning to the reverse direction gives

$$\left|\mathcal{L}^{\text{relax}}_{Y\to X} - \mathcal{L}^{\text{binary}}_{Y\to X}\right| \leq C_2\delta.$$

Summing the two bounds yields

$$\left|\mathcal{L}^{\text{relax}} - \mathcal{L}^{\text{binary}}\right| \leq C\delta.$$

By definition, $\mathcal{L}^{\text{binary}}$ is the penalized discrete counterpart $\mathcal{L}^{\text{disc}}$: the input gates specify the discrete supports $\mathcal{I}_X, \mathcal{I}_Y$, the output gates specify the discrete candidate sets $X_{\text{MP}}, Y_{\text{MP}}$, the sparsity terms penalize selected supports, and the incentive terms reward selected predictable coordinates. Therefore, $\mathcal{L}^{\text{relax}}$ converges uniformly to $\mathcal{L}^{\text{disc}}$ as the relaxed gates become near-binary. □

Discussion. Theorems 2–4 provide partial guarantees under idealized assumptions. Theorem 2 gives sufficient conditions under which the population MP/AO/UP decomposition is well-defined. Theorem 3 establishes finite-sample recovery for an idealized sparse-linear analogue of the relaxed MPD objective. Theorem 4 shows that, when the learned gates are near-binary, the relaxed objective is close to its penalized discrete counterpart. These results do not constitute a complete identifiability or consistency theory for the general nonlinear DGNN setting, which remains an important direction for future work.

B Implementation Details

B.1 Synthetic Data Generation

We generate paired synthetic views

$$X \in \mathbb{R}^{n\times p}, \qquad Y \in \mathbb{R}^{n\times q},$$

with known ground-truth MP, AO, and UP roles. The data are generated directly in the original variable space rather than through latent factors, so that the ground-truth roles correspond to observed variables. Each dataset is assembled from directional cross-view components. For a component in the direction $X \to Y$, source-side MP variables are placed in X, while the corresponding target-side MP and, when needed, AO variables are placed in Y; components in the direction $Y \to X$ are generated analogously. After all informative variables are generated, independent nuisance variables are added as UP variables to match the desired dimensionalities. The columns of both views are then randomly permuted, while the ground-truth role indices are retained.

Role quotas and noise. For each regime, we specify the target numbers of MP and AO variables on both sides,

$$(|X_{\mathrm{MP}}|, |X_{\mathrm{AO}}|, |Y_{\mathrm{MP}}|, |Y_{\mathrm{AO}}|),$$

and assign the remaining variables to UP. The generator uses a quota scheduler to assemble directional components such that these role counts are matched exactly. Independent Gaussian observation noise is added after the noiseless variables are generated. For each noiseless variable v, the noise variance is chosen according to the desired signal-to-noise ratio,

$$\mathrm{SNR} = \frac{\mathrm{Var}(v)}{\mathrm{Var}(\epsilon)}, \qquad v_{\mathrm{obs}} = v + \epsilon.$$

Unless otherwise stated, the same SNR is used for MP, AO, and UP variables.

R1: linear persistent-support AO. R1 tests whether the method can identify AO variables in a setting where the cross-view relationships are predominantly linear. The construction combines two types of components. Some MP variables are related across views by direct affine transformations, so they are mutually predictable without auxiliary support. Other target-side MP variables depend on both a source-side MP variable and an additional private support variable. This private support variable is observed in the target view and is labeled as AO: it is useful for predicting the opposite-view MP variable, but it is not itself intended to be mutually predictable from the opposite view. Thus, AO variables in this regime arise from persistent linear support rather than from nonlinear ambiguity.

Example: For independent $z, u \sim \mathcal{N}(0, 1)$,

$$x^{\mathrm{MP}} = z, \qquad y^{\mathrm{MP}} = \alpha z + \beta u + b, \qquad y^{\mathrm{AO}} = u.$$

Here, y^{AO} is required to recover

$$x^{\mathrm{MP}} = \frac{y^{\mathrm{MP}} - \beta y^{\mathrm{AO}} - b}{\alpha},$$

but is not itself predictable from X.

The full datasets are constructed by concatenating directional components in both orientations ($X \to Y$ and $Y \to X$), adding independent Gaussian UP variables and SNR-controlled observation noise, and randomly permuting the columns within each view. R2 and R3 follow the same dataset assembly procedure, using their respective component constructions described below.

R2: nonlinear persistent-support AO. R2 preserves the same role logic as R1, but replaces the linear dependence on auxiliary support with nonlinear transformations. Target-side MP variables are generated from a combination of shared MP variables and private AO variables passed through nonlinear functions, such as saturating nonlinearities. The AO variables therefore provide stable predictive support for the opposite-view MP variables, but the mapping is no longer well described by a linear model. This regime evaluates whether the DGNN can recover AO variables when the auxiliary contribution is nonlinear and cannot be captured by purely linear sparse association methods.

Example: Given

$$x^{\mathrm{MP}} = z, \qquad y^{\mathrm{MP}} = \tanh(\alpha z) + \beta \tanh(\kappa u), \qquad y^{\mathrm{AO}} = u.$$

The AO variable provides nonlinear auxiliary support for recovering the opposite-view MP variable.

R3: ambiguity-resolution AO. R3 is designed to test the central distinction between mutually predictable variables and auxiliary-only variables. In this regime, MP variables are generated through many-to-one transformations, so the target-side MP variables preserve predictable low-dimensional quantities but discard branch information needed to reconstruct the source-side variables. AO variables are then generated to carry precisely this missing branch or sign information. Thus, AO variables are not arbitrary noise variables: they are mathematically constructed to resolve ambiguities induced by the MP-generating map.

Example: For $z \sim \mathcal{N}(0,1)$ and independent $r > 0$,

$$x^{\mathrm{MP}} = z, \qquad y^{\mathrm{MP}} = |z|, \qquad y^{\mathrm{AO}} = \mathrm{sign}(z)r,$$

so that

$$x^{\mathrm{MP}} = \mathrm{sign}(y^{\mathrm{AO}})y^{\mathrm{MP}}.$$

The AO variable resolves the sign ambiguity while remaining non-MP.

Concretely, the generator samples source variables and applies randomly parameterized many-to-one functions to produce target-side MP variables. These functions include absolute-value transformations, parabola-based mappings, and conic-section mappings such as ellipse- and hyperbola-type quadratic forms. For example, a source variable may be mapped to a target MP variable through a quadratic form, which collapses multiple source values to the same target value. The associated AO variables encode branch information through the sign of the corresponding source quantity, such as the sign relative to a parabola vertex or the signs of coordinates involved in a conic-section mapping. Importantly, this branch signal is not inserted as a clean binary indicator. It is multiplied by an independent random component, producing a continuous AO signal of the form $\mathrm{sign}(\cdot)\, r$. This continuous AO signal is then treated as the noiseless AO variable and receives the same SNR-controlled observation noise as the other variables. Random coefficients, shifts, and conic types are sampled for each component, so the resulting dataset contains heterogeneous ambiguity patterns rather than repeated copies of a single formula.

This construction creates a setting in which MP variables are genuinely predictable across views, but accurate cross-view prediction of the full MP structure requires auxiliary variables that are not themselves MP variables. Therefore, R3 directly tests whether a method can distinguish variables that are predictable from variables that are useful for resolving predictability.

Configurations used in the experiments. For the main recovery and intervention experiments, we use $n = 5000$ samples with $p = 21$ variables in X and $q = 19$ variables in Y. The role counts are:

Table 10: Synthetic configurations used in the main recovery and intervention experiments.

Regime	$\lvert X_{\mathrm{MP}}\rvert$	$\lvert X_{\mathrm{AO}}\rvert$	$\lvert X_{\mathrm{UP}}\rvert$	$\lvert Y_{\mathrm{MP}}\rvert$	$\lvert Y_{\mathrm{AO}}\rvert$	$\lvert Y_{\mathrm{UP}}\rvert$
R1	10	4	7	10	4	5
R2	10	5	6	10	5	4
R3	11	6	4	10	6	3

For Validation of the relaxed objective experiment, we use $n = 5000$ samples with $p = 6$ variables in X and $q = 7$ variables in Y. The role counts are:

Table 11: Synthetic configurations used in the relaxation-validation experiment.

Regime	$\lvert X_{\mathrm{MP}}\rvert$	$\lvert X_{\mathrm{AO}}\rvert$	$\lvert X_{\mathrm{UP}}\rvert$	$\lvert Y_{\mathrm{MP}}\rvert$	$\lvert Y_{\mathrm{AO}}\rvert$	$\lvert Y_{\mathrm{UP}}\rvert$
R1	3	1	2	3	1	3
R2	3	2	1	4	2	1
R3	3	2	1	3	2	2

The results in the main text are reported at SNR $= 13$ dB unless otherwise stated. For the SNR sweep, we vary the SNR while keeping the same role counts and generation mechanism. For the relaxation-validation experiment, we use lower-dimensional variants of the same regimes so that exact combinatorial search is feasible.

B.2 Real-World Datasets and Preprocessing

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). ADNI data can be accessed by qualified researchers through application at the official ADNI website. Specifically, we use two paired-view datasets derived from the Alzheimer's Disease Neuroimaging Initiative Phase 2 (ADNI-2). The downstream classification task in this study is formulated as a three-class classification problem, including cognitively normal (CN), mild cognitive impairment (MCI), and Alzheimer's disease (AD).

In the ADAS-CDR dataset, the ADAS view corresponds to a structured cognitive assessment widely used in Alzheimer's disease studies. It measures core cognitive abilities affected by dementia, including memory, language, orientation, and praxis, through a combination of subject-completed tasks and examiner-rated items. The CDR view, in contrast, provides a global clinical characterization of dementia severity by jointly assessing cognitive symptoms and functional impairment. Unlike ADAS, which mainly quantifies cognitive performance through structured test items, CDR further evaluates how cognitive decline affects everyday functioning across domains such as memory, orientation, judgment and problem solving, community affairs, home and hobbies, and personal care. For both assessments, we retained only the earliest Phase 2 assessment record for each participant. A paired instance was constructed only when the ADAS and CDR assessments were conducted within 30 days of each other and both views contained no missing values. Under these criteria, the final dataset consisted of 1,158 paired instances.

In the qCPs-NB dataset, the qCPs view consists of model-derived cognitive process measures that provide fine-grained characterizations of memory and verbal fluency. Therefore, qCPs offer detailed measurements for a focused subset of cognitive functions. The NB view consists of standard neuropsychological task-level assessments collected in ADNI and provides broader coverage across multiple cognitive domains, including memory, executive function, attention, language, and visuospatial or constructional ability. This dataset therefore contrasts fine-grained process-level cognitive measures with broader task-level neuropsychological assessments. Following the same preprocessing strategy used for the ADAS-CDR dataset, we retained only the earliest Phase 2 assessment record for each participant ID in both views. A paired instance was constructed only when the qCPs and NB assessments were conducted within 30 days of each other. Under these criteria, the final dataset consisted of 1,088 paired instances.

Table 12: Summary of the ADAS-CDR and qCPs-NB paired-view datasets. The left and right subtables correspond to the ADAS-CDR and qCPs-NB datasets, respectively. Each subtable reports the number of features for each assessment view and the sample distribution across CN, MCI and AD classes.

ADAS-CDR dataset		
Number of features	ADAS	13
	CDR	9
Diagnostic distribution	MCI	459
	AD	399
	CN	300

qCPs-NB dataset		
Number of features	qCPs	26
	NB	35
Diagnostic distribution	MCI	446
	AD	346
	CN	296

B.3 Implementation Details

B.3.1 Common Implementation Details

Data splits and random seeds. All experiments use a $70/15/15$ train/validation/test split unless otherwise stated. Unless otherwise stated, all experiments are repeated over 10 random seeds. For each random seed, all methods are evaluated on the same split, and hyperparameters and early-stopping decisions are selected using validation data only. The aggregation format, either mean

(standard deviation) or median (interquartile range), is specified in the corresponding table or figure caption.

Evaluation metrics. For prediction quality, we report the average R^2 and the average concordance correlation coefficient (CCC) over the evaluated target variables. CCC is defined as

$$\mathrm{CCC}(y, \hat{y}) = \frac{2\,\mathrm{Cov}(y, \hat{y})}{\mathrm{Var}(y) + \mathrm{Var}(\hat{y}) + (\mathbb{E}[y] - \mathbb{E}[\hat{y}])^2},$$

CCC penalizes both poor correlation and mean/variance mismatch. For relaxation validation, normalized Hamming distance is the fraction of variables whose recovered MP/AO/UP label differs from the exact discrete solution. All other metrics are computed using their standard definitions.

Compute environment. All experiments were run on a desktop workstation with an NVIDIA RTX 4080 SUPER GPU and an Intel Core i7-14700K CPU. Unless otherwise stated, GPU acceleration was enabled for all experiments.

B.3.2 Architecture of DGNN

The proposed DGNN consists of three main components: a stochastic input feature selector (STG), a multilayer perceptron (MLP) for multi-target regression, and a hard-concrete output gate (HCG) for target selection. Each input feature passes through a learnable stochastic gate, allowing the model to suppress irrelevant features and focus on informative inputs. The selected features are then fed into the MLP, which predicts all target variables. On the output side, each target prediction passes through a learnable hard gate, enabling the model to automatically select targets that are reliably predictable from the input. The output gates direct training toward predictable targets. By jointly optimizing the input gates, output gates, and MLP predictor, our approach performs feature selection and target selection simultaneously, achieving both input sparsity and maximal coverage of predictable targets.

Stochastic Input Feature Selector (STG). Selecting informative input features is essential for robust multi-target prediction. The STG [Yamada et al., 2020] enables differentiable and sparse feature selection through a Gaussian relaxation of the Bernoulli distribution, allowing efficient gradient-based optimization. For each input feature x_d, the STG defines a corresponding stochastic gate. The gate is parameterized by a learnable mean μ_d and incorporates additive Gaussian noise with standard deviation σ, resulting in a stochastic gate value. To ensure that the gate output remains within $[0, 1]$, the sum of μ_d and the noise is clipped to this interval.

To promote sparsity, we add a regularization term based on the expected ℓ_0 norm of the gates:

$$\mathcal{R}_{in}(\mu) = \sum_{d=1}^{D} \Phi\left(-\frac{\mu_d}{\sigma}\right), \tag{5}$$

where $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function. This term approximates the probability that each gate $g_{in,d}$ is active (> 0).

Hard Concrete Output Gate (HCG) The HCG is a gating mechanism that enables differentiable and sparse output selection [Louizos et al., 2018]. It extends the Binary Concrete distribution by applying a stretch-and-clamp operation, allowing the gate values to approximate binary behavior while remaining within the interval $[0, 1]$. Each gate is controlled by a learnable parameter and is sampled through a reparameterization trick involving a temperature parameter and a uniformly sampled random variable.

In the original work, HCG was proposed for input feature selection, where sparsity was encouraged via an L_0-style regularization term. The expected number of active gates is analytically computed as:

$$\mathcal{R} = \sum_{d=1}^{D} Sigmoid\left(\log \alpha_d - \beta \log\left(\frac{-\gamma}{\zeta}\right)\right), \tag{6}$$

which is then added to the loss function to penalize unnecessary features. Here, $\log \alpha_d$ denotes the learnable location parameter of the d-th hard concrete gate, β is the temperature parameter

controlling the sharpness of the continuous relaxation, and γ and ζ define the lower and upper bounds of the stretch interval before the clamp operation, respectively. Following the original hard concrete formulation, we set $\gamma = -0.1$ and $\zeta = 1.1$. The temperature parameter β is treated as a dataset-dependent hyperparameter and specified separately for each experimental setting.

While aggressive gating by HCG is undesirable for input selection due to the risk of prematurely excluding relevant features, this same property is advantageous for output selection in our framework. By introducing HCG gating only after input selection has converged, we ensure that output targets are either decisively selected or excluded, leading to a stable and interpretable ternary partition.

Although the mathematical form of this term is analogous to regularization terms used for input selection, we employ it here as a coverage incentive rather than a penalty. Hence, in our framework, $G_{\text{incentive}}$ is equivalent to $\mathcal{R}$. Since we use it as an incentive rather than a penalty, it is incorporated into the loss function with a negative coefficient. This encourages the selection of as many reliably predictable output targets as possible. For clarity, we refer to $G_{incentive}$ as the output gate activation incentive, and the corresponding loss component as the incentive term.

B.3.3 Training Schedule for DGNN

The three-stage training schedule is designed to alleviate the conflict between input and output selection and to prevent the model from prematurely discarding targets that are difficult to predict before the input–output relationships have been sufficiently explored. Each stage focuses on a distinct optimization objective. In the support discovery stage, the model learns input supports while predicting all targets, thereby exploring associations between input features and the full output space. In the predictable-target discovery stage, the learned input gates are frozen so that the model can identify reliably predictable output targets under a stable input representation. In the joint refinement stage, the selected output targets are fixed, and the model further fine-tunes the input supports and predictors specifically for the chosen targets. This staged design allows input and output selection to be optimized progressively rather than competitively.

Stage 1: Support Discovery The first stage corresponds to support discovery, where the model learns input supports while predicting all targets. During this stage, STG input gates are optimized together with the predictor, whereas output gating is not applied. This allows the model to explore associations between input features and the full output space before any targets are excluded. To encourage exploration and avoid premature hard selection, the input regularization coefficient λ_{in} is kept low, so that the STG gates remain relatively soft during early training.

The loss function for Stage 1 is defined as:

$$\mathcal{L}_{\text{Stage 1}} = \text{MSE} + \lambda_{\text{in}} \mathcal{R}_{\text{in}}. \tag{7}$$

Here, MSE denotes the mean squared error between the predictions and the ground truth over all targets, $\mathcal{R}_{\text{in}}$ is the regularization term associated with the STG input gates, and λ_{in} controls the strength of input gate regularization.

The transition to Stage 2 occurs when the validation performance, measured by the Concordance Correlation Coefficient (CCC), stabilizes. The best model parameters from Stage 1 are then restored, the learned input gates are frozen, and output gating is activated for predictable-target discovery.

Stage 2: Predictable-Target Discovery The second stage corresponds to predictable-target discovery, where the input gates learned in Stage 1 are frozen and HCG output gates are optimized. During this stage, the predictor uses the fixed input supports to assess which target dimensions are reliably predictable. Output gating is therefore introduced only after the input–output associations have been sufficiently explored, preventing the model from discarding difficult targets before the input representation has stabilized.

Let $\mathbf{g}_{\text{out}} \in (0,1)^D$ denote the output gate activations for the D target dimensions, and let MSE_d be the mean squared error for target dimension d. The gated prediction loss is computed as:

$$\text{MSE}_{\text{gated}} = \frac{1}{D} \sum_{d=1}^{D} g_{\text{out},d} \cdot \text{MSE}_d. \tag{8}$$

Target dimensions with larger gate activations contribute more to the prediction loss, whereas dimensions with small gate activations are down-weighted.

The full Stage 2 objective is defined as:

$$\mathcal{L}_{\text{Stage 2}} = \text{MSE}_{\text{gated}} - \lambda_{\text{out}} G_{\text{incentive}}. \tag{9}$$

Here, $G_{\text{incentive}}$ denotes the output gate activation incentive, which encourages the model to select more output targets, and λ_{out} controls the strength of this incentive. This objective balances prediction accuracy and output coverage: the negative incentive term encourages more targets to be selected, while the gated MSE discourages the selection of targets that cannot be predicted reliably under the fixed input supports.

The transition to Stage 3 occurs when the output gate activations stabilize, measured by low variance over recent epochs. Once the output gates satisfy this stability criterion, the selected outputs are fixed for joint refinement.

Although HCG encourages nearly binary gate activations, some gates may remain in intermediate states after optimization. Output dimensions with gate activations greater than 0.5 are retained as selected targets. This step ensures that each target dimension is definitively selected or discarded before Stage 3.

Stage 3: Joint Refinement The third stage corresponds to joint refinement, where the selected outputs from Stage 2 are fixed and the model fine-tunes the input supports and predictors. During this stage, only the target dimensions retained after predictable-target discovery are used for optimization. The input gates are unfrozen and updated together with the predictor, allowing the model to refine feature selection specifically for the selected output subset.

The loss function for Stage 3 is defined as:

$$\mathcal{L}_{\text{Stage 3}} = \text{MSE}_{\text{gated}} + \lambda_{\text{in}} \mathcal{R}_{\text{in}}. \tag{10}$$

Here, $\text{MSE}_{\text{gated}}$ denotes the prediction loss computed over the fixed selected outputs, $\mathcal{R}_{\text{in}}$ is the regularization term associated with the STG input gates, and λ_{in} controls the strength of input gate regularization.

B.3.4 Synthetic Experiments

Synthetic data setup. For all synthetic recovery and intervention experiments, we use the synthetic data configurations described in Appendix B.1. The data-generation seed is fixed for the main synthetic recovery and intervention experiments, while the experimental seed controls the train/validation/test split, model initialization, and random intervention sampling. For the small-scale relaxed-objective validation, the data seed is varied with the experimental seed.

The synthetic generator controls variable scale and observation noise through the prescribed SNR. In the reported synthetic experiments, we use the generated variables directly without additional feature standardization. The implementation supports standardization based on training-set statistics, but this option is disabled in the final synthetic runs to preserve the controlled scale of the synthetic construction.

DGNN configuration. DGNN is trained separately in the two directions $X \to Y$ and $Y \to X$, using the same architecture and hyperparameters unless otherwise stated. Each directional predictor uses an MLP with three hidden layers of width 32 and ReLU nonlinearities. After training, input and output gates are thresholded at 0.5 to obtain discrete supports and predicted MP variables.

For the main synthetic recovery and AO intervention experiments, the input-gate learning rate is initialized at 10^{-3} and the final input-gate learning rate is 10^{-2}. The initial input sparsity coefficient is 0.03, the Stage-3 input sparsity coefficient is initialized at 0.01, the STG noise scale is 0.5, the sparsity coefficient grows by a factor of 1.25, and its maximum value is 0.08. The MLP learning rate is 10^{-2} in the initial stages and 5×10^{-3} in Stage 3. The output-gate learning rate is 2×10^{-3}, the output-gate incentive coefficient ranges from 1.25 to 1.5, and the hard-concrete temperature is 0.25 to 0.4.

For the relaxed-objective validation experiment, we use the same DGNN architecture and training schedule, but use an output-gate incentive coefficient 1.5 and hard-concrete temperature 0.35, following the configuration used for the low-dimensional discrete-validation runs.

Baseline implementations. We compare against three baselines using the same train/validation/test splits as those used for DGNN.

SCCA. Sparse CCA is implemented as a greedy sparse CCA baseline. The sparsity budgets for the two views are selected on the validation split by grid search. Final cross-view prediction is evaluated using the same family of MLP prediction heads as the other baselines.

DSCCA. The nonlinear sparse CCA baseline is implemented using an L_0-gated deep sparse CCA model. The latent dimension is 12, the hidden layers are $(32, 16)$, and the learning rate is 10^{-2}. The hard selection threshold is 0.01. The sparse-gating parameters are $a = 1.4$, $\sigma = 0.2$, and $\lambda_1 = \lambda_2 = 1.0$.

Multi-task Lasso. The Lasso baseline is implemented using bidirectional multi-task sparse regression. The reported runs use MultiTaskLasso with regularization strength $\alpha = 0.1$, intercept fitting enabled, a maximum of 5000 solver iterations, solver tolerance 10^{-4}, and coefficient row-norm threshold 10^{-5} for hard support extraction.

Exact-search predictor. For the exact-search baseline used in the small-scale validation of the relaxation, we replace the MLP prediction head with a random Fourier feature ridge regressor. This is done only to reduce the computational burden of repeatedly evaluating many candidate subsets. We use 256 random Fourier features followed by ridge regression with regularization 10^{-3}. This predictor is used solely inside the exact-search procedure and is not part of DGNN.

Prediction heads for baselines and interventions. For baseline cross-view prediction and AO intervention experiments, we train MLP regression heads with hidden dimensions $(32, 32, 32)$, ReLU activations, and a learning rate of 10^{-3}.

B.3.5 Real-world Dataset Experiments

DGNN configuration. DGNN is trained separately for the two prediction directions, $X \to Y$ and $Y \to X$. In the real-world datasets, these correspond to $ADAS \to CDR$ and $CDR \to ADAS$ for the ADAS–CDR dataset, and $qCPs \to NB$ and $NB \to qCPs$ for the qCPs–NB dataset. The architecture and hyperparameter settings used in these experiments are summarized in Table 13.

Table 13: Hyperparameter settings for Realworld dataset Experiments

Module	Hyperparameters	ADAS → CDR	CDR → ADAS	qCPs → NB	NB → qCPs
MLP	hidden layer	$[32, 32, 32]$	$[64, 64]$	$[64, 64, 64]$	$[64, 64, 64]$
	lr in stage 1	0.005	0.005	0.005	0.005
	lr in stage 3	0.01	0.01	0.01	0.01
Input gate	lr in stage 1	0.08	0.08	0.05	0.05
	lr in stage 3	0.1	0.1	0.1	0.1
	Stage-1 sparsity coefficient	0.01	0.01	0.0005	0.05
	Stage-3 Initial sparsity coefficient	0.08	0.03	0.001	0.03
	Stage-3 sparsity growing factor	1.28	1.28	1.2	1.25
	Stage-3 sparsity max value	0.5	0.5	0.002	0.5
	Input STG noise scale	0.5	0.5	0.5	0.5
Output gate	lr	0.02	0.02	0.02	0.02
	Incentive coefficient	1.5	1.3	1.1	1.5
	Temperature	0.25	0.25	0.25	0.25

Baseline implementation We compare against three baselines using the same train/validation/test splits as DGNN. Because the baseline methods only support a binary partition of features, i.e., selected and unselected, we control their sparsity to enable fair comparisons in both cross-view prediction and downstream classification tasks. Specifically, for each view, the hyperparameters of each baseline are chosen such that the number of selected features approximately matches the median number of MP features selected by DGNN. The hyperparameter settings for all baseline methods are reported in Table 14.

Table 14: Hyperparameter settings for SCCA, DSCCA, and Lasso on the ADAS–CDR and qCPs–NB datasets. For ADAS–CDR, ADAS is treated as X and CDR as Y; for qCPs–NB, qCPs is treated as X and NB as Y.

Method	Parameter	ADAS–CDR	qCPs–NB
SCCA	# X variables	4	15
	# Y variables	7	6
DSCCA	Hidden layers	[64, 64, 64]	[64, 64, 64]
	lr	0.01	0.01
	alpha	1.4	1.4
	sigma	0.25	0.25
	λ_x	1.1	0.3
	λ_y	0.15	1.6
Lasso	α_{XtoY}	0.7	0.08
	α_{YtoX}	0.2	1.05
	eps	10^{-6}	10^{-6}

Cross-view prediction implementation For baseline cross-view prediction and post-hoc intervention experiments, we train MLP regression heads with hidden dimensions $(32, 32, 32)$, ReLU activation, learning rate 10^{-3}.

Downstream classification task implementation We implemented the downstream classification task using support vector classification (SVC) with feature standardization before model fitting. For each random seed, the data were split into training, validation, and held-out test sets. Hyperparameters were selected by grid search over three SVC parameters: the regularization parameter $C \in \{0.01, 0.1, 1, 10\}$, the kernel type $\in \{\texttt{linear}, \texttt{rbf}\}$, and the RBF kernel coefficient $\gamma \in \{\texttt{scale}, 0.01, 0.1\}$. The best configuration was selected according to validation macro-F1, after which the model was retrained on the combined training and validation sets and evaluated on the held-out test set.

C Extended Experimental Results

C.1 Extended Results for Synthetic Experiments

C.1.1 Extended Results for Synthetic Recovery

Tables 15, 16, 17 and 18 provide additional synthetic results at lower and higher SNR levels. These results complement the synthetic experiments in the main text by further testing the three claims studied in Sec. 4.1: recoverability of the MP/AO/UP structure, utility for cross-view prediction, and the distinct role of AO variables. Since the baselines naturally induce only binary partitions, we report the metrics that are well-defined for each method: SCCA and DSCCA are evaluated on the MP versus AO∪UP binary projection, while Lasso is evaluated on the MP∪AO versus UP binary projection. DGNN, in contrast, returns the full ternary partition and is therefore evaluated with respect to all three roles simultaneously.

As shown in Tables 15 and 16, the recovery results show that the conclusions in the main text are stable across noise levels. At $SNR = 7$dB, DGNN still recovers the full ternary structure with high accuracy in all three regimes, despite the substantially harder prediction setting. In R1 and R3, it achieves the strongest overall recovery and prediction performance. In R2, DSCCA obtains a slightly higher MP-F1, but it only identifies a binary cross-view association structure and does not separate AO variables from UP variables. This difference is central to the proposed problem formulation: the goal of MPD is not only to find variables involved in cross-view dependence, but also to distinguish predictable variables from variables that are useful only as auxiliary support.

As shown in Table 17 and 18, at $SNR = 20$dB, the same pattern becomes more pronounced. DGNN achieves the highest average R^2 and CCC across all three regimes, indicating that the learned MP/AO decomposition improves the actual cross-view prediction task used to define mutual predictability. This supports the utility claim in the main text: the recovered auxiliary supports are not only structurally interpretable but also improve the reconstruction of the opposite-view MP variables.

Table 15: Recovery of individual variable partitions across three regimes at $SNR = 7$ dB and $\theta = 0.5$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	MP-F1	AO-F1	UP-F1
R1.SCCA	$0.546_{\pm 0.056}$	–	–
R1.DSCCA	$0.933_{\pm 0.037}$	–	–
R1.Lasso	–	–	$0.940_{\pm 0.053}$
R1.Ours	$\mathbf{0.954_{\pm 0.032}}$	$\mathbf{0.876_{\pm 0.057}}$	$\mathbf{0.996_{\pm 0.012}}$
R2.SCCA	$0.584_{\pm 0.098}$	–	–
R2.DSCCA	$\mathbf{0.924_{\pm 0.047}}$	–	–
R2.Lasso	–	–	$0.943_{\pm 0.054}$
R2.Ours	$0.837_{\pm 0.040}$	$\mathbf{0.789_{\pm 0.031}}$	$\mathbf{1.000_{\pm 0.000}}$
R3.SCCA	$0.530_{\pm 0.063}$	–	–
R3.DSCCA	$0.662_{\pm 0.030}$	–	–
R3.Lasso	–	–	$0.364_{\pm 0.163}$
R3.Ours	$\mathbf{0.822_{\pm 0.052}}$	$\mathbf{0.780_{\pm 0.044}}$	$\mathbf{0.987_{\pm 0.027}}$

Table 16: Recovery of composite variable partitions and predictive performance across three regimes at $SNR = 7$ dB and $\theta = 0.5$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	AO+MP-F1	AO+UP-F1	Avg. R^2	Avg. CCC
R1.SCCA	–	$0.559_{\pm 0.065}$	$0.253_{\pm 0.050}$	$0.329_{\pm 0.068}$
R1.DSCCA	–	$0.933_{\pm 0.049}$	$0.458_{\pm 0.015}$	$0.619_{\pm 0.022}$
R1.Lasso	$0.980_{\pm 0.017}$	–	$0.351_{\pm 0.017}$	$0.466_{\pm 0.020}$
R1.Ours	$\mathbf{0.998_{\pm 0.005}}$	$\mathbf{0.944_{\pm 0.022}}$	$\mathbf{0.508_{\pm 0.011}}$	$\mathbf{0.679_{\pm 0.008}}$
R2.SCCA	–	$0.576_{\pm 0.109}$	$0.175_{\pm 0.030}$	$0.268_{\pm 0.044}$
R2.DSCCA	–	$\mathbf{0.909_{\pm 0.056}}$	$0.276_{\pm 0.023}$	$0.428_{\pm 0.032}$
R2.Lasso	$0.984_{\pm 0.015}$	–	$0.230_{\pm 0.015}$	$0.346_{\pm 0.018}$
R2.Ours	$\mathbf{1.000_{\pm 0.000}}$	$0.878_{\pm 0.021}$	$\mathbf{0.400_{\pm 0.009}}$	$\mathbf{0.591_{\pm 0.008}}$
R3.SCCA	–	$0.490_{\pm 0.072}$	$0.254_{\pm 0.038}$	$0.385_{\pm 0.055}$
R3.DSCCA	–	$0.573_{\pm 0.041}$	$0.322_{\pm 0.010}$	$0.487_{\pm 0.013}$
R3.Lasso	$0.903_{\pm 0.014}$	–	$0.275_{\pm 0.011}$	$0.424_{\pm 0.015}$
R3.Ours	$\mathbf{0.997_{\pm 0.007}}$	$\mathbf{0.849_{\pm 0.034}}$	$\mathbf{0.439_{\pm 0.021}}$	$\mathbf{0.625_{\pm 0.017}}$

The high AO-F1 and UP-F1 further show that DGNN does not obtain this predictive gain by simply selecting more variables; instead, it separates useful auxiliary variables from nuisance variables.

These results also clarify the relationship between MPD and the binary baselines discussed in Sec. 4. Sparse CCA and DSCCA are well-suited for identifying variables involved in cross-view association, but their output corresponds to a coarse MP versus AO∪UP projection. Lasso recovers variables useful for prediction and is therefore competitive on the MP∪AO projection, especially in the persistent-support regimes R1 and R2. However, Lasso cannot determine which useful variables are themselves mutually predictable. DGNN is the only method that directly matches the ternary decomposition problem posed in Eq. (1), which explains why its advantage is most visible on AO-F1, UP-F1, and the joint recovery of MP/AO/UP.

C.1.2 AO intervention

Tables 19 and 20 further test the third main claim from Sec. 4.1: AO variables are functionally meaningful support variables rather than unpredictable noise. After learning the decomposition, we remove one recovered role from the input side and evaluate how much cross-view prediction degrades. The "All" row uses the complete recovered support, while $-$MP, $-$AO, and $-$UP remove the corresponding recovered roles. The random-removal baseline removes a mixed-role subset of the same size and serves as a control for the effect of removing variables independent of their learned role.

The intervention results are consistent with the semantics of the proposed decomposition. Removing MP variables produces the largest degradation in both CCC and R^2, confirming that MP variables form the core cross-view predictable structure. Removing AO variables also causes a clear performance drop at both SNR levels, even though AO variables are not themselves classified as mutually

Table 17: Recovery of individual variable partitions across three regimes at $SNR = 20$ dB and $\theta = 0.65$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	MP-F1	AO-F1	UP-F1
R1.SCCA	$0.617_{\pm 0.065}$	–	–
R1.DSCCA	$0.935_{\pm 0.022}$	–	–
R1.Lasso	–	–	$\mathbf{1.000_{\pm 0.000}}$
R1.Ours	$\mathbf{0.960_{\pm 0.014}}$	$\mathbf{0.918_{\pm 0.028}}$	$\mathbf{1.000_{\pm 0.000}}$
R2.SCCA	$0.513_{\pm 0.057}$	–	–
R2.DSCCA	$\mathbf{0.905_{\pm 0.031}}$	–	–
R2.Lasso	–	–	$0.979_{\pm 0.035}$
R2.Ours	$0.874_{\pm 0.027}$	$\mathbf{0.823_{\pm 0.028}}$	$\mathbf{1.000_{\pm 0.000}}$
R3.SCCA	$0.583_{\pm 0.081}$	–	–
R3.DSCCA	$0.652_{\pm 0.031}$	–	–
R3.Lasso	–	–	$0.711_{\pm 0.078}$
R3.Ours	$\mathbf{0.901_{\pm 0.049}}$	$\mathbf{0.773_{\pm 0.113}}$	$\mathbf{0.902_{\pm 0.109}}$

Table 18: Recovery of composite variable partitions and predictive performance across three regimes at $SNR = 20$ dB and $\theta = 0.65$, with results aggregated over bidirectional predictions. "–" indicates that the corresponding metric is not applicable to the method.

Method	AO+MP-F1	AO+UP-F1	Avg. R^2	Avg. CCC
R1.SCCA	–	$0.605_{\pm 0.071}$	$0.342_{\pm 0.052}$	$0.370_{\pm 0.057}$
R1.DSCCA	–	$0.930_{\pm 0.025}$	$0.610_{\pm 0.018}$	$0.719_{\pm 0.025}$
R1.Lasso	$\mathbf{1.000_{\pm 0.000}}$	–	$0.560_{\pm 0.004}$	$0.619_{\pm 0.003}$
R1.Ours	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{0.963_{\pm 0.012}}$	$\mathbf{0.808_{\pm 0.009}}$	$\mathbf{0.882_{\pm 0.006}}$
R2.SCCA	–	$0.492_{\pm 0.051}$	$0.278_{\pm 0.036}$	$0.339_{\pm 0.042}$
R2.DSCCA	–	$0.888_{\pm 0.040}$	$0.444_{\pm 0.015}$	$0.583_{\pm 0.022}$
R2.Lasso	$0.995_{\pm 0.008}$	–	$0.400_{\pm 0.007}$	$0.495_{\pm 0.008}$
R2.Ours	$\mathbf{1.000_{\pm 0.000}}$	$\mathbf{0.899_{\pm 0.017}}$	$\mathbf{0.672_{\pm 0.012}}$	$\mathbf{0.804_{\pm 0.008}}$
R3.SCCA	–	$0.554_{\pm 0.084}$	$0.540_{\pm 0.053}$	$0.632_{\pm 0.067}$
R3.DSCCA	–	$0.576_{\pm 0.044}$	$0.657_{\pm 0.041}$	$0.779_{\pm 0.031}$
R3.Lasso	$0.938_{\pm 0.016}$	–	$0.611_{\pm 0.023}$	$0.732_{\pm 0.023}$
R3.Ours	$\mathbf{0.984_{\pm 0.016}}$	$\mathbf{0.889_{\pm 0.055}}$	$\mathbf{0.821_{\pm 0.020}}$	$\mathbf{0.899_{\pm 0.013}}$

predictable targets. This directly supports the motivation of MPD: some variables should be distinguished from ordinary private or unpredictable variables because they are necessary for predicting the opposite-view MP set.

The effect of AO removal is especially clear in R3, where AO variables are designed to resolve branch, sign, phase, or many-to-one ambiguities. In this regime, removing AO variables substantially reduces prediction quality, showing that the model relies on AO variables to disambiguate otherwise predictable cross-view structure. In contrast, removing UP variables has little negative effect and can even slightly improve performance, which is consistent with UP variables behaving as nuisance variables for cross-view prediction.

Finally, the random-removal control typically leads to an intermediate degradation between role-specific removals. This indicates that the observed performance changes are not explained only by the number of removed variables. Rather, they reflect the functional roles recovered by DGNN: MP variables define what can be predicted across views, AO variables provide information needed to make those predictions, and UP variables do not contribute to the cross-view prediction objective.

C.1.3 Stability under Correlated or Redundant Variables

The decomposition remains stable even when variables are strongly correlated. A variable that is strongly correlated with an MP variable will be identified as part of the MP, whereas a variable that is strongly correlated with a UP variable will be classified as UP. For variables strongly correlated with an AO variable, the assignment depends on the trade-off between predictive contribution and support cost: under strong but imperfect correlation, one or both variables may be selected as AO. If two variables are exact duplicates, either may be selected as the AO variable, while the other is assigned

Table 19: AO intervention across three regimes at $\mathrm{SNR} = 7.0$ dB. Entries in the first column denote the removed input subset. −Rand. removes a random mixed-role subset matched in size to the corresponding intervention by sampling variables from the recovered partition without enforcing a single role. All metrics are aggregated over both prediction directions and reported as mean$_{\pm\mathrm{std}}$ over 10 random seeds.

Intervention	**Avg. CCC**			**Avg.** R^2		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**
All	$0.675_{\pm 0.005}$	$0.585_{\pm 0.008}$	$0.608_{\pm 0.048}$	$0.510_{\pm 0.007}$	$0.403_{\pm 0.007}$	$0.421_{\pm 0.039}$
−MP	$0.064_{\pm 0.028}$	$0.146_{\pm 0.038}$	$0.365_{\pm 0.065}$	$0.046_{\pm 0.022}$	$0.097_{\pm 0.028}$	$0.245_{\pm 0.051}$
−AO	$0.561_{\pm 0.018}$	$0.398_{\pm 0.030}$	$0.280_{\pm 0.052}$	$0.407_{\pm 0.013}$	$0.262_{\pm 0.021}$	$0.194_{\pm 0.039}$
−UP	$0.678_{\pm 0.005}$	$0.588_{\pm 0.007}$	$0.617_{\pm 0.045}$	$0.515_{\pm 0.008}$	$0.408_{\pm 0.006}$	$0.433_{\pm 0.039}$
−Rand.	$0.579_{\pm 0.030}$	$0.502_{\pm 0.027}$	$0.531_{\pm 0.061}$	$0.434_{\pm 0.023}$	$0.345_{\pm 0.017}$	$0.378_{\pm 0.049}$

Table 20: AO intervention across three regimes at $\mathrm{SNR} = 20.0$ dB. Entries in the first column denote the removed input subset. −Rand. removes a random mixed-role subset matched in size to the corresponding intervention by sampling variables from the recovered partition without enforcing a single role. All metrics are aggregated over both prediction directions and reported as mean$_{\pm\mathrm{std}}$ over 10 random seeds.

Intervention	**Avg. CCC**			**Avg.** R^2		
	R1	**R2**	**R3**	**R1**	**R2**	**R3**
All	$0.885_{\pm 0.010}$	$0.794_{\pm 0.021}$	$0.861_{\pm 0.019}$	$0.814_{\pm 0.015}$	$0.672_{\pm 0.030}$	$0.782_{\pm 0.020}$
−MP	$0.081_{\pm 0.051}$	$0.230_{\pm 0.069}$	$0.325_{\pm 0.065}$	$0.076_{\pm 0.049}$	$0.201_{\pm 0.063}$	$0.244_{\pm 0.047}$
−AO	$0.708_{\pm 0.031}$	$0.495_{\pm 0.043}$	$0.516_{\pm 0.088}$	$0.590_{\pm 0.023}$	$0.367_{\pm 0.030}$	$0.451_{\pm 0.073}$
−UP	$0.886_{\pm 0.011}$	$0.806_{\pm 0.016}$	$0.873_{\pm 0.023}$	$0.817_{\pm 0.016}$	$0.691_{\pm 0.024}$	$0.793_{\pm 0.030}$
−Rand.	$0.758_{\pm 0.019}$	$0.666_{\pm 0.029}$	$0.750_{\pm 0.041}$	$0.434_{\pm 0.023}$	$0.550_{\pm 0.022}$	$0.652_{\pm 0.041}$

to UP. Thus, correlation does not destabilize the recovered predictive support, although the identity of the selected AO representative may vary when the variables contribute equally to prediction.

We evaluate this behavior using a controlled experiment based on R1. Specifically, we replace one UP variable with a variance-matched proxy for a ground-truth AO variable, a:

$$a_\rho = \rho a + \sqrt{1-\rho^2}\,\xi, \qquad \rho \in \{0, 0.5, 0.9, 0.99, 1.0\},$$

where ξ is independent of a.

As shown in Table 21, pair recovery remains 1.0, and Avg. CCC remains between 0.823 and 0.831 across all correlation levels. Under strong but imperfect correlation, occasional co-selection reflects the trade-off between predictive contribution and support cost. At $\rho = 1$, either variable is selected in $5/10$ runs, reflecting the non-uniqueness of AO selection when the variables are exact duplicates.

C.1.4 Computational Complexity and Runtime

Computational complexity. Let $d = p + q$ denote the total dimensionality of the two views. With L hidden layers of width h and n samples, one training epoch of the bidirectional DGNN has time complexity

$$O\big(n\left[hd + Lh^2\right]\big)\,.$$

Its parameter memory is $O(hd + Lh^2)$. Including the in-memory dataset and mini-batch activations, the total training memory is

$$O\big(nd + hd + Lh^2 + bLh\big)\,,$$

where b is the mini-batch size. For fixed n, h, L, and b, both per-epoch training time and total training memory scale linearly with the total dimensionality d.

Runtime comparison. We benchmarked the end-to-end runtime required to obtain partitions for both views at $d \in \{40, 120, 400, 1000\}$. All methods were run CPU-only with fixed hyperparameters, and the results were pooled across 10 seeds. Table 22 reports the average runtime for each method.

Table 21: Controlled redundancy experiment in R1 over 10 seeds. Pair recovery counts the pair as recovered if at least one of $\{a, a_\rho\}$ is selected as AO.

ρ	Pair recovery	# AO selected/pair	$P(a \in \widehat{X}_{\mathrm{AO}})$	$P(a_\rho \in \widehat{X}_{\mathrm{AO}})$	Avg. CCC
0.00	1.000	1.000	1.000	0.000	0.831 ± 0.014
0.50	1.000	1.000	1.000	0.000	0.831 ± 0.014
0.90	1.000	1.200	1.000	0.200	0.823 ± 0.021
0.99	1.000	1.100	0.800	0.300	0.829 ± 0.016
1.00	1.000	1.000	0.500	0.500	0.831 ± 0.014

Table 22: Average end-to-end runtime for obtaining partitions for both views (seconds).

Method	$d = 40$	$d = 120$	$d = 400$	$d = 1000$
Lasso	15.7	32.6	38.6	44.6
SCCA	83.9	149.7	577.2	2780.0
DSCCA	56.0	66.7	79.1	104.3
DGNN	62.5	193.8	528.4	1489.0

DGNN completed both-view partitioning with an average runtime increasing from 62.5 seconds at $d = 40$ to 1489.0 seconds at $d = 1000$. It was slower than Lasso and DSCCA across the tested dimensionalities, but faster than SCCA at $d = 40$, 400, and 1000.

C.2 Extended Results for Real World Datasets

C.2.1 Additional Dataset and Experimental Results.

We have extended our evaluation to an additional real-world dataset from the Alzheimer's Disease Neuroimaging Initiative (ADNI). Specifically, the dataset includes two clinical assessments: the Mini-Mental State Examination (MMSE), comprising 6 features, and the Montreal Cognitive Assessment (MoCA), comprising 8 features. The dataset contains 665 records in total.

We conducted experiments on this dataset for both *cross-view prediction* and *classification with feature partitions*. All experimental protocols were aligned with those described in the main text. The corresponding results are reported below.

The results reported in Tables 23 and 24 are consistent with our main findings, providing further evidence of the effectiveness and robustness of our method in real-world settings.

C.2.2 Cross-view prediction.

Additional cross-view prediction results on ADAS–CDR. Table 25 further evaluates cross-view predictability using an intermediate CCC threshold of 0.6, together with the number of output features whose CCC exceeds 0.5, 0.6, and 0.7. The proposed method consistently identifies a larger set of reliably predictable variables than all baselines. In the ADAS $\rightarrow$ CDR direction, our method selects a median of 7 features with CCC > 0.5 and CCC > 0.6, whereas the baseline methods identify only around 4–5 such features. The advantage remains at the stricter CCC > 0.7 threshold, where our method identifies a median of 5 features compared with 3 for the baselines. In the reverse CDR $\rightarrow$ ADAS direction, the same pattern holds: our method identifies medians of 4 and 3 features above CCC thresholds of 0.5 and 0.6, respectively, while the baselines identify only around 2 and 1 features. These results show that DGNN does not merely improve average prediction quality, but also being able to identify more variables that can be reliably predicted across views.

The qCPs-NB results show a similar trend. As shown in Table 26, the proposed method again identifies substantially more predictable variables across all CCC thresholds in both prediction directions. This confirms that the improved cross-view predictability of DGNN is consistent across both real-world datasets.

Table 23: Cross-system feature prediction on the MMSE–MoCA dataset.

Direction	Methods	Avg. R^2	$\overline{\text{CCC}}_{>.5}$	$\overline{\text{CCC}}_{>.7}$	$\#\text{CCC}_{>.5}$	$\#\text{CCC}_{>.7}$
MMSE → MoCA	Ours	$\mathbf{0.525_{\pm 0.098}}$	$\mathbf{0.720_{\pm 0.062}}$	$\mathbf{0.743_{\pm 0.036}}$	**2**	**2**
	LASSO	$0.249_{\pm 0.084}$	$0.557_{\pm 0.040}$	–	1	0
	DSCCA	$0.239_{\pm 0.080}$	$0.543_{\pm 0.022}$	–	1	0
	SCCA	$0.231_{\pm 0.083}$	$0.557_{\pm 0.040}$	–	1	0
MoCA → MMSE	Ours	$\mathbf{0.529_{\pm 0.091}}$	$0.725_{\pm 0.051}$	$0.762_{\pm 0.040}$	**2.5**	**2**
	LASSO	$0.264_{\pm 0.120}$	$\mathbf{0.744_{\pm 0.044}}$	$0.762_{\pm 0.016}$	1	1
	DSCCA	$0.281_{\pm 0.114}$	$0.734_{\pm 0.049}$	$\mathbf{0.763_{\pm 0.016}}$	1	1
	SCCA	$0.237_{\pm 0.115}$	$0.734_{\pm 0.044}$	$0.762_{\pm 0.040}$	1	1

Table 24: Comparison of feature partitions across methods for downstream classification on the MMSE–MoCA dataset.

Method	Setting	MMSE	MoCA
All	All	$0.630_{\pm 0.035}$	$0.573_{\pm 0.033}$
Lasso	Unsel	$0.504_{\pm 0.063}$	$0.573_{\pm 0.033}$
	Sel	$0.636_{\pm 0.034}$	$0.548_{\pm 0.044}$
DSCCA	Unsel	$0.541_{\pm 0.100}$	$0.550_{\pm 0.041}$
	Sel	$0.628_{\pm 0.031}$	$0.592_{\pm 0.042}$
SCCA	Unsel	$0.484_{\pm 0.075}$	$0.546_{\pm 0.044}$
	Sel	$0.632_{\pm 0.039}$	$\mathbf{0.596_{\pm 0.047}}$
Ours	AO	$0.519_{\pm 0.083}$	$0.519_{\pm 0.052}$
	MP	$0.633_{\pm 0.037}$	$\mathbf{0.596_{\pm 0.047}}$
	AO+MP	$\mathbf{0.639_{\pm 0.030}}$	$0.576_{\pm 0.034}$
	UP	$0.427_{\pm 0.039}$	$0.508_{\pm 0.052}$
	AO+UP	$0.504_{\pm 0.085}$	$0.546_{\pm 0.044}$

C.2.3 Classification with feature partitions.

We evaluate the utility of the learned variable partitions through downstream classification tasks. While the main text reports accuracy as the primary evaluation metric, Tables 27–30 present a more comprehensive set of results, including precision, recall, and F1-score, along with the corresponding standard deviations. These results are reported separately for each evaluation setting, method, and feature partition, providing additional evidence on the downstream discriminative utility of the identified partitions.

C.2.4 Ablation on Model Components.

ADAS-CDR Table 31 provides additional evaluation metrics that are not reported in the main text. The table shows that the full model maintains strong prediction quality while identifying more high-concordance targets under stricter CCC thresholds. In the ADAS → CDR direction, the standard model identifies a median of 7 targets with CCC > 0.5 and CCC > 0.6, and 5 targets with CCC > 0.7. In contrast, the 'no Gate' and 'no OutputGate' variants identify substantially fewer targets above the stricter thresholds, especially for CCC > 0.7. In the reverse CDR → ADAS direction, the

Table 25: Performance comparison.

Direction	Methods	$\overline{\text{CCC}}_{>.6}$	$\#\text{CCC}_{>.5}$	$\#\text{CCC}_{>.6}$	$\#\text{CCC}_{>.7}$
ADAS	SCCA	$0.711_{\pm 0.031}$	5.00[5.00,5.00]	5.00[5.00,5.00]	3.00[2.00,4.00]
→	DSCCA	$0.712_{\pm 0.032}$	5.00[5.00,5.00]	5.00[1.25,5.00]	3.00[2.00,4.00]
CDR	Lasso	$0.716_{\pm 0.032}$	4.50[4.00,5.00]	4.00[4.00,5.00]	3.00[2.00,5.00]
	Ours	$\mathbf{0.739_{\pm 0.034}}$	**7.00[7.00,7.00]**	**7.00[7.00,7.00]**	**5.00[5.00,7.00]**
CDR	SCCA	$0.647_{\pm 0.028}$	2.00[2.00,2.00]	1.00[1.00,2.00]	0.00[0.00,0.00]
→	DSCCA	$0.642_{\pm 0.029}$	2.00[1.25,2.00]	1.00[1.00,1.00]	0.00[0.00,0.00]
ADAS	Lasso	$0.649_{\pm 0.030}$	2.00[2.00,2.00]	1.00[0.25,2.00]	0.00[0.00,0.00]
	Ours	$\mathbf{0.676_{\pm 0.020}}$	**4.00[4.00,4.00]**	**3.00[2.00,3.00]**	**1.00[1.00,1.00]**

Table 26: Performance comparison.

Direction	Methods	$\overline{\text{CCC}}_{>.6}$	$\#\text{CCC}_{>.5}$	$\#\text{CCC}_{>.6}$	$\#\text{CCC}_{>.7}$
qCPs	SCCA	$0.650_{\pm 0.028}$	2.00[2.00,2.00]	2.00[1.00,2.00]	0.00[0.00,0.00]
$\rightarrow$	DSCCA	$0.658_{\pm 0.020}$	1.50[1.00,2.00]	1.00[0.25,2.00]	0.00[0.00,0.00]
NB	Lasso	$0.648_{\pm 0.034}$	2.00[2.00,2.00]	1.50[1.00,2.00]	0.00[0.00,0.00]
	Ours	$\mathbf{0.685_{\pm 0.028}}$	**6.00[6.00,6.00]**	**6.00[5.25,6.00]**	**2.50[1.25,3]**
NB	SCCA	$0.665_{\pm 0.171}$	8.00[7.00,8.00]	6.00[5.00,6.00]	1.00[1.00,1.00]
$\rightarrow$	DSCCA	$0.665_{\pm 0.021}$	8.00[6.25,8.00]	5.00[4.25,6.00]	1.00[1.00,1.00]
qCPs	Lasso	$0.667_{\pm 0.018}$	7.00[7.00,8.00]	5.00[5.00,6.00]	1.00[1.00,1.00]
	Ours	$\mathbf{0.671_{\pm 0.018}}$	**15.0[15.0,15.0]**	**12.0[9.25,13.75]**	**3.50[1.50,4.00]**

Table 27: Comparison of feature partitions across methods for downstream classification on the ADAS side of the ADAS–CDR dataset.

Methods	Setting	Accuracy	Precision	Recall	F1
	ADAS All	$0.627_{\pm 0.021}$	$0.639_{\pm 0.020}$	$0.621_{\pm 0.020}$	$0.626_{\pm 0.020}$
SCCA	Unsel	$0.606_{\pm 0.029}$	$0.613_{\pm 0.025}$	$0.593_{\pm 0.028}$	$0.598_{\pm 0.026}$
	Sel	$0.623_{\pm 0.018}$	$0.645_{\pm 0.017}$	$0.603_{\pm 0.016}$	$0.613_{\pm 0.015}$
DSCCA	Unsel	$0.583_{\pm 0.041}$	$0.599_{\pm 0.026}$	$0.565_{\pm 0.051}$	$0.565_{\pm 0.059}$
	Sel	$0.621_{\pm 0.028}$	$0.621_{\pm 0.068}$	$0.606_{\pm 0.040}$	$0.605_{\pm 0.059}$
Lasso	Unsel	$0.536_{\pm 0.040}$	$0.466_{\pm 0.105}$	$0.498_{\pm 0.065}$	$0.463_{\pm 0.095}$
	Sel	$0.610_{\pm 0.023}$	$0.627_{\pm 0.022}$	$0.598_{\pm 0.027}$	$0.606_{\pm 0.024}$
Our	AO	$0.513_{\pm 0.024}$	$0.422_{\pm 0.073}$	$0.458_{\pm 0.030}$	$0.406_{\pm 0.055}$
	MP	$0.633_{\pm 0.016}$	$0.646_{\pm 0.018}$	$0.620_{\pm 0.012}$	$0.628_{\pm 0.012}$
	AO+MP	$\mathbf{0.639_{\pm 0.020}}$	$\mathbf{0.655_{\pm 0.020}}$	$\mathbf{0.629_{\pm 0.020}}$	$\mathbf{0.636_{\pm 0.017}}$
	UP	$0.494_{\pm 0.027}$	$0.375_{\pm 0.029}$	$0.433_{\pm 0.027}$	$0.365_{\pm 0.032}$
	AO+UP	$0.518_{\pm 0.030}$	$0.461_{\pm 0.074}$	$0.477_{\pm 0.036}$	$0.444_{\pm 0.063}$

Table 28: Downstream classification performance of different feature partitions on the CDR side of the ADAS–CDR dataset.

Methods	Setting	Acc.	Precision	Recall	F1
	CDR All	$0.764_{\pm 0.030}$	$0.775_{\pm 0.027}$	$0.787_{\pm 0.028}$	$0.768_{\pm 0.029}$
SCCA	Unsel	$0.694_{\pm 0.068}$	$0.705_{\pm 0.088}$	$0.713_{\pm 0.094}$	$0.682_{\pm 0.106}$
	Sel	$0.765_{\pm 0.034}$	$\mathbf{0.781_{\pm 0.033}}$	$0.787_{\pm 0.030}$	$0.769_{\pm 0.033}$
DSCCA	Unsel	$0.580_{\pm 0.121}$	$0.461_{\pm 0.230}$	$0.543_{\pm 0.152}$	$0.478_{\pm 0.194}$
	Sel	$0.764_{\pm 0.031}$	$\mathbf{0.781_{\pm 0.027}}$	$0.786_{\pm 0.029}$	$0.767_{\pm 0.031}$
Lasso	Unsel	$0.611_{\pm 0.112}$	$0.517_{\pm 0.200}$	$0.588_{\pm 0.151}$	$0.535_{\pm 0.182}$
	Sel	$0.757_{\pm 0.028}$	$0.775_{\pm 0.029}$	$0.781_{\pm 0.026}$	$0.761_{\pm 0.028}$
Our	AO	$0.453_{\pm 0.025}$	$0.409_{\pm 0.109}$	$0.388_{\pm 0.025}$	$0.293_{\pm 0.044}$
	MP	$\mathbf{0.768_{\pm 0.028}}$	$\mathbf{0.781_{\pm 0.027}}$	$\mathbf{0.790_{\pm 0.026}}$	$\mathbf{0.772_{\pm 0.028}}$
	AO+MP	$0.766_{\pm 0.025}$	$0.777_{\pm 0.024}$	$0.789_{\pm 0.022}$	$0.770_{\pm 0.024}$
	UP	$0.411_{\pm 0.032}$	$0.196_{\pm 0.128}$	$0.348_{\pm 0.031}$	$0.216_{\pm 0.055}$
	AO+UP	$0.456_{\pm 0.026}$	$0.417_{\pm 0.100}$	$0.392_{\pm 0.025}$	$0.299_{\pm 0.043}$

Table 29: Downstream classification performance of different feature partitions on the qCPs side of the qCPs–NB dataset.

Methods	Setting	Acc.	Precision	Recall	F1
	qCPs All	$\mathbf{0.590_{\pm 0.016}}$	$0.617_{\pm 0.020}$	$\mathbf{0.580_{\pm 0.014}}$	$\mathbf{0.590_{\pm 0.013}}$
SCCA	Unsel	$0.562_{\pm 0.025}$	$0.590_{\pm 0.027}$	$0.550_{\pm 0.023}$	$0.559_{\pm 0.023}$
	Sel	$0.582_{\pm 0.024}$	$0.612_{\pm 0.022}$	$0.568_{\pm 0.026}$	$0.579_{\pm 0.025}$
DSCCA	Unsel	$0.563_{\pm 0.038}$	$0.590_{\pm 0.041}$	$0.550_{\pm 0.039}$	$0.559_{\pm 0.039}$
	Sel	$0.579_{\pm 0.022}$	$0.611_{\pm 0.024}$	$0.566_{\pm 0.021}$	$0.576_{\pm 0.020}$
Lasso	Unsel	$0.573_{\pm 0.021}$	$0.601_{\pm 0.017}$	$0.560_{\pm 0.026}$	$0.569_{\pm 0.025}$
	Sel	$0.577_{\pm 0.021}$	$0.611_{\pm 0.015}$	$0.563_{\pm 0.020}$	$0.575_{\pm 0.019}$
Our	AO	$0.531_{\pm 0.024}$	$0.557_{\pm 0.034}$	$0.516_{\pm 0.020}$	$0.522_{\pm 0.021}$
	MP	$0.587_{\pm 0.023}$	$\mathbf{0.619_{\pm 0.020}}$	$0.575_{\pm 0.024}$	$0.586_{\pm 0.023}$
	AO+MP	$0.584_{\pm 0.021}$	$0.617_{\pm 0.017}$	$0.575_{\pm 0.027}$	$0.585_{\pm 0.025}$
	UP	$0.530_{\pm 0.047}$	$0.546_{\pm 0.092}$	$0.502_{\pm 0.056}$	$0.499_{\pm 0.080}$
	AO+UP	$0.547_{\pm 0.041}$	$0.571_{\pm 0.044}$	$0.532_{\pm 0.033}$	$0.540_{\pm 0.035}$

Table 30: Downstream classification performance of different feature partitions on the NB side of the qCPs–NB dataset.

Methods	Setting	Acc.	Precision	Recall	F1
	NB All	$0.591_{\pm 0.019}$	$0.605_{\pm 0.014}$	$0.582_{\pm 0.021}$	$0.587_{\pm 0.018}$
SCCA	Unsel	$0.574_{\pm 0.031}$	$0.586_{\pm 0.030}$	$0.568_{\pm 0.030}$	$0.572_{\pm 0.029}$
	Sel	$0.576_{\pm 0.028}$	$0.593_{\pm 0.027}$	$0.569_{\pm 0.027}$	$0.575_{\pm 0.028}$
DSCCA	Unsel	$0.572_{\pm 0.035}$	$0.580_{\pm 0.034}$	$0.562_{\pm 0.034}$	$0.565_{\pm 0.033}$
	Sel	$0.576_{\pm 0.026}$	$0.592_{\pm 0.022}$	$0.571_{\pm 0.031}$	$0.575_{\pm 0.027}$
Lasso	Unsel	$0.569_{\pm 0.037}$	$0.586_{\pm 0.038}$	$0.562_{\pm 0.038}$	$0.568_{\pm 0.038}$
	Sel	$0.576_{\pm 0.021}$	$0.590_{\pm 0.029}$	$0.568_{\pm 0.022}$	$0.572_{\pm 0.025}$
Our	AO	$0.573_{\pm 0.036}$	$0.597_{\pm 0.041}$	$0.560_{\pm 0.035}$	$0.568_{\pm 0.037}$
	MP	$0.577_{\pm 0.023}$	$0.587_{\pm 0.025}$	$0.572_{\pm 0.026}$	$0.573_{\pm 0.026}$
	AO+MP	$\mathbf{0.605_{\pm 0.030}}$	$\mathbf{0.618_{\pm 0.027}}$	$\mathbf{0.599_{\pm 0.033}}$	$\mathbf{0.604_{\pm 0.032}}$
	UP	$0.523_{\pm 0.051}$	$0.538_{\pm 0.061}$	$0.512_{\pm 0.049}$	$0.516_{\pm 0.051}$
	AO+UP	$0.563_{\pm 0.047}$	$0.572_{\pm 0.045}$	$0.555_{\pm 0.049}$	$0.557_{\pm 0.047}$

standard model also preserves more targets above CCC > 0.6 than the 'no Gate' and 'no InputGate' variants, while matching the best count at CCC > 0.7.

Table 31: Additional evaluation metrics of ablation study on model components on the ADAS–CDR dataset.

Direction	Variants	$\overline{\text{CCC}}_{>.6}$	$\#\text{CCC}_{>.5}$	$\#\text{CCC}_{>.6}$	$\#\text{CCC}_{>.7}$
ADAS → CDR	no Gate	$0.639_{\pm 0.024}$	$6.00[4.25, 6.00]$	$2.50[1.00, 4.50]$	$0.00[0.00, 1.00]$
	no InputGate	$0.716_{\pm 0.051}$	$\mathbf{7.00[7.00, 7.00]}$	$\mathbf{7.00[6.00, 7.00]}$	$\mathbf{5.00[1.00, 6.75]}$
	no OutputGate	$0.664_{\pm 0.028}$	$6.50[5.25, 7.00]$	$4.50[1.50, 5.75]$	$0.500[0.00, 2.00]$
	standard	$\mathbf{0.739_{\pm 0.034}}$	$\mathbf{7.00[7.00, 7.00]}$	$\mathbf{7.00[7.00, 7.00]}$	$\mathbf{5.00[5.00, 7.00]}$
CDR → ADAS	no Gate	$0.655_{\pm 0.037}$	$4.00[4.00, 5.00]$	$2.00[2.00, 2.75]$	$0.00[0.00, 0.750]$
	no InputGate	$\mathbf{0.682_{\pm 0.052}}$	$4.00[3.25, 4.00]$	$2.00[2.00, 2.75]$	$\mathbf{1.00[0.00, 1.00]}$
	no OutputGate	$0.657_{\pm 0.030}$	$4.00[4.00, 6.00]$	$\mathbf{3.00[2.00, 3.75]}$	$\mathbf{1.00[0.25, 1.00]}$
	standard	$0.676_{\pm 0.020}$	$4.00[4.00, 4.00]$	$\mathbf{3.00[2.00, 3.00]}$	$\mathbf{1.00[1.00, 1.00]}$

qCPs-NB Table 32 and Table 33 provide the complete evaluation metrics for the model component ablation study on the qCPs-NB dataset. Overall, the observed trends are consistent with those on ADAS-CDR.

Overall, removing the output-selection mechanism is particularly harmful. This includes both the 'no Gate' and 'no outputGate' variants, since neither model has an explicit mechanism for identifying which target variables are predictable across views. As a result, these variants are forced to predict all

Table 32: Ablation study on model components on the qCPs-NB dataset.

Direction	Variants	Avg. R^2	$\overline{\text{CCC}}_{>.5}$	$\overline{\text{CCC}}_{>.6}$	$\overline{\text{CCC}}_{>.7}$
qCPs → NB	no Gate	$-1.235_{\pm 1.119}$	$0.569_{\pm 0.030}$	$0.624_{\pm 0.014}$	–
	no InputGate	$0.323_{\pm 0.285}$	$0.672_{\pm 0.038}$	$0.676_{\pm 0.034}$	$\mathbf{0.726_{\pm 0.011}}$
	no OutputGate	$-1.613_{\pm 2.333}$	$0.550_{\pm 0.025}$	$0.608_{\pm 0.006}$	–
	Ours	$\mathbf{0.514_{\pm 0.046}}$	$\mathbf{0.683_{\pm 0.029}}$	$\mathbf{0.685_{\pm 0.028}}$	$0.721_{\pm 0.011}$
NB → qCPs	no Gate	$-0.613_{\pm 0.150}$	$0.547_{\pm 0.024}$	$0.618_{\pm 0.020}$	–
	no InputGate	$0.445_{\pm 0.034}$	$0.643_{\pm 0.028}$	$0.667_{\pm 0.018}$	$0.714_{\pm 0.009}$
	no OutputGate	$-0.472_{\pm 0.064}$	$0.556_{\pm 0.021}$	$0.644_{\pm 0.030}$	–
	Ours	$\mathbf{0.448_{\pm 0.046}}$	$\mathbf{0.645_{\pm 0.034}}$	$\mathbf{0.671_{\pm 0.018}}$	$\mathbf{0.724_{\pm 0.010}}$

Table 33: Ablation study on model components on the qCPs-NB dataset.

Direction	Variants	#CCC>0.5	#CCC>0.6	#CCC>0.7
qCPs → NB	no Gate	$5.50[4.00, 6.00]$	$1.00[0.00, 2.50]$	$0.00[0.00, 0.00]$
	no InputGate	$\mathbf{6.00[6.00, 6.00]}$	$\mathbf{6.00[5.00, 6.00]}$	$1.00[0.00, 3.50]$
	no OutputGate	$5.50[4.25, 6.00]$	$0.00[0.00, 0.75]$	$0.00[0.00, 0.00]$
	Ours	$\mathbf{6.00[6.00, 6.00]}$	$\mathbf{6.00[5.25, 6.00]}$	$\mathbf{2.50[1.25, 3.00]}$
NB → qCPs	no Gate	$5.50[4.00, 7.50]$	$0.00[0.00, 0.00]$	$0.00[0.00, 0.00]$
	no InputGate	$\mathbf{15.0[15.0, 15.0]}$	$\mathbf{12.0[10.3, 12.0]}$	$3.00[2.25, 3.75]$
	no OutputGate	$6.00[4.50, 8.50]$	$0.00[0.00, 0.750]$	$0.00[0.00, 0.00]$
	Ours	$\mathbf{15.0[15.0, 15.0]}$	$\mathbf{12.0[9.25, 13.75]}$	$\mathbf{3.50[1.50, 4.00]}$

target variables, including targets that are weakly predictable or essentially unpredictable from the opposite view. This explains both their substantially lower average R^2 and their smaller number of high-concordance targets. In contrast, the full model and the 'no InputGate' variant both include an output gate to focus the prediction objective on reliably predictable MP variables, which leads to a better trade-off between prediction accuracy and the number of selected predictable targets. However, the 'no InputGate' variant does not learn a sparse predictive support and therefore cannot distinguish auxiliary support variables from nuisance variables. Hence, these results confirm that both input and output gates are needed: the output gate is critical for selecting mutually predictable targets, while the input gate is necessary for sparse support discovery and for inducing the full ternary decomposition.

C.2.5 Ablation on Training Stage.

ADAS-CDR Tables 34 and 35 present the training-stage ablation results on the ADAS-CDR dataset. Overall, the full three-stage training scheme achieves the most favorable trade-off between predictive accuracy and the number of identified MP variables. In the ADAS→CDR direction, the advantage of the complete schedule is particularly clear. Compared with single-stage training, our method improves the average R^2 from 0.394 to 0.558 and consistently identifies more targets across all CCC thresholds.

The ablation variants further reveal the role of each stage. When the support discovery stage is removed and only the second and third stages are used, the number of identified MPs drops substantially in the ADAS→CDR direction, from 7.00 to 3.50 under CCC > 0.5 and from 7.00 to 3.00 under CCC > 0.6. This suggests that learning input supports while predicting the full output space is important for sufficiently exploring input–output associations before target selection. In contrast, using only the first and second stages preserves the number of MP variables at lower CCC thresholds, but leads to clearly lower predictive performance than the full model. This indicates that the final joint refinement stage is necessary to further adapt the input supports and predictors to the selected targets.

The CDR→ADAS direction is more challenging. Single-stage training identifies very few MPs, suggesting that directly coupling input and output selection may prematurely discard predictable targets. Although the variant using only the second and third stages achieves a slightly higher average R^2, it selects only one MP variable across all CCC thresholds, indicating an overly selective solution. In contrast, the full three-stage method recovers more MPs while maintaining competitive predictive performance.

Table 34: Ablation study on training stages on the ADAS-CDR dataset.

Direction	Training Stage	Avg. R^2	$\overline{\text{CCC}}_{>.5}$	$\overline{\text{CCC}}_{>.6}$	$\overline{\text{CCC}}_{>.7}$
ADAS → CDR	single stage	$0.394_{\pm 0.043}$	$0.674_{\pm 0.051}$	$0.697_{\pm 0.034}$	$0.747_{\pm 0.014}$
	1st & 2nd stage	$0.438_{\pm 0.134}$	$0.697_{\pm 0.049}$	$0.712_{\pm 0.038}$	$0.751_{\pm 0.024}$
	2nd & 3rd stage	$0.379_{\pm 0.128}$	$0.684_{\pm 0.023}$	$0.696_{\pm 0.019}$	$0.751_{\pm 0.022}$
	Ours	$\mathbf{0.558_{\pm 0.011}}$	$\mathbf{0.733_{\pm 0.047}}$	$\mathbf{0.739_{\pm 0.034}}$	$\mathbf{0.757_{\pm 0.014}}$
CDR → ADAS	single stage	$0.217_{\pm 0.052}$	$0.645_{\pm 0.097}$	$0.695_{\pm 0.070}$	$\mathbf{0.769_{\pm 0.059}}$
	1st & 2nd stage	$0.399_{\pm 0.091}$	$0.636_{\pm 0.025}$	$0.671_{\pm 0.032}$	$0.759_{\pm 0.036}$
	2nd & 3rd stage	$\mathbf{0.430_{\pm 0.169}}$	$\mathbf{0.689_{\pm 0.087}}$	$\mathbf{0.725_{\pm 0.048}}$	$0.748_{\pm 0.040}$
	Ours	$0.421_{\pm 0.096}$	$0.649_{\pm 0.028}$	$0.676_{\pm 0.020}$	$0.759_{\pm 0.040}$

Table 35: Ablation study on training stages on the ADAS-CDR dataset.

Direction	Training Stage	#CCC > 0.5	#CCC > 0.6	#CCC > 0.7
ADAS → CDR	single stage	$7.00[7.00, 7.00]$	$6.00[4.25, 6.75]$	$3.50[1.00, 5.00]$
	1st & 2nd stage	$7.00[7.00, 7.00]$	$6.50[6.00, 7.00]$	$3.50[2.25, 5.00]$
	2nd & 3rd stage	$3.50[3.00, 4.00]$	$3.00[2.25, 3.75]$	$1.50[1.00, 2.00]$
	Ours	$\mathbf{7.00[7.00, 7.00]}$	$\mathbf{7.00[7.00, 7.00]}$	$\mathbf{5.00[5.00, 7.00]}$
CDR → ADAS	single stage	$1.00[1.00, 2.00]$	$1.00[1.00, 1.00]$	$0.00[0.00, 0.750]$
	1st & 2nd stage	$\mathbf{4.00[4.00, 4.00]}$	$\mathbf{3.00[2.00, 3.75]}$	$\mathbf{1.00[1.00, 1.00]}$
	2nd & 3rd stage	$1.00[1.00, 1.00]$	$1.00[1.00, 1.00]$	$\mathbf{1.00[0.250, 1.00]}$
	Ours	$\mathbf{4.00[4.00, 4.00]}$	$\mathbf{3.00[2.00, 3.00]}$	$\mathbf{1.00[1.00, 1.00]}$

qCPs-NB Tables 36 and 37 exhibit a similar trend on the qCPs-NB dataset.

Table 36: Ablation study on training stages on the qCPs-NB dataset.

Direction	Methods	Avg. R^2	$\overline{\text{CCC}}_{>.5}$	$\overline{\text{CCC}}_{>.6}$	$\overline{\text{CCC}}_{>.7}$
qCPs → NB	single stage	$0.116_{\pm 0.100}$	$0.588_{\pm 0.049}$	$0.643_{\pm 0.026}$	–
	1st & 2nd stage	$0.381_{\pm 0.254}$	$0.665_{\pm 0.040}$	$0.670_{\pm 0.034}$	$\mathbf{0.728_{\pm 0.019}}$
	2nd & 3rd stage	$0.284_{\pm 0.190}$	$0.633_{\pm 0.038}$	$0.647_{\pm 0.023}$	–
	Ours	$\mathbf{0.514_{\pm 0.046}}$	$\mathbf{0.683_{\pm 0.029}}$	$\mathbf{0.685_{\pm 0.028}}$	$0.721_{\pm 0.011}$
NB → qCPs	single stage	$0.128_{\pm 0.156}$	$0.597_{\pm 0.049}$	$0.645_{\pm 0.033}$	$\mathbf{0.725_{\pm 0.012}}$
	1st & 2nd stage	$0.365_{\pm 0.134}$	$0.631_{\pm 0.041}$	$0.664_{\pm 0.026}$	$0.723_{\pm 0.019}$
	2nd & 3rd stage	$0.379_{\pm 0.072}$	$0.643_{\pm 0.029}$	$0.668_{\pm 0.019}$	$0.721_{\pm 0.014}$
	Ours	$\mathbf{0.448_{\pm 0.046}}$	$\mathbf{0.645_{\pm 0.034}}$	$\mathbf{0.671_{\pm 0.018}}$	$0.724_{\pm 0.010}$

C.2.6 Stability and Interpretability

Figure 1 provides an overview of the seed-wise stability and feature-level assignment consistency of the recovered partitions on the ADAS-CDR dataset. MP exhibits the highest seed-wise stability among the three partitions. Specifically, the mean Jaccard similarity is 0.746 for ADAS-Cog and 0.975 for CDR, both with relatively small standard deviations. In contrast, AO shows moderate stability, with mean Jaccard similarities ranging from approximately 0.30 to 0.45. UP is the least stable partition, with mean Jaccard similarities of 0.265 for ADAS-Cog and 0.330 for CDR. These results indicate that MP forms a robust core, whereas AO and UP are more sensitive to seed-to-seed variation. The assignment-frequency heatmaps provide a feature-level perspective consistent with the Jaccard analysis.

As shown in Figure 2, the partition statistics and feature-assignment heatmaps on the qCPs-NB dataset exhibit a pattern broadly consistent with that observed on the ADAS-CDR dataset. Specifically, MP shows the highest seed-wise stability among the three partitions, with mean stability values ranging from approximately 0.7 to 0.9 across the two views. In contrast, AO and UP generally show lower stability, mostly falling within the 0.3-0.5 range. A notable difference from the ADAS-CDR dataset is that the UP partition in the NB view is substantially more stable. This suggests that NB contains multiple view-specific features that are not represented in qCPs.

Table 37: Ablation study on training stages on the qCPs-NB dataset.

Direction	Methods	#CCC > 0.5	#CCC > 0.6	#CCC > 0.7
qCPs → NB	single stage	$5.00[3.25, 5.00]$	$1.50[0.00, 4.00]$	$0.00[0.00, 0.00]$
	1st & 2nd stage	$\mathbf{6.00[6.00, 6.00]}$	$\mathbf{6.00[5.25, 6.00]}$	$1.00[0.00, 2.75]$
	2nd & 3rd stage	$3.50[2.25, 4.00]$	$3.00[1.25, 3.75]$	$0.00[0.00, 0.00]$
	Ours	$\mathbf{6.00[6.00, 6.00]}$	$\mathbf{6.00[5.25, 6.00]}$	$\mathbf{2.50[1.25, 3.00]}$
NB → qCPs	single stage	$9.50[8.25, 10.8]$	$2.50[1.25, 9.25]$	$0.00[0.00, 1.50]$
	1st & 2nd stage	$\mathbf{15.0[14.3, 15.0]}$	$10.5[6.00, 12.0]$	$2.00[0.250, 3.50]$
	2nd & 3rd stage	$10.0[9.00, 11.0]$	$7.50[5.50, 10.0]$	$1.00[0.250, 2.75]$
	Ours	$\mathbf{15.0[15.0, 15.0]}$	$\mathbf{12.0[9.25, 13.75]}$	$\mathbf{3.50[1.50, 4.00]}$

Partition	Dataset	Mean $\pm$ Std
MP	ADAS	0.746 ± 0.036
MP	CDR	0.975 ± 0.002
AO	ADAS	0.453 ± 0.018
AO	CDR	0.300 ± 0.175
UP	ADAS	0.265 ± 0.043
UP	CDR	0.330 ± 0.158

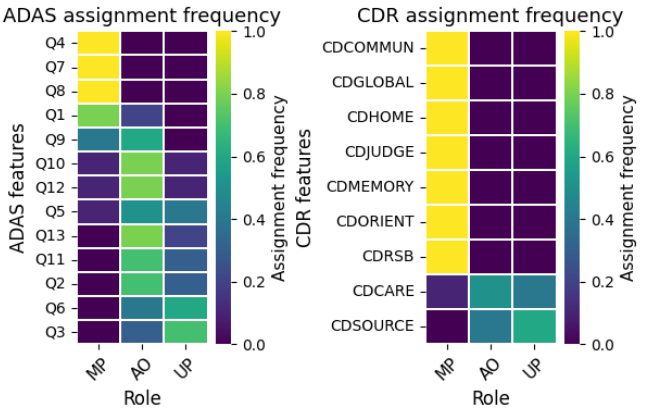


Figure 1: Partition statistics (left) and assignment frequency heatmaps (right) on ADAS–CDR. The table reports the mean $\pm$ standard deviation of the three partitions across seeds, while the heatmaps visualize the feature-level assignment frequency to each partition

Case study of ADAS We illustrate the interpretability of the ternary partitions on the ADAS-CDR dataset and their cross-view semantic relationships.

On the ADAS side, the MP variables consist of Q4 (Delayed Word Recall), Q7 (Orientation), Q8 (Word Recognition), and Q1 (Word Recall), which primarily measure memory and orientation functions. These domains have direct or closely aligned counterparts in CDR, such as memory (CDMEMORY) and orientation (CDORIENT). The identification of these variables as MP is therefore consistent with their shared semantic roles across the two assessments.

In contrast, Q10 (Comprehension of Spoken Language) and Q12 (Spoken Language Ability) are frequently assigned to AO. Both variables capture language-related abilities, yet CDR does not include an explicit language domain. As a result, these variables are difficult to predict from CDR, explaining their exclusion from MP. However, language abilities can still influence broader functional outcomes measured in CDR, such as community affairs (CDCOMMUN) and personal care or judgment-related functions (CDCARE, CDJUDGE), which rely on communication and comprehension. This may explain why these variables, while not directly predictable, provide complementary information for predicting CDR-measured impairment.

Finally, Q3 (Constructional Praxis), which measures visuoconstructional ability, is most frequently assigned to UP. This is consistent with the absence of a direct counterpart in CDR, which does not explicitly assess visuospatial construction at the task level. Although such abilities may indirectly affect functional performance, they are not a primary dimension captured by CDR, leading to limited cross-view predictability and utility.

Overall, this case study provides qualitative evidence that all three partitions capture distinct and meaningful roles: MP variables align with shared domains across views, AO variables provide auxiliary yet functionally relevant information, and UP variables correspond to dimensions with limited cross-view relevance.

Partition	Dataset	Mean $\pm$ Std
MP	qCPs	0.934 ± 0.007
MP	NB	0.902 ± 0.002
AO	qCPs	0.483 ± 0.024
AO	NB	0.417 ± 0.010
UP	qCPs	0.212 ± 0.038
UP	NB	0.603 ± 0.010

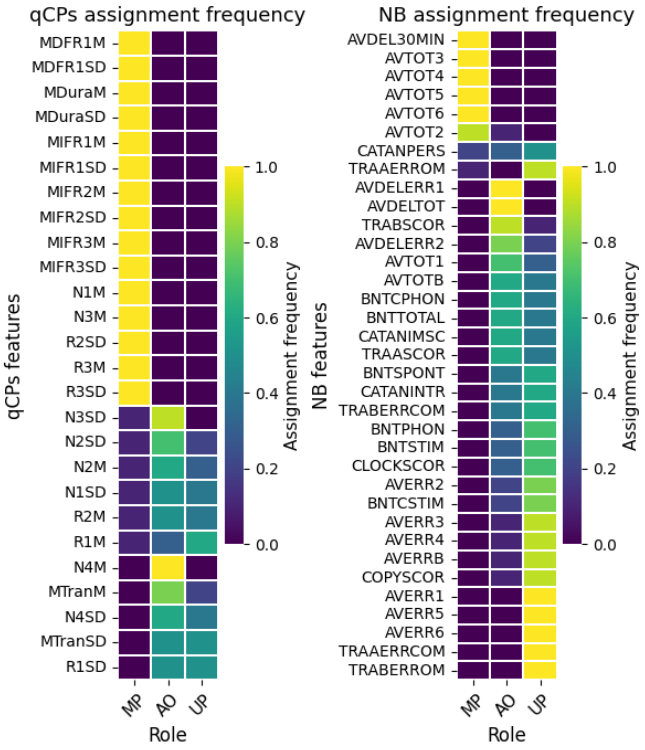


Figure 2: Partition statistics (left) and assignment frequency heatmaps (right) on qCPs–NB. The table reports the mean $\pm$ standard deviation of three partitions across seeds, while the heatmaps visualize the feature-level assignment frequency to each partition

Potential application to Parkinson's disease. A further prospective application concerns the relationship between motor examinations and assessments of activities of daily living in Parkinson's disease. Consider a patient may undergo a motor examination without completing an activities-of-daily-living assessment. MPD could estimate predictable functional limitations, identify motor features that are important for those estimates, and distinguish measurements that provide unique clinical information. This could support assessment reduction, rehabilitation planning, and decisions about which variables should be retained in abbreviated clinical protocols.As this scenario is not evaluated in our experiments, it is presented only as a potential extension and would require dedicated clinical validation before practical use.

Broader impacts. This work may have positive impact by helping researchers interpret variable-level relationships across different assessment instruments, as illustrated by the ADAS–CDR case study. The learned MP/AO/UP partitions can support exploratory analysis and hypothesis generation, but should not be interpreted as causal mechanisms or used for clinical decision-making without external validation and expert oversight. Although our real-world experiments use publicly available and de-identified ADNI-derived data, future applications to sensitive medical datasets should follow appropriate privacy, governance, and fairness safeguards.

C.2.7 Sensitivity Analysis of Incentive Weights

We further examine how the output-gate incentive weight λ_{out} affects the number of candidate MP variables identified by DGNN. Since Lin's concordance correlation coefficient (CCC) measures both correlation and agreement between predicted and observed values, we use CCC > 0.5 as a permissive lower bound for practically meaningful predictive concordance in this medical prediction setting, and additionally report stricter thresholds CCC > 0.6 and CCC > 0.7 as robustness checks.

Figure 3 shows the sensitivity results on the ADAS–CDR dataset. When λ_{out} is small, DGNN selects only a few candidate MP variables. This is expected because the prediction loss term penalizes

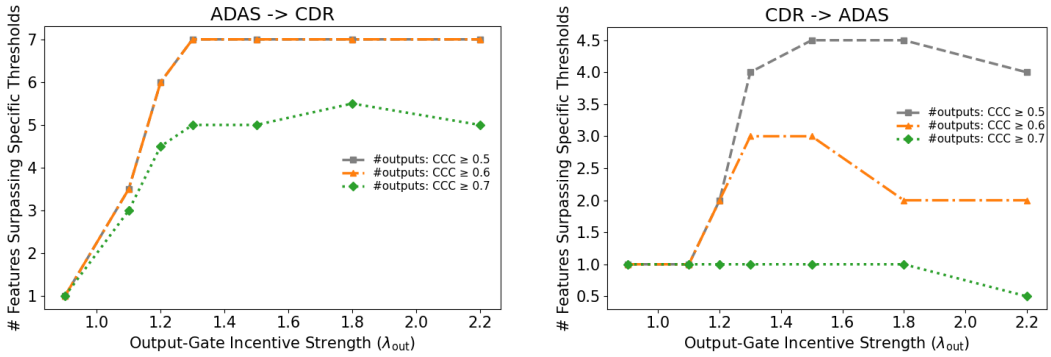


Figure 3: Sensitivity analysis of incentive weights in ADAS-CDR dataset.

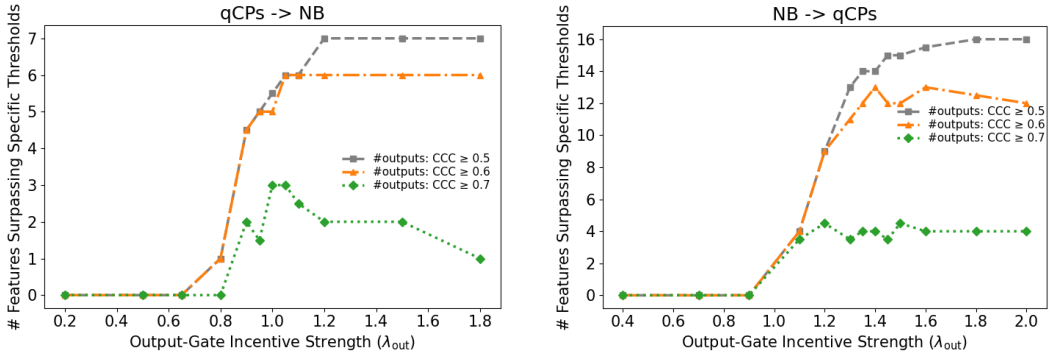


Figure 4: Sensitivity analysis of incentive weights in qCPs-NB dataset.

active output gates whose targets are not accurately reconstructed, while a weak output incentive provides little pressure to retain additional targets. As λ_{out} increases, the incentive term counteracts this conservative behavior and encourages the model to open more output gates. Consequently, the number of candidate MP variables passing the CCC thresholds increases and then stabilizes, forming a clear plateau over a range of λ_{out} values.

This pattern supports the intended role of λ_{out} as a coverage-control parameter for MP discovery. A small value tends to under-select predictable outputs, producing an overly conservative MP set. A moderate value allows the model to recover a larger set of variables with nontrivial held-out predictive concordance. The plateau suggests that, after the main cross-view predictable structure has been recovered, further increasing the incentive does not lead to an unbounded increase in meaningful candidate MP variables. A similar phenomenon is observed on the qCPs–NB dataset in Figure 4.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the MPD formulation, the DGNN relaxation, and the main empirical claims. These are supported by the theoretical analysis and experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the Discussion, including the current focus on pairwise views, reliance on differentiable optimization, and the need for further theoretical and information-theoretic analysis.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results are stated with their assumptions, and the corresponding proofs are provided in the appendix with appropriate numbering and cross-references.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the synthetic data-generation procedure, dataset preprocessing, model architecture, training schedule, baselines, evaluation metrics, and hyperparameter/configuration details needed to reproduce the main experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is publicly available at `https://github.com/UOW-AMRL/NeurIPS26-MPD`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the data splits, preprocessing, baselines, evaluation metrics, model architecture, training stages, and hyperparameter selection procedure in the main text and appendix, which are sufficient to understand the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main experimental tables report mean and standard deviation over 10 runs or random seeds, and the text states the source of variability. These error bars cover the experiments supporting the main empirical claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The required compute resources are described in the Implementation Details section of the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses synthetic data and public-access ADNI-derived datasets, and is conducted as a methodological study without interventions, deception, or direct interaction with human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses positive impacts for interpreting relationships across medical assessment instruments, as illustrated by the ADAS–CDR case study, and also notes risks from causal over-interpretation, insufficient validation, inappropriate clinical use, and privacy/fairness concerns in future applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk models or scraped datasets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the ADNI database using the required acknowledgment language, states that the ADNI-2 derived datasets were obtained from ADNI and accessed through its official data-access process, and does not redistribute the raw data.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The proposed method and synthetic data-generation procedure are documented in the paper and appendix. The implementation code is publicly available on GitHub.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or direct interaction with human subjects. The real-world experiments use existing ADNI-derived, de-identified data accessed through the ADNI data-access process.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The study does not involve crowdsourcing or direct human-subject experiments; it only performs secondary analysis of existing, de-identified ADNI-derived data accessed through the ADNI data-access process.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.